Proceedings of META'2016

6th International Conference on Metaheuristics and Nature Inspired computing

Sponsors

Table of contents

Metaheuristics - General stream

Algorithm Selector and Prescheduler in the ICON challenge

François Gonard[1,2,3], Marc Schoenauer[2,3,1], and Michèle Sebag[3,2]

[1] Technological Research Institute SystemX**, 8 avenue de la Vauve 91127 Palaiseau cedex France
[2] INRIA
[3] LRI, CNRS UMR8623 and Université Paris-Sud, Bat. 660 Claude Shannon 91405 Orsay cedex France

Abstract. Algorithm portfolios are known to offer robust performances, efficiently overcoming the weakness of every single algorithm on some particular problem instances. Two complementary approaches to get the best out of an algorithm portfolio is to achieve algorithm selection (AS), and to define a scheduler, sequentially launching a few algorithms on a limited computational budget each. The presented *Algorithm Selector And Prescheduler* system relies on the joint optimization of a pre-scheduler and a *per instance* AS, selecting an algorithm well-suited to the problem instance at hand. ASAP has been thoroughly evaluated against the state-of-the-art during the ICON challenge for algorithm selection, receiving an honourable mention. Its evaluation on several combinatorial optimization benchmarks exposes surprisingly good results of the simple heuristics used; some extensions thereof are presented and discussed in the paper.

1 Introduction

In quite a few domains related to combinatorial optimization, such as satisfiability, constraint solving or operations research, it has been acknowledged for some decades that there exists no universal algorithm, dominating all other algorithms on all problem instances. This result, referred to as No Free Lunch theorem [20], has prompted the scientific community to design algorithm portfolios addressing the various types of difficulties involved in the problem instances, *i.e.*, such that at least one algorithm in the portfolio can efficiently handle any problem instance [9, 6]. Algorithm portfolios thus raise a new issue, that of selecting *a priori* an algorithm well suited to the application domain [12]. This issue, referred to as *Algorithm Selection* (AS) and first formalized by Rice [18], is key to the successful transfer of algorithms outside of research labs. It has been tackled by a number of authors in the last years (more in section 2).

Algorithm selection comes in different flavors, depending on whether the goal is to yield an optimal performance in expectation with respect to a given distribution of problem instances (global AS), or an optimal performance on a particular problem instance (*per instance* AS). Note that the measure of performance depends on the domain (e.g., time-to-solution in satisfiability, or time to reach the optimal solution up to a given precision in optimization[4]). This paper focuses on the *per-instance* setting, aimed at achieving peak performance on every problem instance.

In some domains, it is often the case that some problems can be solved in no time by some algorithms. It thus makes sense to allocate a part of the computational budget to a *pre-scheduler*, sequentially launching a few algorithms with a small computational budget each. The pre-scheduler is expected to solve "easy" instances in a first stage; in a second stage, AS is only launched on problem instances which have not been solved in the pre-scheduler phase. Note that the pre-scheduler enables to extract some additional information characterizing the problem at hand, which can be used together with the initial information about the problem instance, to support the AS phase.

This paper presents the *Algorithm Selector And Prescheduler* system (ASAP), aimed at algorithm selection in the domain of combinatorial optimization (section 3). The main contribution lies in the joint optimization of both a pre-scheduler and a per-instance algorithm selector. The extensive empirical validation of ASAP is conducted on the ICON challenge on algorithm selection [11]. This challenge leverages the Algorithm Selection library [1], aimed at the fair, comprehensive

** This research work has been funded by the French Program "Investissements d'Avenir".

[4] One often considers the joint problems of selecting an algorithm and the optimal hyper-parameters thereof, referred to as *Algorithm Configuration* (AC), as the choice of the hyper-parameter values governs the algorithm performance. AC is outside the scope of the paper and will not be further considered.

and reproducible benchmarking of AS approaches on 13 domains ranging from satisfiability to operations research (section 4).

The comparative empirical validation of ASAP demonstrates its good performances comparatively to state-of-art pre-schedulers and AS approaches (section 5), and its complementarity with respect to the prominent Zilla algorithms [22]. The paper concludes with a discussion of the limitations of the ASAP approach, and some perspectives for further research.

2 Related work

2.1 Algorithm selectors

The algorithm selection issue, aimed at selecting the algorithm best suited to the problem at hand, was first formalized by Rice [18] as follows. Given a problem space mapping each problem instance onto a description $\mathbf{x}$ thereof (usually $\mathbf{x}$ in $\mathbb{R}^d$) and the set $\mathcal{A}$ of algorithms in the portfolio, let us denote $\mathcal{G}(\mathbf{x}, a)$ a performance model, mapping each ($\mathbf{x}$, a) pair onto the performance of algorithm a onto problem instance $\mathbf{x}$. AS most naturally follows from such a performance model by selecting for each problem instance $\mathbf{x}$ the algorithm a with optimal $\mathcal{G}(\mathbf{x}, a)$.

$$AS(\mathbf{x}) = \arg\max_{a \in \mathcal{A}} \{\mathcal{G}(\mathbf{x}, a)\} \tag{1}$$

The performance model is most usually built by applying machine learning approaches onto a dataset reporting the algorithm performances on a comprehensive set of benchmark problem instances (with the exception of [5], using a multi-armed bandit approach). Such machine learning approaches range from k-nearest neighbors [15] to ridge regression [22], random forests [23], collaborative filtering [19, 14], or learning to rank approaches [16].

As expected, the efficiency of the machine learning approaches critically depends on the quality of the training data: the representativity of the problem instances used to train the performance model, and even more importantly, the description of the problem instances. Considerable care has been devoted to the definition of descriptive features in the SAT and Constraint domains [21].

2.2 Schedulers

Besides AS, an algorithm portfolio can also take advantage of parallel computer architectures, by launching several algorithms working independently or in cooperation on the considered problem instance (see e.g. [24], [10]). Schedulers embed the parallel solving strategies in a sequential computer architecture, by considering a sequence of κ (algorithm a_i, time-out τ_i) pairs, where the problem instance is successively tackled by algorithm a_i with a computational budget τ_i, until being solved. Notably, the famed restart strategy, launching a same algorithm with different random seeds or different initial conditions can be viewed as a particular case of scheduling strategy [6]. Likewise, AS can be viewed as a particular case of scheduler with $\kappa = 1$ and τ_1 set to the overall computational budget.

As shown by [23], schedulers and AS can be combined together along a multi-stage process, where a scheduler solves easy instances in a first stage, and remaining instances are handled by the AS and tackled by the selected algorithm in the next stage. [10] build *per-instance* schedules where the AS is one of the component algorithms incorporated.

3 Overview of ASAP

This section first discusses the rationale for the ASAP approach, before detailing the pre-scheduler and AS modules in ASAP.V1. Extensions thereof, forming ASAP.V2, are presented thereafter.

3.1 Analysis

A benchmark suite most generally involves easy and hard problem instances. The difficulty is that the hardness of a problem instance depends on the considered algorithm. As shown on Fig. 1 in the case of the SAT11-HAND dataset (section 4), while several algorithms might solve 20% of the

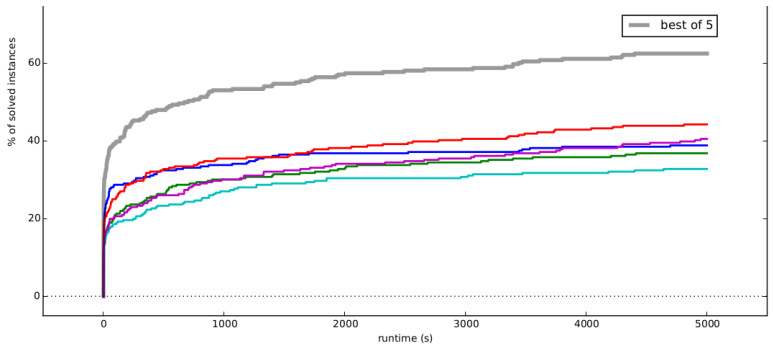


Fig. 1. Percentage of solved instances vs. runtime on the SAT11-HAND dataset, for 5 algorithms and the oracle (selecting the best algorithm out of 5 for each problem instance).

problem instances within seconds, the oracle (selecting the best one out of these algorithms for each problem instance) solves about 40% of the problem instances within seconds.

Accordingly, one might want to launch each one of these algorithms for a few seconds each on each problem instance: after this stage, referred to as pre-scheduler stage, circa 40% of the overall problem instances would be solved.

Definition 1 (Pre-scheduler). *Let $\mathcal{A}$ be a set of algorithms. A κ-component pre-scheduler, defined as a sequence of κ (algorithm a_i, time-out τ_i) pairs,*

$$((a_i, \tau_i)_{i=1}^{\kappa}) \ \ with (a_i, \tau_i) \in \mathcal{A} \times \mathbb{R}^+, \ \forall i \in 1, \ldots, \kappa$$

sequentially launches algorithm a_j on any problem instance $\mathbf{x}$ until either a_j solves $\mathbf{x}$, or time τ_j is reached, or a_j stops without solving $\mathbf{x}$. If $\mathbf{x}$ has been solved, the execution stops. Otherwise, j is incremented while $j \leq \kappa$.

Note that a pre-scheduler contributes to reduce the impact of the AS failures. An AS failure is manifested as a problem instance $\mathbf{x}$ for which the AS selects an inappropriate algorithm (requiring much computational resources to solve $\mathbf{x}$ or even failing to solve it), although there exists another algorithm which could have solved $\mathbf{x}$ in no time. Since a pre-scheduler increases the chance for each problem instance to be solved in no time, everything else being equal, it therefore mitigates the chances and impact of AS failures.

After this discussion, the ASAP system involves two modules, a pre-scheduler and an AS. The pre-scheduler is meant to solve as many problem instances as possible in a first stage, and the AS takes care of the remaining problem instances. A primary decision concerns the division of labor between the two modules: how to split the available runtime between the two, and how many algorithms are involved in the pre-scheduler (parameter κ). It is clear that the number of problem instances solved by a module will increase with its computational budget, everything else being equal; the pre-scheduler and the AS modules are interdependent. For simplicity and tractability however, the maximal runtime allocated to the pre-scheduler is fixed to T_{ps}^{max} (10% of the overall computational budget in the experiments, section 5), and the number κ of algorithms in the pre-scheduler is set to 3. [10] and [13] use a most close setup for their fixed-split selection schedules, except they do not constrain the AS component to take the last part of the schedule.

Given T_{ps}^{max} and κ, ASAP tackles the optimization of the pre-scheduler and the AS modules. It is clear that both optimization problems remain inter-dependent: the AS should mostly focus on the problem instances which are not solved by the pre-scheduler, while the pre-scheduler should symmetrically focus on the problem instances which are most uncertain or badly identified by the AS. Formally, this interdependence is handled as follows:

- A performance model $\mathcal{G}(\mathbf{x}, a)$ is built for each algorithm over all training problem instances, defining AS_{init} (Eq. 1);
- A pre-scheduler is built to optimize the joint performance (pre-scheduler, AS_{init}) over all training problem instances;

– Another performance model $\mathcal{G}2(\mathbf{x}, a)$ is built over all training problem instances, using an additional boolean feature that indicates for each problem instance whether it was solved by the above pre-scheduler; let AS_{post} denote the AS based on performance model $\mathcal{G}2(\mathbf{x}, a)$.

ASAP finally is composed of the pre-scheduler followed by AS_{post}.

3.2 ASAP.V1 pre-scheduler

Let $(a_i, \tau_i)_{i=1}^{\kappa}$ denote a pre-scheduler, with overall computational budget $T_{ps} = \sum_{i=1}^{\kappa} \tau_i$, and let $\mathcal{F}\left((a_i, \tau_i)_{i=1}^{\kappa}\right)$ denote the associated domain-dependent performance. ASAP.V1 considers for simplicity equal time-outs ($a_i = \frac{T_{ps}}{\kappa}, i = 1 \ldots \kappa$). The pre-scheduler is thus obtained by solving the following optimization problem:

$$\max_{T_{ps} \leq T_{ps}^{max}, a_1, \ldots a_\kappa} \left\{ \mathcal{F}\left((a_i, \frac{T_{ps}}{\kappa})_{i=1}^{\kappa} \right) \right\} \tag{2}$$

This mixed optimization problem is tackled in a hierarchical way, determining for each value of T_{ps} the optimal κ-uple of algorithms $a_1 \ldots a_\kappa$. Thanks to both small κ values ($\kappa = 3$ in the experiments) and small number of algorithms (≤ 31 in the ICON challenge, section 4), the optimal κ-uple is determined by exhaustive search conditionally to the T_{ps} value.

The ASAP.V1 pre-scheduler finally relies on the 1-dimensional optimization of the overall computational budget T_{ps} allocated to the pre-scheduler. In all generality, the optimization of T_{ps} is a multi-objective optimization problem, e.g. balancing the overall number of problems solved and the overall computational budget. Multi-objective optimization commonly proceeds by determining the so-called Pareto front, made of non-dominated solutions. In our case, the Pareto front depicts how the performance varies with the overall computational budget, as illustrated on Fig. 2, where the performance is set to the number of solved instances.

In multi-objective decision making [3, 2], the choice of a solution on the Pareto front is tackled using post-optimal techniques [4], including: i) compromise programming, where one wants to find the point the closest to an ideal target in the objective space; ii) aggregation of the objectives into a single one, e.g., using linear combination; or iii) marginal rate of return. The last heuristics consists of identifying the so-called "knees", that is, the points where any small improvement on a given criterion is obtained at the expense of a large decrease on another criterion, defining the so-called marginal rate of return. The vanilla marginal rate of return is however sensitive to strong local discontinuities; for instance, it would select point A in Fig. 2. Therefore, a variant taking into account the global shape of the curve, and measuring the marginal rate of improvement w.r.t. the extreme solutions on the Pareto front is used (e.g., selecting point K instead of point A in Fig.2).

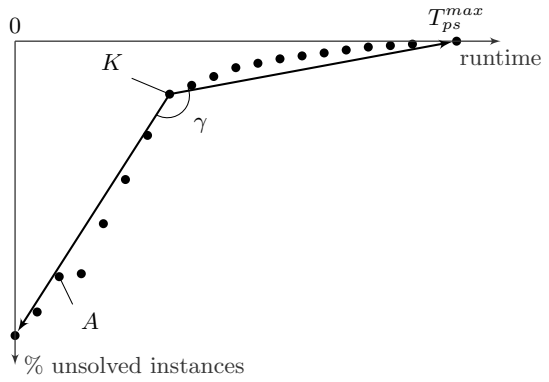


Fig. 2. Among a set of Pareto-optimal solutions, solution A has the best marginal rate of return; solution K, which maximizes the average rate of return w.r.t. the extreme solutions of the Pareto front (maximizing angle γ), is the knee selected in ASAP.

3.3 ASAP.V1 algorithm selector

As detailed in section 3.1, the AS relies on the performance model learned from the training problem instances. Two learning algorithms are considered in this paper: random forests and k-nearest neighbors. One hyper-parameter was adapted for each ML approach (all other hyper-parameters being set to their default value, using the Python scikit-learn library [17]), based on a few preliminary experiments: 35 trees are used for the RandomForest algorithm and the number of neighbors is set to $k = 3$ for the k-nearest neighbors. In the latter case, the predicted value associated to problem instance $\mathbf{x}$ is set to the weighted sum of the performance of its nearest neighbors, weighted by their relative distance to $\mathbf{x}$:

$$\widehat{\mathcal{F}}(\mathbf{x}, a) = \frac{\sum_i ||\mathbf{x} - \mathbf{x}_i|| \mathcal{F}(a, \mathbf{x}_i)}{\sum_i ||\mathbf{x} - \mathbf{x}_i||}$$

where $\mathbf{x}_i$ ranges over the 3 nearest neighbors of $\mathbf{x}$. Features were normalized (zero mean, unit variance) before selecting the neighbors.

A main difficulty comes from the descriptive features forming the representation of problem instances. Typically, the feature values are missing for some groups of features, for quite a few problem instances, due to diverse causes (computation exceeded time limit, exceeded memory, presolved the instance, crashed, other, unknown). The lack of feature value is handled by i) replacing the missing value by the feature average value; ii) adding to the set of descriptive features 7 additional boolean features per group of initial features, indicating whether the feature group values are available or the reason why they are missing otherwise[5] .

3.4 ASAP.V2

Several extensions of ASAP.V1 have been considered after the closing of the ICON challenge, aimed at exploring a richer pre-scheduler-AS search space while preventing the risk of overfitting induced by a larger search space.

We investigated the use of different time-outs for each algorithm in the pre-scheduler, while keeping the set of algorithms $(a_1, \ldots, a_\kappa)$ and the overall computational budget T_{ps}. The sequential optimization strategy (section 3.2), deterministically selecting T_{ps} as the solution with maximal average return rate, exhaustively determining the κ-uple of algorithms conditionally to T_{ps}, is thus extended to optimize the $(\tau_1, \ldots \tau_{\kappa-1})$ vector conditionally to $\sum_{i=1}^{\kappa-1} \leq T_{ps}$, using a prominent continuous black-box optimizer, specifically the Covariance-Matrix Adaptation-Evolution Strategy (CMA-ES) [7].

This extended search space is first investigated by considering the **raw optimization criterion** $\mathcal{F}_{raw}\left((a_i, \tau_i)_{i=1}^{\kappa}\right)$ defined in section 3.2, that is, the cumulative performance of ASAP over all training problem instances. However a richer search space entails some risk of overfitting, where the higher performance on data used to optimize ASAP (training data) is obtained at the expense of a lower performance on test data. Generally speaking, the datasets used to train an AS are small ones.

A **penalized optimization criterion** is thus considered:

$$\mathcal{F}_{L2}\left((a_i, \tau_i)_{i=1}^{\kappa}\right) = \mathcal{F}\left((a_i, \tau_i)_{i=1}^{\kappa}\right) + w \sum_{i=1}^{\kappa} \left(\tau_i - \frac{T_{ps}}{\kappa}\right)^2$$

which penalizes the L_2 distance between the (τ_i) vector and the uniform time outs $(\tau_i = \frac{T_{ps}}{\kappa})$. The rationale for this penalization is to prevent brittle improvements on the training set due to opportunistic adjustments of the τ_is, at the expense of stable performances on further instances. The penalization weight w is adjusted using a nested CV process.

A **randomized optimization criterion** is also considered. By construction, the fitness function aggregates the performances of all training problem instances. As the training problem instances sample the problem domain, this fitness defines a noisy optimization problem. Sophisticated

[5] The increase in the overall number of features is handled by an embedded feature selection mechanism, removing all features with negligible importance criterion ($< 10^{-5}$ in the experiments) in a separately learned 10-trees random forest regression model.

approaches have been proposed to address this noisy optimization issue in non-convex optimization-based machine learning settings (see e.g. [8]). Another approach is proposed here, based on the bootstrap principle: in each CMA-ES generation, the set of n problem instances used to compute the performance is uniformly drawn with replacement from the n-size training set. In this manner, each optimization generation considers a slightly different optimization objective noted $\mathcal{F}_{rand}$, thereby discarding the insignificant improvements.

Finally, a **probabilistic optimization criterion** is considered, handling the ASAP performance on a single problem instance as a random variable with a triangle-shape distribution (Fig. 3) centered on the actual performance $p(\mathbf{x})$, with support in $[p(\mathbf{x}) - \theta, p(\mathbf{x}) + \theta]$, and taking the expectation thereof. The merit of this triangular probability distribution function is to allow for an analytical computation of the overall fitness expectation, noted $\mathcal{F}_{dfp}$.

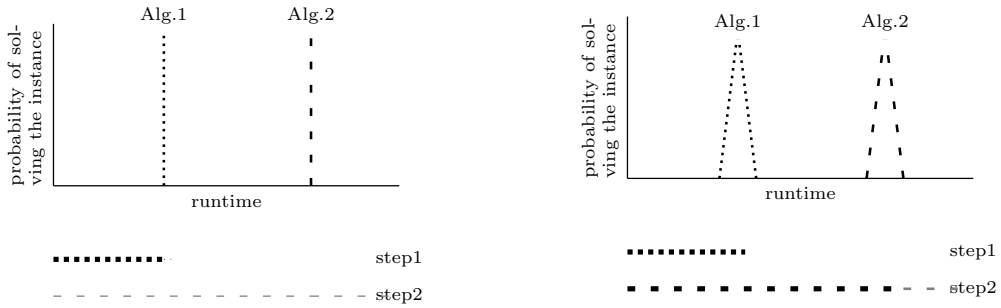


Fig. 3. Schedule execution difference between punctual and triangular pdf. On the left (punctual pdf), schedule stops during step 1. On the right (triangular pdf, part of the instance is not solved during step 1 and the schedule executes until step 2 solves the rest.

4 Experimental setting: The ICON challenge

4.1 ASlib data format

Due to the difficulty of comparing the many algorithm selection systems and the high entry ticket to the AS field, a joint effort was undertaken to build the Algorithm Selection Library (ASlib), providing comprehensive resources to facilitate the design, sharing and comparison of AS systems [1]. ASlib (version 1.0.1) involves 13 datasets, also called scenarios (Table 1), gathered from recent challenges and surveys in the operations research, artificial intelligence and optimization fields. The interested reader is referred to [1] for a more comprehensive presentation.

Table 1. ASlib datasets (V1.0.1)

dataset	# instances	# algorithms	# features
ASP-POTASSCO	1294	11	138
CSP-2010	2024	2	86
MAXSAT12-PMS	876	6	37
PREMARSHALLING-ASTAR-2013	527	4	16
PROTEUS-2014	4021	22	198
QBF-2011	1368	5	46
SAT11-HAND	296	15	115
SAT11-INDU	300	18	115
SAT11-RAND	600	9	115
SAT12-ALL	1614	31	115
SAT12-HAND	767	31	115
SAT12-INDU	1167	31	115
SAT12-RAND	1362	31	115

Each dataset includes i) the performance and computation status of each algorithm on each problem instance; ii) the description of each problem instance, as a vector of the expert-designed feature values (as said, this description considerably facilitates the comparison of the AS systems); iii) the computational status of each such feature (e.g. indicating whether the feature could be computed, or if it failed due to insufficient computational or memory resources). Last but not least, each dataset is equi-partitioned into 10 subsets, to enforce the reproducibility of the 10 fold CV assessment of every AS algorithm.

4.2 The ICON Challenge on Algorithm Selection

The ICON Challenge on Algorithm Selection, within the ASlib framework, was carried on between February and July 2015 to evaluate AS systems in a fair, comprehensive and reproducible manner[6]. Each submitted system was assessed on the 13 ASlib datasets [1] with respect to three measures: i) number of problem instances solved; ii) extra runtime compared with the virtual best solver (VBS, also called oracle); and iii) Penalized Average Time-10 (PAR10) which is the cumulative runtime needed to solve all problem instances (set to ten times the overall computational budget whenever the problem instance is unsolved).

As the whole datasets were available to the community from the start, the evaluation was based on hidden splits between training and test set. Each submitted system provides a dataset-dependent, instance-dependent schedule of algorithms, optionally preceded by a dataset-dependent presolver (single algorithm running on all instances during a given runtime before the per-instance schedule runs). Each system can also, in a dataset-dependent manner, specify the groups of features to be used (in order to save the time needed to compute useless features).

Two baselines are considered: the oracle, selecting the best algorithm for each problem instance; and the single best (SB) algorithm, with best average performance over all problem instances in the dataset. The baselines are used to normalize every system performance over all datasets, associating performance 0 to the oracle (respectively performance 1 to the single best), supporting the aggregation of the system results over all datasets.

5 Experimental validation

5.1 Comparative results

Table 2 reports the results of all submitted systems on all datasets (the statistical significance tests are reported in Fig. 5). The general trend is that zilla algorithms dominate all other algorithms on the SAT datasets, as expected since they have consistently dominated the SAT contests in the last decade. On non-SAT problems however, zilla algorithms are dominated by ASAP_RF.V1.

The robustness of the ASAP approach is demonstrated as they never rank last; they however perform slightly worse than the single best on some datasets. The rescaled performances of ASAP_RF.V1 is compared to zilla and autofolio (Fig. 4, on the left), demonstrating that ASAP_RF.V1 offers a balanced performance, significantly lower than for zilla and autofolio on the SAT problems, but significantly higher on the other datasets; in this respect it can be viewed as a low-risk system.

5.2 Sensitivity analysis

The sensitivity analysis conducted after the closing of the challenge compares ASAP.V2 (with different time-outs in the pre-scheduler) and ASAP.V1, and examines the impact of the different optimization criteria, aimed at avoiding overfitting: the raw fitness, the L2-penalized fitness, the randomized fitness and the probabilistic fitness (section 3.4).

The impact of the hyper-parameters used in the AS (number of trees set to 35, 100, 200, 300 and 500 trees in the Random Forest) is also investigated.

Table 3 summarizes the experimental results that each ASAP.V2 configuration would have obtained in the ICON challenge framework, together with the actual submissions results, including

[6] The codes of all submitted systems and the results are publicly available, http://challenge.icon-fet.eu/challengeas

Table 2. Normalized performances of submitted systems, aggregated across all folds and all measures (the lower, the better). Ranks of zilla (challenge winner) and ASAP_RF.V1 (honourable mention) are given in parenthesis. Numbers were computed from the challenge outputs.

	ASAP_RF.V1	ASAP_kNN.V1	autofolio	flexfolio	sunny	sunny-presolv	zilla	zillafolio
ASP-POTASSCO	0.294 (2)	0.359	0.299	0.314	0.37	0.336	0.319 (5)	**0.283**
CSP-2010	**0.146** (1)	0.247	0.288	0.223	0.263	0.406	0.2 (3)	0.157
MAXSAT12-PMS	0.168 (4)	0.159	0.45	**0.149**	0.166	0.224	0.201 (5)	0.233
PREMARSHALLING-ASTAR-2013	0.349 (4)	0.369	0.359	0.307	0.325	**0.296**	0.374 (7)	0.385
PROTEUS-2014	0.16 (4)	0.177	0.222	**0.056**	0.134	0.103	0.245 (8)	0.223
QBF-2011	0.097 (2)	**0.091**	0.169	0.096	0.142	0.162	0.191 (7)	0.194
SAT11-HAND	0.341 (4)	0.318	0.342	0.342	0.466	0.464	0.328 (3)	**0.302**
SAT11-INDU	1.036 (5)	0.957	**0.875**	1.144	1.13	1.236	0.905 (2)	0.966
SAT11-RAND	0.104 (6)	0.09	**0.046**	0.226	0.116	0.088	0.053 (2)	0.067
SAT12-ALL	0.392 (5)	0.383	0.306	0.502	0.509	0.532	**0.273** (1)	0.322
SAT12-HAND	0.334 (5)	0.31	**0.256**	0.434	0.45	0.467	0.272 (2)	0.296
SAT12-INDU	0.955 (6)	0.919	0.604	0.884	1.074	1.018	0.618 (3)	**0.594**
SAT12-RAND	1.032 (5)	1.122	0.862	1.073	1.126	0.97	**0.779** (1)	0.79

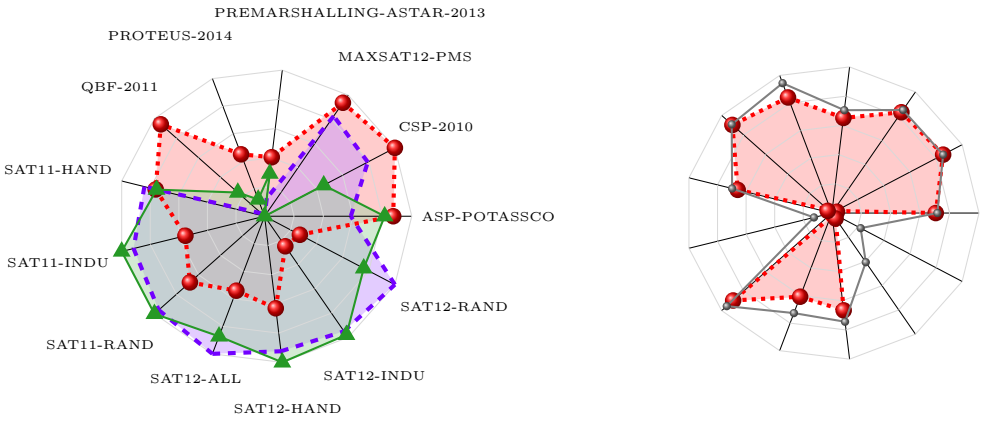


Fig. 4. On the left: per-dataset performances of ASAP_RF.V1 (balls, dotted line), zilla (no marker, dashed line) and autofolio (triangles, solid line) scaled to the range of performance of all submitted systems. As a comparison, the per-dataset best submitted system (small balls, solid line) and ASAP_RF scores before rescaling are depicted on the right.

systems that were not competing in the challenge: llama-regr and llama-regrPairs from the organizers, and autofolio-48 which is identical to autofolio but with 48h time for training (12h was the time limit authorized in the challenge) [11].

The significance analysis, using a Wilcoxon signed-rank test, is reported in Fig. 5. A first result is that all ASAP.V2 variants improve on ASAP.V1 with significance level 1%. A second result is that ASAP.V2 with the probabilistic optimization criterion is not statistically significantly different from zilla, autofolio and zillafolio.

A third and most surprising result is that the difference between the challenge-winner zilla and most of ASAP.V2 variants is not statistically significant.

[8] As the CSP-2010 dataset gives the choice between only two algorithms, the pre-scheduler consists of a single algorithm running for the whole time devoted to the pre-scheduler. For that particular case, all ASAP_RF.V2 variants with the same selector hyperparameter are identical.

Table 3. Optimized pre-scheduler performances[8] aggregated across all datasets, all splits and all measures (the lower, the better). The hyperparameters for f_{L^2} and f_{dfp} were chosen after preliminary experiments using the cross validation provided with ASlib. For each configuration of the selector, the best-evaluated fitness function appears in bold

fitness function (if relevant)	f_{L^2}	f_{dfp}	f_{rand}	f_{raw}	none
ASAP_RF.V2 35	0.416	0.414	0.412	**0.410**	0.414
ASAP_RF.V2 100	0.404	**0.398**	0.405	0.402	0.414
ASAP_RF.V2 200	0.404	0.402	0.402	**0.399**	0.405
ASAP_RF.V2 300	0.399	0.399	0.402	**0.393**	0.405
ASAP_RF.V2 500	0.398	**0.394**	0.398	0.398	0.401
ASAP_RF.V1	0.416				
ASAP_kNN.V1	0.423				
autofolio	0.391				
flexfolio	0.442				
sunny	0.482				
sunny-presolv	0.485				
zilla	0.366				
zillafolio	0.37				
autofolio-48			0.375		
llama-regrPairs			0.395		
llama-regr			0.425		

equivalent to the means over the columns of table 2

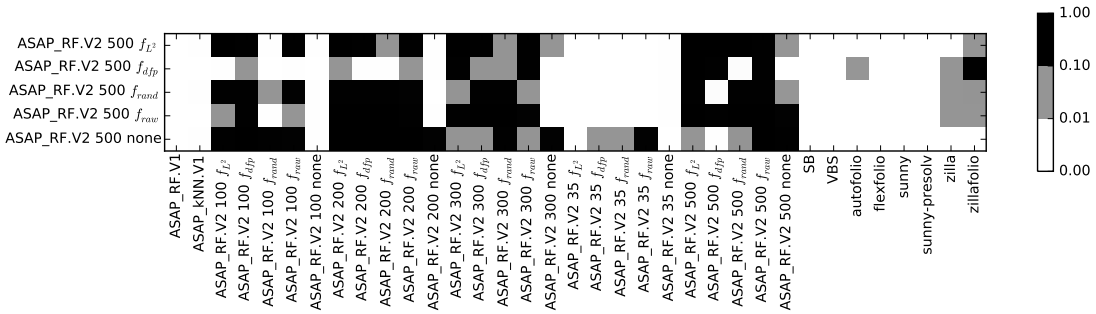


Fig. 5. Wilcoxon signed-rank test p-value between ASAP.V2 variants with 500 trees and every other systems including the single best algorithm (SB) and the virtual best solver (VBS). In particular, the different fitness functions do not differ significantly from each other in most cases.

Fig. 6 details per dataset the performance improvement between ASAP.V2 (500 trees, f_{L^2} version) and ASAP.V2 (500 trees, f_{dfp} version) and on the other hand ASAP.V1_RF (35 trees). Note that ASAP.V2 outperforms the per-dataset best submission to the challenge for 3 datasets.

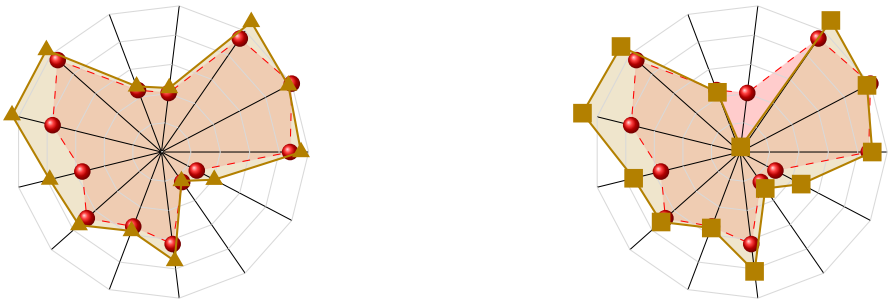


Fig. 6. Per-dataset performances of ASAP_RF.V2 with 500 trees, optimized using f_{L^2} (on the left with triangles) and using f_{dfp} (on the right with squares) scaled to the range of performance of the challenge submitted systems. As a comparison, ASAP_RF.V1 appears with the balls and dashed line.

6 Discussion

A new hybrid algorithm selection approach, the ASAP system was presented in this paper, combining a pre-scheduler and a per-instance algorithm selector. ASAP.V1 introduced a selector learned conditionally to a predetermined schedule so that it focuses on instances that were not solved by the pre-scheduler . ASAP.V2 completes the loop as it re-adapts the schedule to the new AS. The main message is that the scheduler and the AS must be optimized jointly to reflect the division of labor achieved by these two components

ASAP.V1, thoroughly evaluated in the ICON challenge on algorithm selection (ranked 4th) received an honourable mention, due to its novelty and good performance comparatively to the famed and long-known Zilla algorithms.

The ASAP.V2 extension achieved significantly better results along the same challenge setting. It must be emphasized that these results must be comforted by additional experiments on fresh data. A main lesson learned is the importance of the regularization, as the amount of available data does not permit to consider richer AS search spaces without incurring a high risk of overfitting. The probabilistic performance criterion successfully contributed to a more stable optimization problem. Further research work will be devoted to extending this criterion.

References

1. Bischl, B. et al.: ASlib: A Benchmark Library for Algorithm Selection. arXiv:1506.02465v1 (2015)
2. Branke, J., Deb, K., Dierolf, H., Osswald, M.: Finding knees in multi-objective optimization. In: Parallel Problem Solving from Nature-PPSN VIII. pp. 722–731. Springer (2004)
3. Das, I.: On characterizing the "knee" of the pareto curve based on normal-boundary intersection. Structural Optimization 18(2-3), 107–115 (1999)
4. Deb, K.: Multi-objective evolutionary algorithms: Introducing bias among pareto-optimal solutions. In: Advances in evolutionary computing, pp. 263–292. Springer
5. Gagliolo, M., Schmidhuber, J.: Algorithm portfolio selection as a bandit problem with unbounded losses. Annals of Mathematics and Artificial Intelligence 61(2), 49–86 (2011)
6. Gomes, C.P., Selman, B.: Algorithm portfolios. Artificial Intelligence 126(1), 43–62 (2001)
7. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized Evolution Strategy with Covariance Matrix Adaptation. Evolutionary Computation 11(1), 1–18 (2003)
8. Heidrich-Meisner, V., Igel, C.: Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In: Danyluk, A.P. et al. (ed.) ICML. pp. 401–408. ACM Intl Conf. Proc. 382 (2009)
9. Huberman, B.A., Lukose, R.M., Hogg, T.: An economics approach to hard computational problems. Science 275(5296), 51–54 (1997)
10. Kadioglu, S., Malitsky, Y., Sabharwal, A., Samulowitz, H., Sellmann, M.: Algorithm selection and scheduling. In: Proc. 17th CP, pp. 454–469. LNCS 6876, Springer (2011)
11. Kotthoff, L.: ICON challenge on algorithm selection. CoRR abs/1511.04326 (2015)
12. Leyton-Brown, K., Nudelman, E., Andrew, G., McFadden, J., Shoham, Y.: A portfolio approach to algorithm selection. In: Proc. IJCAI. pp. 1542–1543 (2003)
13. Malitsky, Y., Sabharwal, A., Samulowitz, H., Sellmann, M.: Algorithm portfolios based on cost-sensitive hierarchical clustering. In: Proc. 23rd IJCAI. pp. 608–614. AAAI Press (2013)
14. Mısır, M., Sebag, M.: Algorithm selection as a collaborative filtering problem. Tech. rep., INRIA (2013)
15. O´Mahony, E., Hebrard, E., Holland, A., Nugent, C., O´Sullivan, B.: Using case-based reasoning in an algorithm portfolio for constraint solving. In: Proc. ICAICS. pp. 210–216 (2008)
16. Oentaryo, R.J., Handoko, S.D., Lau, H.C.: Algorithm selection via ranking. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI) (2015)
17. Pedregosa, F. et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
18. Rice, J.R.: The algorithm selection problem. Advances in Computers 15, 65–118 (1976)
19. Stern, D., Herbrich, R., Graepel, T., Samulowitz, H., Pulina, L., Tacchella, A.: Collaborative expert portfolio management. In: Proc. 24th AAAI. pp. 179–184 (2010)
20. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)
21. Xu, L., Hutter, F., Hoos, H., Leyton-Brown, K.: Features for SAT (2012), university of British Columbia
22. Xu, L., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Satzilla: portfolio-based algorithm selection for sat. Journal of Artificial Intelligence Research pp. 565–606 (2008)
23. Xu, L., Hutter, F., Shen, J., Hoos, H.H., Leyton-Brown, K.: Satzilla2012: improved algorithm selection based on cost-sensitive classification models. Balint et al.(Balint et al., 2012a) pp. 57–58 (2012)
24. Yun, X., Epstein, S.L.: Learning algorithm portfolios for parallel execution. In: Hamadi, Y., Schoenauer, M. (eds.) Proc LION 6, pp. 323–338. LNCS 7219, Springer (2012)

A New Self-Regulated Harmony Search Algorithm for the Phase Equilibrium Description of a Supercritical Extraction System

Cristian Jiménez, Iván Amaya, Rodrigo Correa

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Universidad Industrial de Santander, Cra. 27 Cll. 9, Colombia.

Ph.D. student, cristian.jimenez@correo.uis.edu.co; Professor, ivan.amaya2@correo.uis.edu.co; Professor, crcorrea@uis.edu.co

Abstract

This article describes the ongoing work dealing with the prediction and estimation of vapor-liquid thermodynamic properties using global optimization algorithms. For the present case, phase equilibrium parameters for the system of supercritical CO_2 (sCO_2) and some essential oils, were estimated using the corrected version of van der Waals and Wong-Sandler mixing rules, the Peng-Robinson state equation and the more common thermodynamic models for non-linear parameter estimation in equilibrium modeling, namely, Van Laar, NRTL and UNIQUAC. We proposed a variant of the traditional harmony search algorithm, i.e. self-regulated harmony search (SFHS), which was used in this task. Here, we include preliminary simulation results for the system, sCO2: α-pineno using the Wong-Sandler rule and the van Laar model. Results show a good agreement between the experimental results reported in the literature, and the models predictions using the SFHS algorithm. Furthermore, SFHS seems to be a promising algorithm for processing phase equilibrium data.

Keywords: Optimization, Harmony search, property estimation.

1. Introduction

Supercritical fluid extraction (SFE) is now one of the well-known green technologies. Some industries are currently using it instead of traditional undesirable solvents. SFE advantages include high extraction rates, high selectivity and efficiency, as well as, solvent recycling. SFE uses the so called green solvents, such as water, and carbon dioxide. This last supercritical no polar fluid is currently used on a large scale in several areas, for example for coffee beans decaffeination, biodiesel production, chemical reactions, cholesterol extraction from butter, and essential oils production, [1]-[3]. Nevertheless, these processes need to be optimized because of their high operative cost and the requirement for high selectivity in the presence of several other components, and the extraction in the shortest time. Obviously, the optimum will depend on the purpose of the extraction. Among the factors needed to be optimized, are the thermodynamic parameters that control phase equilibrium. The vital information for designing extractors are the data equilibrium between liquid and supercritical gas, besides reliable data on mass transfer and hydrodynamics. Although there are limited alternatives for modelling the vapor-liquid equilibrium in a SFE process, the use of empirical equations of state is probably the most common approach. The Peng-Robinson state equation which is cubic in the volume is a clear example, as is illustrated next. This state equation is extensively used for thermodynamic properties calculations, and is given by,

$$Z = \frac{Pv}{RT} = \frac{v}{v-b} - \frac{\frac{a}{RT}+d-2\sqrt{\frac{ad}{RT}}}{(v+b)+\left[\frac{b}{v}(v-b)\right]} \quad (1)$$

where for pure components,

$$a = a(T_c)(1+\kappa)^2 \qquad a(T_c) = \frac{0.45724\,R^2T_c^2}{P_c} \qquad \kappa = 0.37464 + 1.54226\,\omega - 0.26992\,\omega^2$$

$$b = \frac{0.0778\,RT_c}{P_c} \qquad d = \frac{a(T_c)\kappa^2}{RT_c}$$

Because the SFE thermodynamic system is a mixture, we require the use of a mixing rule. In this case we selected the Wong-Sandler mixing rule, along with the Van Laar model for calculating the activity coefficient,

$$a = \frac{DQRT}{1-D} \qquad ; D = \sum_i x_i \frac{a_i}{b_i RT} - \left\{\frac{\sqrt{2}}{\ln(1+\sqrt{2})}\right\}\left\{\frac{G^E}{RT}\right\};\ \frac{G^E}{RT} = \sum_i x_i \ln\gamma_i$$

$$b = \frac{Q}{1-D} \qquad Q = \sum_i \sum_j x_i\,x_j \left[\frac{b_i+b_j}{2} - \frac{\sqrt{a_i a_j}}{RT}\left(1-k_{ij}\right)\right]$$

$$\ln\gamma_i = \frac{\sum_j z_j a_{ij}}{1-z_i}\left[1 - \frac{z_i \sum_j z_j a_{ij}}{z_i \sum_j z_j a_{ij} + (1-z_i)\sum_j z_j a_{ji}}\right]^2 \quad (2)$$

Now, for the formulation of the required objective function, we can apply either the ordinary least squares method (OLS) or the maximum likelihood criterion. The former strategy was used in this in-progress work. With the help of the above equation, the objective function (OF) was structured as,

$$OF = \sum_i^N \left[P_i^{exp} - P_i^{cal}\right]^2$$

where N is the number of experimental data, is the experimental value, and the corresponding pressure at the bubble point evaluated using the Peng-Robinson equation. For the simulation results presented below, the adjustable parameters were $(a_{12}; a_{21}; k_{12})$.

A. The SFHS algorithm

The Harmony Search (HS) algorithm was proposed a little bit over a decade ago in [4]. Throughout its existence different variants have appeared trying to improve multiple components of the strategy. A recently proposed modification was presented last year in [5]. This algorithm is similar to the original HS approach, and it was inspired on the success of the ABHS variants [6]. The authors propose a variation of the Fretwidth (FW(j)) on each iteration, based on the following scenarios: (1) Start with a fixed value, FWini, (2) If the best solution in HM is improved, look for good values around the current FW and, (3) After FWsat non-successful iterations, switch to an exponential decay. The first case requires no further comments. The third one is quite close to the one proposed in ABHS, so an in-depth explanation can be found in [5]. The remaining case (i.e. scenario two) is the core of our proposal and it is ruled by eq. (3). Here, a random number uniformly distributed between zero and one () is multiplied by a constant (), to stochastically adjust the Fretwidth (FW(j)) around the

value that leads to improving the solution (). The constant, , is included as a way of controlling the level of adjustment, and it was set to one for the current work. Moreover, represents the midpoint for random generation of new fretwidths. This stochastic behavior is maintained until the third scenario is repeated. The logic of the current proposal is summarized in Fig. 1.

$$FW(j) = A_j + (r_{FW} - 0.5) \cdot C_{FW} \cdot A_j \quad (3)$$

For this algorithm, the Harmony Memory Size (HMS) indicates how many solutions are stored and considered during the execution of the algorithm. The Harmony Memory Considering Rate (HMCR) and Pitch Adjusting Rate (PAR) are values that must be located between zero and one, since they relate to the probability of selecting a given path in our proposed algorithm. Based on previously reported recommendations we defined FWini=0.5, FWmax=2.0, FWmin=, FWsat=1000, and SatHS=10000. The remaining parameters, i.e. HMS, PAR, and HMCR, were defined for each simulation, [5,6].

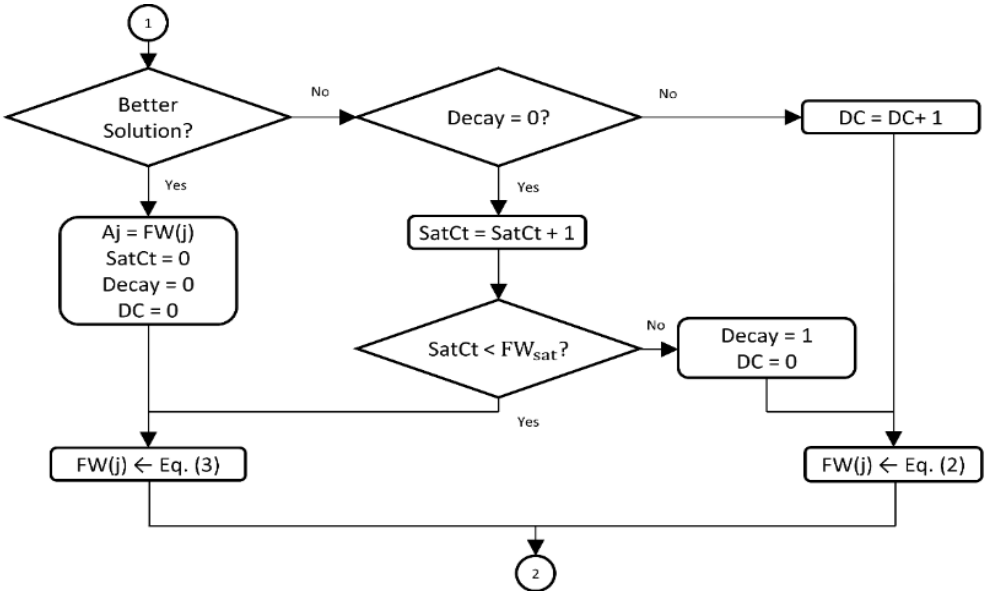


Fig 1. Logic of the SFHS variant, where DC is the decay counter, j is the current iteration, SatCt is the saturation counter, and FW_sat is the saturation limit, [5].

The overall logic of our proposed algorithm is described in the following pseudocode:

1) Define the execution parameters: memory size (HMS), memory considering rate (HMCR), pitch adjusting rate (PAR), initial Fretwidth (FW_{ini}), maximum Fretwidth (FW_{max}), amplitude constant (C_{FW}), and saturation limit (FW_{sat}). Also, define the saturation criterion to be used as a last resource for stopping the algorithm in case it does not converge after a given number of iterations (Sat_{HS}).
2) Generate a random initial matrix, HM, of size HMS × N, where N represents the number of dimensions.
3) Generate a random number. If it is lower than HMCR, go to step 4. Otherwise, take a random value from the search domain and go to step 6.

4) Pick the value located at a random row of HM, and at the column corresponding to the component which is being updated.
5) Generate a random number. If it is lower than PAR, adjust the pitch.
6) Repeat steps 3 to 5 for the remaining dimensions.
7) Evaluate the new candidate solution. If it is better than the worst solution stored in HM, replace it, discard the worst, and go to step 8. Otherwise, go to step 10.
8) Store the current Fretwidth, FW(j), into A_j. Also, reset the saturation counter (SatCt), the decay flag (Decay), and the decay counter (DC).
9) Update the Fretwidth and go to step 13.
10) Check if Decay=0. If so, increase the saturation counter,SatCt, and go to step 11. Otherwise, increase the decay counter, DC, and go to step 12.
11) Check if the saturation counter is smaller than the saturation limit, i.e. $\mathrm{SatCt} < \mathrm{FW_{sat}}$. If so, go to step 9. Otherwise, switch the decay flag to one, reset the decay counter (DC), and go to step 12.
12) Update the Fretwidth, and go to step 13.
13) Repeat for NI iterations.
14) Report results and end the process.

2. Methodology

The first part of this work was dedicated to verify the correctness of the SFHS algorithm. Later and after using it for calculating the optimum value for some benchmark functions, we ran simulations for the prediction of the thermodynamic parameters of the selected system sCO2:α-pineno. In order to generate the vapor-liquid equilibrium data, we use the state equation and the fugacity coefficients values for the binary mixture. Besides, the fundamental principle of equality of the fugacity coefficients (isofugacity condition) for a component present in both phases was used. Finally, we compiled for comparison purpose, the experimental data reported in [7]. Table I includes the critical properties of the two components of the mixture (sCO2: α-pineno).

Table I.
Critical properties of the mixture components, [7]

	T_c [K]	**P_c [MPa]**	**V_c[cm^3/mole]**	**Acentric factor**
CO_2	304.200	7.375	94.000	0.239
α-pineno	630.000	2.890	484.500	0.313

3. Results

The parameters for the SFHS algorithm were: FWini = 0.5, FWmax = 2.0, FWmin = , FWsat = 1000, and SatHS = 10000. Similarly, HMS = 5.0, PAR= 0.5, HMCR = 0.9. Fig. 2 shows an example of the variation of the iterations number, as well as of the convergence time, for the shifted Jong function when using SFHS with the above parameters.

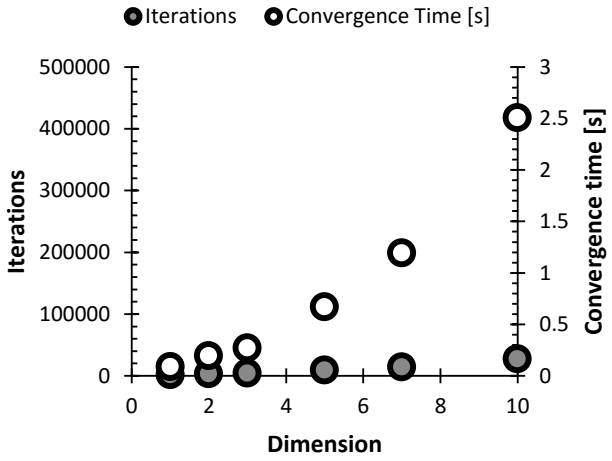


Fig.2. The iterations number and convergence time for the shifted Jong function.

As observed, the convergence time is highly dependent of the number of unknown parameters, but this is not the case for the iterations number. Some of the parameters for the SFHS algorithm used for most of the simulations were: FWMin = ; FWMax = 2.0; FWIni = 0.5; FWsat = 1000; SatHS=1e4; CBW = 1.0; δ≤; PAR=0.8; HMCR=0.9; HMS=0.5. The maximum number of iterations was one million. We repeated each of the simulations 30 times. A slight variation of the results was detected. We believe that one of the possible reasons for this behavior, is the apparent random nature of the optimization algorithm, as well as the error propagation. Table II includes the values of the interaction parameters.

TABLE II.

Averaged values for the interaction parameters

T, [K]	$a12$	$a21$	$k12$
313.15	1.2139	0.9580	0.7587
323.15	1.6946	0.7911	0.9841
328.15	0.9864	0.9189	0.7751

Fig. 3 shows the vapor (supercritical phase)-liquid equilibrium, for three temperatures. In it, the model prediction for the experimental values P-(x;y) was adjusted. The Peng-Robinson estate equation along with the Wong-Sandler mixing rule and the van Laar model, seems to be a useful path for modeling the vapor-liquid equilibrium at high pressures above critical conditions. However, this type of equation of state which includes mixing rules for supercritical fluid mixtures, becomes very sensitive to the value of the interaction parameters. Keep in mind, that there is some theoretical basis to the cubic equations of state, but this is does not hold for such adjustable parameters.

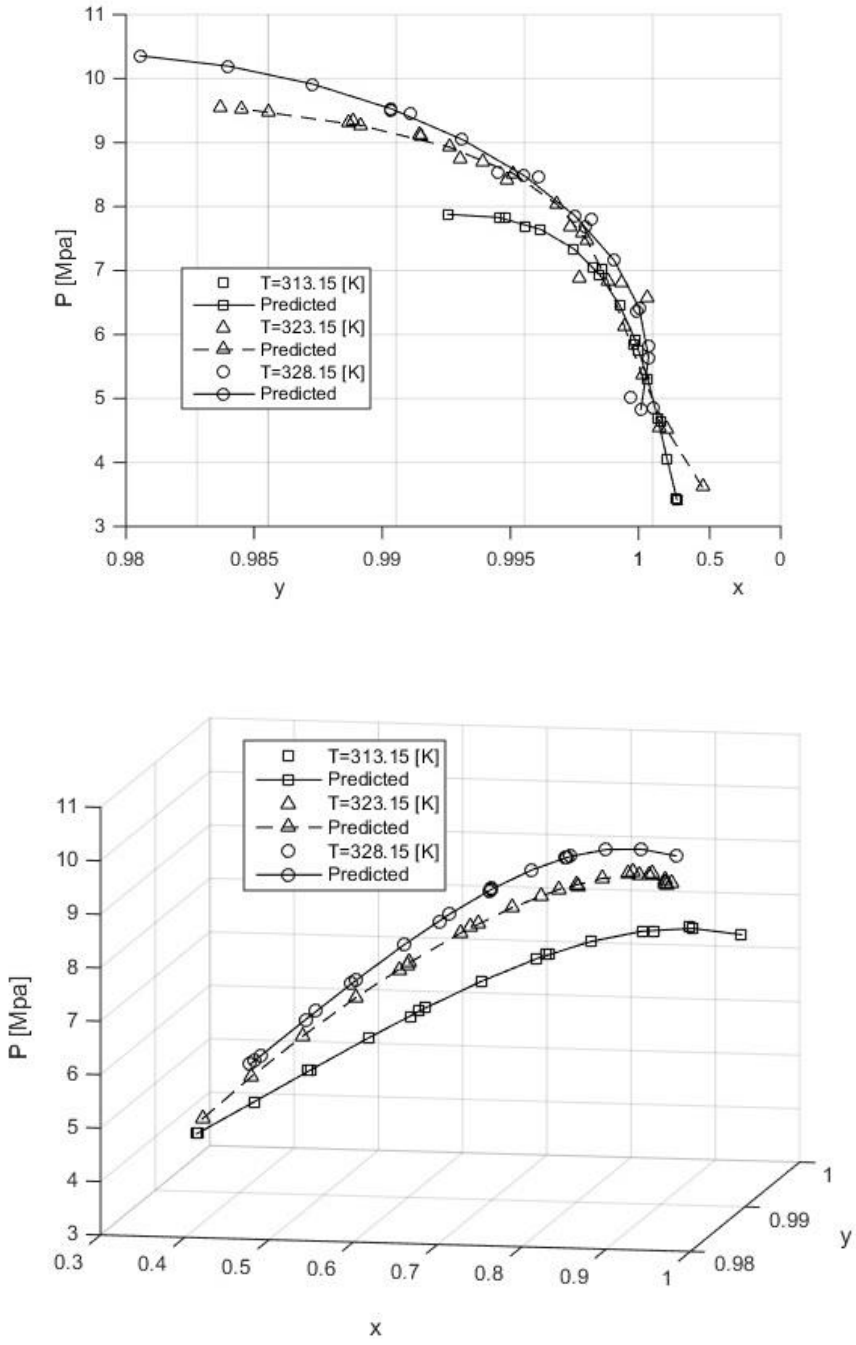


Fig 3. V-L equilibrium at three temperatures

In order to know the molar volume of the liquid phase for the mixture, we recalculated it using the Peng-Robinson state equation along with the parameters found by simulation. Figure 4 shows the expected tendency for the volume variation, as a function of the pressure and temperature increment.

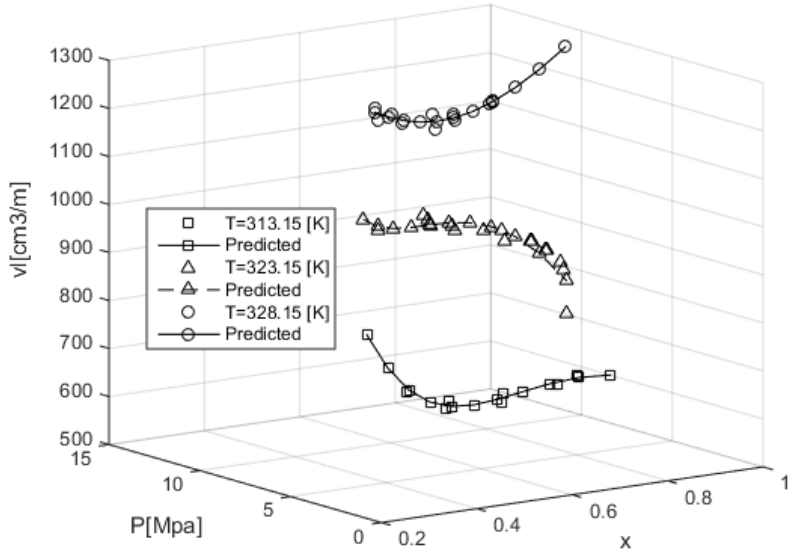


Fig 4. Liquid molar volume variation

4. CONCLUSIONS

We contrasted simulation results with previous experimental work reported in the literature for the system sCO2: α-pineno and we found a good agreement. Nevertheless, we observed a small variation of the interaction parameters values with reasonable and acceptable accuracy. Further work is needed for improving the simulation methodology (including new thermodynamic models, and the minimization of the Gibbs free energy), and on the development of approaches to enhance the reliability and reproducibility of this new variant SFHS, and other algorithms.

Nomenclature:
P = pressure in atm
v = molar volume in liters/g-mol
T = temperature in K
R = gas constant (0.08206 atm-liters/g-mol K)
Tc = critical temperature in K
Pc = critical pressure in atm
γ = activity coefficient
ω = acentric factor

References

1. Sodeifian G., Sajadian S., Ardestani N., “Extraction of dracocephalum kotschyi Boiss using supercritical carbon dioxide: experimental and optimization”, The Journal of Supercritical Fluids, Volume 107, January 2016, Pages 137–144. doi:10.1016/j.supflu.2015.09.005
2. Jokić, S., Bijuk, M., Aladić, K., Bilić, M., Molnar, M., “Optimisation of supercritical CO2 extraction of grape seed oil using response surface methodology”, International Journal of Food Science and Technology, Volume 51, Issue 2, 1 February 2016, Pages 403-410. doi: 10.1111/ijfs.12986
3. Rostamian, H., Lotfollahi, M.N., “New functionality for energy parameter of Redlich-Kwong equation of state for density calculation of pure carbon dioxide and ethane in liquid, vapor and supercritical phases”, Periodica Polytechnica: Chemical Engineering, Volume 60, Issue 2, 2016, Pages 93-97. doi: 10.3311/PPch.8221
4. Geem Z., J. Kim, G. Loganathan, “A new heuristic optimization algorithm: harmony search,” Simulation, nº 76, 2001, Pages 60-68. doi: 10.1177/003754970107600201
5. Amaya I., Cruz J., Correa R., “Harmony search algorithm: a variant with self-regulated fretwidth,” Appl. Math. Comput., nº 266, 2015, pages 1127-1152. doi:10.1016/j.amc.2015.06.040
6. Contreras J., Amaya I, Correa R., “An improved variant of the conventional harmony search algorithm,” Appl. Math. Comput., nº 227, 2014, Pages 824-830. doi:10.1016/j.amc.2013.11.050
7. Pavlicek J., M. Ritcher, “High pressure vapor-liquid equilibrium in carbon dioxide- ∝-pinine system”, Fluid Phase Equilibria, 90,1993, Pages125-133. Doi:10.1016/0378-3812(93)85007-9

Minimizing the makespan on a two machine flow-shop under preventive maintenance activities

I. Krimi[1,2], R. Benmansour[1], S. Hanafi[1], and N. Elhachemi[2]

[1] University of Valenciennes and Hainaut Cambrésis, LAMIH, UMR CNRS 8201, 59313 Valenciennes Cedex 9, France.
{Rachid.Benmansour,Said.Hanafi}@univ-valenciennes.fr
[2] Mohammadia School of Engineers, Rabat Maroc.
issamkrimi@research.emi.ac.ma
elhachemi@emi.ac.ma

1 Introduction

One of the most significant issue in manufacturing systems is production scheduling. Better scheduling system has significant impact on cost reduction, increased productivity, customer satisfaction and overall competitive advantage. The vast majority of scheduling research are based on the common assumption that the machines are available all the time. However, this assumption may not be true in real industry. In order to increase the productivity, machines on production lines should be maintained periodically in order to minimize the makespan. In this paper, we study a two-machine flow shop scheduling problem where machines must be maintained periodically. Although this problem has largely been addressed in the past, the periodic maintenance issue has not been taken sufficiently into account. In the flow-shop scheduling literature, there is a major job feature which is the preemption. In our work, we assume that preemption is not allowed. Liao *et al.* [3] introduced a new notion which is the unavailability constraint due to preventive maintenance after a fixed number of jobs. They provided a Mixed Integer Programming (MIP) model, two heuristics and also a Branch and Bound method for analyzing heuristic's performance. Nouri *et al.* [1] dealt with a non-permutation two-machine flow-shop with learning effect and availability constraints. They propose a MIP and an effective improving heuristic. Hnaien *et al.* [2] were interested by with the same problem but under the unavailability constraints considered only on one machine. They investigated two mixed-integer programming (MIP) models for this problem. Then they proposed a branch and bound algorithm based on a set of new lower bounds and heuristics.

2 MIP formulation

We consider a set of n jobs $N = \{1, 2, .., n\}$ to be executed on two machines represented by a set $K = \{1, 2\}$. First, we mention that each machine can perform only one job at time. Each job must follow the same route. It should be executed on the first machine then on the second one. Furthermore, we assume that the machines are subject to periodic maintenance of period T. Our approach consists on splitting the scheduling horizon into batches. Each one is defined as the period between two successive maintenance tasks. We mention that T represents also the length of each batch. Hereafter, we present the notation used to formulate our problem:

n	number of jobs
k	machine index, $k = 1, 2$
i	job index, $i = 1, .., n$
P_{ik}	processing time of job i on machine k
T	availability period
d	duration of maintenance task
l	batch index
M	big integer value

The variables C_{max}, C_{ik} represent, respectively, the makespan and the completion time of job i on the machine k.

$$x_{ijk} = \begin{cases} 1 \text{ if job } i \text{ precedes job } j \text{ on machine } k, \\ 0 \text{ otherwise.} \end{cases}$$

$$b_{ilk} = \begin{cases} 1 \text{ if job } i \text{ belongs to bach } l \text{ on machine } k, \\ 0 \text{ otherwise.} \end{cases}$$

$$z_{lk} = \begin{cases} 1 \text{ if batch } l \text{ is used on machine } k, \\ 0 \text{ otherwise.} \end{cases}$$

$$min\ C_{max} \tag{1}$$
$$s.c.\ C_{ik} \geq P_{ik} \quad \forall (i,k) \in N \times K; \tag{2}$$
$$C_{ik} \geq P_{ik} + C_{jk} - Mx_{ijk} \quad \forall (i,j) \in N^2, i \neq j, k \in K; \tag{3}$$
$$C_{jk} \geq P_{jk} + C_{ik} - M(1 - x_{ijk}) \quad \forall (i,j) \in N^2, i \neq j, k \in K; \tag{4}$$
$$C_{max} \geq C_{i2} \quad \forall i \in N; \tag{5}$$
$$C_{i2} - P_{i2} \geq C_{i1} \quad \forall i \in N; \tag{6}$$
$$\sum_{l=1}^{n} b_{ilk} = 1 \quad \forall (i,k) \in N \times K; \tag{7}$$
$$\sum_{i=1}^{n} P_{ik} b_{ilk} \leq T z_{lk} \quad \forall (l,k) \in L \times K; \tag{8}$$
$$C_{ik} - P_{ik} \geq (l-1)(T+d) b_{ilk} \quad \forall (i,l,k) \in N \times L \times K; \tag{9}$$
$$C_{ik} \leq (lT + (l-1)d) b_{ilk} + (1 - b_{ilk})M \quad \forall (i,l,k) \in N \times L \times K; \tag{10}$$
$$C_{ik} \geq 0 \quad \forall (i,k) \in N \times K; \tag{11}$$
$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in N^2; \tag{12}$$
$$b_{ilk} \in \{0,1\} \quad \forall (i,l,k) \in N \times L \times K; \tag{13}$$
$$z_{lk} \in \{0,1\} \quad \forall (l,k) \in L \times K; \tag{14}$$

The objective function (1) is to minimize makespan. Constraints (2) guarantee that completion times of the jobs must be higher than or equal to their corresponding processing times. Constraints (3)-(4) avoid the overlap of jobs. Constraints (5) specify that the makespan is equal to the completion time of the last job executed on machine 2. Constraints (6) guarantee that the first operation of a job should be executed on the first machine completely before its executing on the second machine. Constraints (7) assure that a job must be included in exactly one batch. This latter could be different for each machine. Constraints (8) assume that processing times of jobs in a batch do not exceed its length T. Constraints (9)-(10) define the starting and completion times for each job included in a batch. The completion time was defined as non-negative variables by constraint (11) and also constraints (12)-(13)-(14)set up the binary variables x_{ijk},b_{ilk},z_{lk}.
We obtain a MIP model for a permutation scheduling by adding the following constraints :
$x_{ij1} = x_{ij2} \quad \forall (i,j) \in N^2$

2.1 Lower bound

We present a lower bound for the problem based on Hnaien *et al.* [2]:

$$LB = \sum_{i=1}^{n} P_{i1} + \lfloor \frac{\sum_{i=1}^{n} P_{i1}}{T} - 1 \rfloor \cdot d + \max_{1 \leq i \leq n} \{P_{i2}\}. \tag{15}$$

3 Variable Neighborhood Search

Variable Neighborhood Search is a single-solution based meta-heuristic for solving optimization problems proposed by Mladenovic and Hansen [4]. The principal idea is to explore successfully a

set of different neighborhood in order to provide a better solution by moving from one neighborhood to another. The changes of neighborhood are used to escape from local optimum. We use three procedure to generate an initial solution: randomly, by applying SPT and LPT rules. The solution of a permutation schedule is represented by a sequence of jobs. With this representation, we can use three neighborhood structure presented as follows:

- Interchange N_1: an interchange neighbor is defined by exchanging two job positions.
- Swap N_2: a swap neighbor is defined by exchanging two successive job positions.
- Reverse N_3: a reverse neighbor is defined by reversing jobs order between two different positions.

The shaking procedure can be easily implemented using random moves from a given neighborhood and we mention also that we use a best improvement local search. The stopping criteria was fixed at 100 iterations.

4 Computational results

In order to assess the performance of the two approaches, several experiments were performed on 15 randomly generated instances for each n. The processing time were generated randomly in $[1, 100]$, the length of batches was $T = 100$ and the maintenance duration was $d = 1$. We used CPLEX 12.6 solver and the proposed VNS algorithm was coded in the C language. All instances were solved on an Intel Core i5 2.7 GHz and 8 Go of RAM.
Table 1 represents the results of the MIP. It is able to solve optimally instances of size up to 35 jobs.

Table 1. MIP average results

Jobs	Cpx_Opt	CPU_Time
10	488	2
15	950	15
20	968	386
25	1781	633
30	2035	1699
35	2448	1200

Table 2. VNS average results

Jobs	Gap_Cpx(%)	Gap_VNS(%)	CPU_time
10	0	0.04	3600
20	0	0.06	3600
30	0	0.79	3600
50	12.46	2.92	3600
100	20.78	10.34	3600
500	27.51	12.34	3600

Table 4 provides the results of VNS. We define the gap as follows: for $n > 30$: Gap_VNS= $\frac{Best\_VNS-LB}{LB}$. Otherwise Gap_VNS= $\frac{Best\_VNS-CPX\_Opt}{CPX\_Opt}$

5 Conclusion

In this work, we investigate a two machine flow-shop scheduling problem with a preventive maintenance. We propose a MIP formulation to solve optimally this problem for small instances and we present an efficient and effective VNS designed to solve the big instances.

References

1. B. Nouri, P. Fattahi, and R. Ramezanian. Minimizing total flow time for the non-permutation flow shop scheduling problem with learning effects and availability constraints. *Journal of Manufacturing Systems*, 2013.
2. F. Hnaien, F. Yalaoui, and A. Mhadhbi. Makespan minimization on a two-machine flowshop with an availability constraint on the first machine. *International Journal of Production Economics*, 2015.
3. L. Liao and C. Tsai. Heuristic algorithms for two-machine flowshop with availability constraints. *Computers & Industrial Engineering*, 2009.
4. N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 1997.

Determination of US Commercial Banks' Probability of Default by Structural Models with Subordinated VG Process

P. Gurný[1]

Departament of Finance, Economic Faculty, VSB_TU Ostrava, Czech Republic, Sokolska tr. 33, 701 21, Ostrava
petr.gurny@vsb.cz

Keywords : credit risk modelling; probability of default, subordinated Variance Gamma process.

1 Introduction

The internationalisation of financial markets has significantly expanded investment opportunities and risk. No doubt the probability of default (PD) is one of the key input factors of credit risk modelling and measuring. Estimating each borrower's risk level by assigning a different PD is now widely done by many banks (especially since 2008, when most of the world began to experience a period of financial and economic turmoil). The PD has many methodologies for its estimation, but this paper will be devoted to the structural approach, which was proposed in 1974 by Robert Merton [7] in his seminal paper on the valuation of corporate debt, where he introduced a model for assessing a credit risk of a company by characterizing a company's equity as a call option on its assets. It was natural to use the Black-Scholes [1] option pricing framework from 1973 for estimation the price of this option. The vast majority of previously-proposed empirical studies based on structural models were devoted to the nonfinancial institutions, mainly because financial institution defaults occur relatively scarcely. Nevertheless, the recent global financial crisis has clearly demonstrated the necessity for the correct determination of the probability of default also for financial institution. For this reason, the proposed paper is aimed to determine the probability of default of financial institutions, particularly US commercial banks.

Generally, the Merton model is built on the number of simplifying assumptions. Notwithstanding, one of the most important drawbacks is an assumption that company assets follow the log-normal distribution (the natural logarithm of returns on the company value is distributed normally). But it is well known that log-return of equities are not normal distributed. As a matter of fact, several empirical investigations have shown that log-return of equities present skew distributions with excess kurtosis which leads to a greater density in the tails and the normal distribution (se e.g. Fama [3, 4, and 5] or Rachev and Mittnik [8]. From this reason, there is presently a trend of analysing and simulating the time series of particular variables using Lévy processes (see e.g. Cont and Tankov [2], Schoutens [9 and 10] or Tichý [11]), because these series can fit also the higher moments of the probability distributions. This could be crucial moment within correct estimation of PD, especially for financial institution.

In this research the possibility of using the structural model based on Merton's framework within determination of US commercial banks' probability of default is verified. First, the classical assumptions of Merton's model are applied, particularly the assumption of normal distributions of assets' market value. Secondly, a structural credit risk model based on the Variance Gamma (VG) process is implemented in order to overcome some drawbacks of the Merton model, thus particularly the assumption of normal distribution. Finally, an empirical comparison between the results obtained from Merton model with classical assumptions and models with subordinated Lévy processes are proposed.

Our empirical analysis so far suggests, that normality assumption can lead to the serious mistakes in the pricing of the risk and that the probability of default is generally underestimated by the classical Merton

[1] The working paper is based on research activities sponsored through GACR Project No. P403/14-15175P and ESF Project No. CZ.1.07/2.3.00/20.0296. The support is greatly acknowledged.

model. These first results will be further discussed and compared with other selected models. Neural networks and genetic algorithms will be used at the following analysis, since the empirical analysis suggest an interesting possibility of using these models to estimate the probability of default.

2 Methodology

The core concept of the Merton model [7] is very well known and was introduced in 1974 to treat company's equity and debt as a contingent claim written on company's assets value. Using classical geometric Brownian motion and in accordance the Black-Scholes option pricing theory [1] we get PD estimation as follows:

$$PD_t = \mathrm{P}\left[A_T \le L\right] = \int_{-\infty}^{-\frac{\ln\left(\frac{A_t}{L}\right)+\left(\mu-\frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}} \phi(x)dx \tag{1}$$

where μ is the expected return (drift coefficient), σ is the volatility (diffusion coefficient), both unobserved, A_t is the company's asset value at time t, L is face value of debt at time T and ϕ is the probability density function of a standard normal variable.

Using subordinated processes we are usually able to capture empirically observed anomalies which are presented in the evolution of return processes over time. That is, we substitute the physical time with a so-called intrinsic (operational) time which provides distribution tail effects often observed in the market (see Rachev and Mittnik [8] or Schoutens [9]). Thus, if $W=\{W(t), t\ge 0\}$ is a stochastic process (we will suppose that W is a standard Brownian motion) and $T=\{T(t), t\ge 0\}$ is a non-negative stochastic process, a new process $Z=\{Z(t)=W(T(t)), t\ge 0\}$ may be formed and it is defined as subordinated to W by the intrinsic time process T. Thus, we model the company's assets value process A_t by a stochastic equation:

$$A(t)=A(t_0)\exp\left\{\int_{t_0}^{t}\mu(s)ds+\int_{t_0}^{t}\rho(s)dT(s)+\int_{t_0}^{t}\sigma(s)dW(T(s))\right\}, \tag{2}$$

where the drift in the physical time scale $\mu(s)$, the drift in the intrinsic time scale $\rho(s)$ and the volatility $\sigma(s)$ are generally assumed to be constant. Following the same notation as in the Merton's framework, the value of a European call option at time t (the value of company's equity) with exercise price L (face value of a zero-coupon debt instrument) and time to maturity t [2] is given by

$$E_t = A(t_0)F_+\left(\ln\left(\frac{A(t_0)}{L_{r,t_0,t}}\right)\right) - L_{r,t_0,t}F_-\left(\ln\left(\frac{A(t_0)}{L_{r,t_0,t}}\right)\right), \tag{3}$$

where

$$F_\pm(x)=\int_0^{+\infty}\Phi\left(\frac{x\pm\frac{1}{2}y}{\sqrt{y}}\right)dF_Y(y), \tag{4}$$

$\Phi(\cdot)$ is the cumulative distribution function of the standard normal variable, F_Y is the cumulative distribution function of a random variable $Y=\int_{t_0}^{t}\sigma^2(s)dT(s)$ and $L_{r,t_0,t}=L\exp\left(-\int_{t_0}^{t}r(s)ds\right)$ is the discounted exercise price (time-dependent function $r(t)$ defines the short term interest rate). Considering a continuous distribution of the random variable Y with density function f_Y then $F_\pm(x)$ can now be

[2] Here we change the notation of maturity time from *T* (used in the Merton's framework) to *t* since *T* denotes the intrinsic time process in the subordinated option pricing models.

numerically integrated over the finite interval $[0,1]$ the transformation $y = u(1-u)^{-3}$ (see Rachev and Mittnik [8]), i.e.:

$$F_{\pm}(x) = \int_0^{+\infty} \Phi\left(\frac{x \pm \frac{1}{2}\lambda y}{\sqrt{\lambda y}} \right) f_Y(y)\,dy = \int_0^1 \Phi\left(\frac{x \pm \frac{1}{2}\lambda u(1-u)^{-3}}{\sqrt{\lambda u(1-u)^{-3}}} \right) f_Y\left(u(1-u)^{-3}\right) \frac{1+2u}{(1-u)^4}\,du \ .$$

Analogously to the Merton model (using *delta* Greeks for monitoring of the variation in the derivative price with respect to the parameters that enter into the option formula) the probability of default can be finally estimated under the risk neutral probability measure as follows:

$$PD_t = F_+\left(\ln\left(\frac{L_{r,t_0,t}}{A(t_0)} \right) \right) = \int_0^{+\infty} \Phi\left(\frac{\ln\left(\frac{L_{r,t_0,t}}{A(t_0)} \right) + \frac{1}{2}y}{\sqrt{y}} \right) dF_Y(y)\,. \tag{5}$$

Gurný, Ortobelli and Giacometti [6] used this approach for PD estimation of chosen non-financial companies. They worked with Stable Lévy model represented by Mandelbrot-Taylor distribution and they also suggested alternative parameter estimation for subordinated processes. In this research, described approach is used for the subordinated VG process.

References

[1] Black, F., Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81, 637-659.

[2] Cont, R., Tankov, P. (2004). *Financial Modelling with Jump Processes*. Chapman & Hall/CRC press.

[3] Fama, E. (1963). Mandelbrot and the Stable Paretian Hypothesis. *Journal of Business 36*, 420-429.

[4] Fama, E. (1965). The Behaviour of Stock Market Prices. *Journal of Business 38*, 34-105.

[5] Fama, E. (1965). Portfolio Analysis in a Stable Paretian Market. *Management Science 11*, 404-419.

[6] Gurný, M. Ortobelli, S. L., Giacometti, R. (2013). Structural Credit Risk Models with Subordinated Processes, *Working paper, University of Bergamo*.

[7] Merton, R. C. (1974). On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. *Journal of Finance*, 29, 449-470.

[8] Rachev, S., Mittnik, S. (2000). Stable Paretian Model in Finance. Wiley, Chichester.

[9] Schoutens, W. (2003). *Lévy Processes in Finance: Pricing Financial Derivates*. Wiley.

[10] Schoutens, W. (2009). *Lévy Processes in Credit risk*. Wiley.

[11] Tichý, T. (2011). *Lévy process in Finance: Selected Application with Theoretical Backround*. Series on Advanced Economic Issues, Faculty of Economics, VŠB-TU Ostrava.

Portfolio optimization problem with cardinality constraints

A. Kresta[1] and T. Tichý[2]

1. VSB-TU Ostrava, Sokolska tr. 33, 702 00 Ostrava
ales.kresta@vsb.cz

2. VSB-TU Ostrava, Sokolska tr. 33, 702 00 Ostrava
tomas.tichy@vsb.cz

Keywords : portfolio optimization, cardinality constraints, binary particle swarm optimization.

1 Introduction

Portfolio optimization consists of a portfolio selection problem in which we want to find the optimum way of investing a particular amount of the money in a given set of securities or assets, see Fernandez and Gomez (2007). While without any information about the investor's risk attitude we are not able to find one optimal portfolio, we are looking for the Pareto efficient frontier. This frontier contains all possible portfolios for which we are not able to lower the risk with the same level of the expected return or to heighten the expected return without increasing the risk.

The model in which we consider simple portfolio optimization without any limitation on the maximum number of the different assets we want to hold will be referred as general portfolio optimization problem,

$$w = \begin{cases} \arg\min\limits_{x} k \cdot \rho\left(\mathbf{R} \times x\right) - \left(1-k\right) \cdot E\left(\mathbf{R} \times x\right) \\ \sum\limits_{i=1}^{n} x_i = 1 \\ x_i \geq 0, \quad i = 1,...,n \end{cases} \tag{1}$$

where w is the vector of optimal weights, ρ is chosen risk measure and $E\left(\mathbf{R} \times x\right)$ is portfolio expected return and k is the parameter which specifies the investor's risk aversion. Matrix $\mathbf{R}$ represents random returns from particular assets (investment possibilities).

As any other model even the unconstrained portfolio optimization model has many premises, which make it very simplified. On the one hand it allows us to find the solution easily; on the other hand the model could not be utilized for a real world application. In the real world application we want to limit the number of assets we invest in. Introducing m as the desired number of different assets in the portfolio, we can extend the previous model to the cardinality constrained case. The model in which we limit the number of different assets held will be referred as cardinality constrained portfolio optimization problem,

$$w = \begin{cases} \arg\min\limits_{x} k \cdot \rho\left(\mathbf{R} \times x\right) - \left(1-k\right) \cdot E\left(\mathbf{R} \times x\right) \\ \sum\limits_{i=1}^{n} x_i = 1 \\ \sum\limits_{i=1}^{n} I_{x_i > 0} = m \\ x_i \geq 0, \quad i = 1,...,n \end{cases} \tag{2}$$

where I_x is logical variable taking value of 1 when x is true and 0 otherwise.

There are also different risk measures ρ which can be applied, such as variance, which leads to Markowitz (1952) mean-variance quadratic programming problem, Value at Risk (VaR), Conditional Value at Risk (CVaR) and others. In our work we focus on CVaR, which is defined as mean loss if it exceeds VaR, see e.g. Rockafellar and Uryasev (2002). Assuming a random-variable profit *X*, with known cumulative distribution function F_X, VaR can be defined as follows,

$$VaR_{R,\alpha} = -\inf\left\{x \in R : F_X\left(x\right) \geq \alpha\right\}, \tag{3}$$

and the definition of CVaR is following:

$$CVaR_{X,\alpha} = -\mathrm{E}\left[x \mid x < -VaR_{X,\alpha}\right]. \tag{4}$$

Having the vector of (future) possible profits *X*, each with the equal probability, CVaR at probability level α can be estimated as follows,

$$CVaR_{X,\alpha} = -\frac{1}{\alpha}\left(\frac{1}{n}\sum_{a=1}^{[\alpha n]-1} X_a + \left(\alpha - \frac{[\alpha n]-1}{n}\right)X_{[\alpha n]}\right), \tag{5}$$

where [*x*] stands for the smallest integer larger than *x* and *n* is the quantity of data utilized for CVaR calculation. In this case the probability of occurrence is the same for each element of X.

It is obvious that optimization problem (2) in case of variance as a risk measure represents a mixed-integer nonlinear (quadratic) programming problem for which the computationally effective algorithm does not exist, see Chang et al. (2002). In this work we apply the binary particle swarm optimization (BPSO) method and other heuristic methods in order to solve the problem (2) for variance and CVaR as a risk measure and we compare the obtained results.

The original particle swarm algorithm was introduced and discussed in Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995). It imitates birds flocking and fish schooling as it is searching in D-dimensional real numbers space for the best position. In this algorithm the certain number of particles is utilized, each particle's position representing solution of the problem. Particles move across the search space partially randomly and partially in the dependence of the personal and global best position discovered so far. Objective function imitates the space richness for food. In Kennedy and Eberhart (1997) the particle swarm algorithm was modified to operate on a binary variables and binary version of the particle swarm algorithm was introduced.

References

[1] FERNANDEZ, A.; GOMEZ, S. (2007). Portfolio selection using neural networks. *Computers and Operations Research* 34(4), 1177-1191.

[2] MARKOWITZ, H. M. (1952). Portfolio selection. *Journal of Finance* 7(1), 77–91.

[3] ROCKAFELLAR, R. T., URYASEV, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance* 26(7): 1443–1471.

[4] CHANG, T. J., et al. (2002). Heuristics for cardinality constrained portfolio optimisation. *Computers and Operations Research*, 27 (13), 1271-1302.

[5] EBERHART, R.; KENNEDY, J. (1995). New optimizer using particle swarm theory. In *Proceedings of the International Symposium on Micro Machine and Human Science*

[6] KENNEDY, J.; EBERHART, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks - Conference Proceedings*.

[7] KENNEDY, J.; EBERHART, R. (1997). A discrete binary version of the particle swarm algorithm. In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*.

Selected algorithms for portfolio optimization problems

A. Kresta[1], Sergio Ortobelli[1], Noureddine Kouaissah[1] and T. Tichý[1]

1. VSB-TU Ostrava, Sokolska tr. 33, 702 00 Ostrava
tomas.tichy@vsb.cz

Keywords: portfolio optimization, risk measures, dependency measures, large scale portfolio problem, optimization algorithm

In this work potential usage and impact of various algorithms of metaheuristic specification for portfolio optimization are studied. In particular, we focus on problems, in which different risk and dependency measures are involved. First of all, we use semidefinite positive correlation measures consistent with the choices of risk-averse investors and propose new portfolio selection models that optimize the relation between the portfolio and one or two market benchmarks. We also consider the large scale portfolio problems. Obviously, particular approaches are evaluated using real date from various markets.

References

Barak, S., Dahooie, J.H., Tichý, T. Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick. *Expert Systems with Applications* **42** (23): 9221-9235, 2015.

Ortobelli, S., Lando, T., Petronio, F., Tichý, T. Asymptotic Multivariate Dominance: a Financial Application. *Working paper submitted to Methodology and Computing in Applied Probability*.

Ortobelli, S., Lando, T., Petronio, F., Tichý, T. Asymptotic stochastic dominance rules for sums of i.i.d. random variables. *Journal of Computational and Applied Mathematics* 300, 432−448, 2016.

Ortobelli, S., Cassader, M., Vitali, S., Tichý, T. Portfolio selection strategy for the fixed income markets with immunization on average. *Annals of Operations Research.* To appear in 2017.

Ortobelli, S., Tichý, T. On the impact of semidenite positive correlation measures in portfolio theory. *Annals of Operations Research* 235: 625−652, 2015. ISSN 0254-5330. 10.1007/s10479-015-1962-x.

Tichý, T. *Lévy Processes in Finance: Selected applications with theoretical background*. SAEI, vol. 9. Ostrava: VŠB-TU Ostrava, 2011.

A trajectory-based metaheuristic for the Electric Vehicle Routing Problem

R. Linfati[1] and J.W. Escobar[2]

1. Universidad del Bio-Bio, Concepcion, CHILE
rlinfati@ubiobio.cl
2. Universidad del Valle, COLOMBIA
john.wilmer.escobar@correounivalle.edu.co

The Electric Multi-Depot Vehicle Routing Problem (eMDVRP) is a variant of the classical MDVRP problem, with additional constraints related to the use of electric vehicles; the vehicle can travel a limited distance from the depot, but the vehicle can go to a "recharge point" where it can be recharged (total or partially) to increase its maximal distance. The classic neighborhood (example, 2-opt) easily leads to an infeasible solution, and then we will focus on developing a new set of good neighborhoods for this problem. Efficient neighborhoods are presented, and used in trajectory-based metaheuristics. Additionally, to verify the correct implementation of the algorithms are considered instances from the VRP and MDVRP, comparing the solution quality and execution time.

Keywords: vehicle routing problem, electric vehicles, tabu search, variable neighborhood search.

1 Introduction

Due to new regulations and further technological progress in the field of electric vehicles, the research community faces the new challenge of incorporating the electric energy based restrictions into vehicle routing problems [1], examples are the minimization of emissions [2], energy consumption [3], reverse logistics and others [4], [5]. The advances in the use of electric energy for motors defines a new generation of problems related to the battery of the vehicles [6], [7], recharge system [8], electric vehicles battery swap stations location routing problem (BSS–EV–LRP) [10], hybrid of traditional and electric vehicles, etc. [7] presents key contributions of combinatorial optimization for an efficient electric vehicles management, energy shortest path problem, energy vehicle routing problem, facility location, and electric vehicles redistribution.

Historically, however, six major barriers to alternative fuel vehicles success have arisen: limited numbers of refueling stations, high refueling costs, onboard fuel-storage issues (i.e., limited range), safety and liability concerns, improvements in the competition (i.e., more efficient combustion engines), and high initial costs for consumers [9]. The logistics giant DHL announced a new pilot project which would introduce electrified delivery vans for its vehicle fleet in US and Germany [10]. In 2013, UPS had more than 100 electric vehicles in its fleet operating in US [10]. Companies like Amazon are evaluating the use of drones for delivery [11].

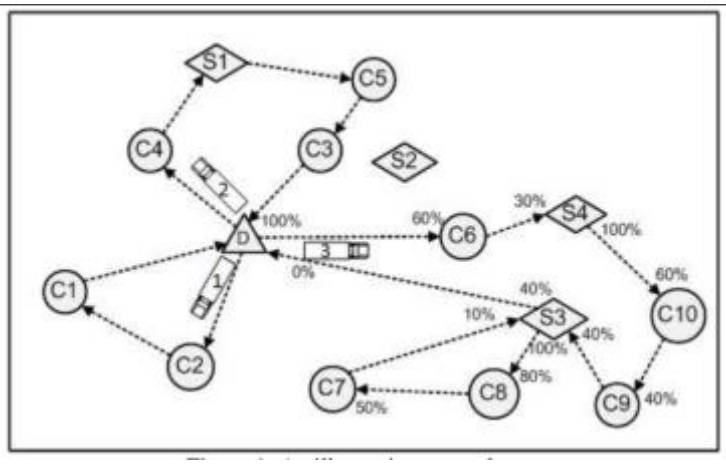


Figure 1: Example of eVRP, source: [54]

In the literature, there exists research to obtain a solution for multiple techniques that can be classified into two main groups. The exact algorithms are procedures that can obtain the optimal solution (when the upper bound is equal to the lower bound) but needs a lot of computation time, e.g., dynamic programing [12], branch and cut [13], branch, cut, and price [14], network flow formulation [15], and integer linear programing [16]. Heuristics algorithms (and metaheuristics) are procedures than can obtain a solution of good quality, without guaranteeing optimality in a short computing time, e.g., saving algorithm [17], cluster first–route second [18], tabu search [19], probabilistic change of neighborhood [20], granular tabu search [21], very large neighborhood [22], evolutionary algorithm [23], ant colony [24], and memetic Algorithm [25]. Surveys of techniques of solution can be found in [26] and [27]. The granular tabu search was originally proposed by [28] in the context of a tabu search (TS) for the CVRP. Granular neighborhoods are defined by means of a restricted set of arcs, which is determined by restricting the neighborhoods to include only elements which are likely to be part of high-quality solutions. For the CVRP, [28] showed that their granular TS (GTS) is able to strongly reduce run-time while nearly keeping solution quality. Subsequently, granular neighborhoods have been successfully used in solution methods addressing the following routing problems: a dynamic VRP with soft time windows, the split delivery VRP, the multi-depot VRP, the capacitated location-routing problem, the dial-a-ride problem and the team orienteering problem with time windows [29].

2 The Electric Multi-Depot Vehicle Routing Problem (eMDVRP)

This problem is a variant of the classical MDVRP problem, with additional constraints related to the use of electric vehicles; the vehicle can travel a limited distance from the depot, but the vehicle can go to a "recharge point" where it can be recharged (total or partially) to increase its maximal distance. This problem can be evaluated by the classical objective function of the minimal total distance, or by a bi-objective perspective. The dataset of a similar problem can be adapted to this problem and published online to compare performances with other research works. The eMDVRP is variant of the classic MDVRP, but the classic neighborhood (example, 2-opt) easily leads to an infeasible solution, and then we will focus on developing a new set of good neighborhoods for this problem.

The hypothesis is: A granular neighborhood can be designed that exploits the structure of the eMDVRP, and be used in trajectory-based metaheuristics.

a. Granular search space

The proposed approach use the granular search space proposed in [28], which is based on the utilization of a sparse graph containing the edges incident to the depots, the edges belonging to the best solutions found so far, and the edges whose cost is smaller than a granularity threshold$\Theta = \beta \bar{z}$, where $\bar{z}$ is the average cost of the edges in the best solution found so far, and β is a sparsification factor which is dynamically updated during the search.

b. Neighbourhood structures

The proposed approaches allow infeasible solutions with respect to the depot and the vehicle capacities. Given a feasible solution S, we assign to its objective function $F_1(S)$ a value equal to the sum of the traveling costs of the edges belonging to the routes traversed by S. In addition, for any solution S infeasible with respect to the depot capacity, we add to $F_1(S)$ a penalty term obtained by multiplying the over depot capacity by a dynamically changing penalty factor. A similar approach is used to calculate the objective function value of any solution S infeasible with respect to the route capacity (for further details see [19]).

c. Computational results

In this paper we present the design and implement of an efficient framework for the eMDVRP. The performance of the proposed algorithms have been evaluated by considering benchmark instances adapted from the literature. Additionally, to verify the correct implementation of the algorithms (efficiently and effectively) is considered the case when the vehicle starts with infinite energy from the depot, transforming the problem in the classical MDVRP; also considered the case when the amount of deposits is only one, that is, the classical VRP. A comparison with a set of classic benchmark instances is presented for these problems, comparing the solution quality and execution time.

Acknowledge

This work has been partially supported by Universidad del Valle from Colombia; and by Universidad del Bío-Bío from Chile. In addition, this work has been partially supported by FONDECYT by grant 11150370. This support is gratefully acknowledged.

References

- [1] Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). The Vehicle Routing Problem: Latest Advances and New Challenges: latest advances and new challenges (Vol. 43). Springer Science & Business Media.
- [2] Figliozzi, M. (2010). Vehicle routing problem for emissions minimization. Transportation Research Record: Journal of the Transportation Research Board, 2197(1), 1-7.
- [3] Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 48(1), 100-114.
- [4] Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: Past and future trends. Expert Systems with Applications, 41(4), 1118-1138.
- [5] Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. Transportation Research Part B: Methodological, 45(8), 1232-1250.
- [6] Artmeier, A., Haselmayr, J., Leucker, M., & Sachenbacher, M. (2010, May). The optimal routing problem in the context of battery-powered electric vehicles. In

Workshop CROCS at CPAIOR-10, 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability.
- [7] Touati-Moungla, N., & Jost, V. (2012). Combinatorial optimization for electric vehicles management. Journal of Energy and Power Engineering, 6(5), 738-743.
- [8] Felipe, Á., Ortuño, M. T., Righini, G., & Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transportation Research Part E: Logistics and Transportation Review, 71, 111-128.
- [10] Yang, J., & Sun, H. (2015). Battery swap station location-routing problem with capacitated electric vehicles. Computers & Operations Research, 55, 217-232.
- [9] Romm, J. (2006). The car and fuel of the future. Energy Policy, 34(17), 2609-2614.
- [11] D'Andrea, R. (2014). Guest Editorial Can Drones Deliver?. Automation Science and Engineering, IEEE Transactions on, 11(3), 647-648.
- [12] Christofides, N., Mingozzi, A., & Toth, P. (1981). State-space relaxation procedures for the computation of bounds to routing problems. Networks, 11(2), 145-164.
- [13] Laporte, G., Nobert, Y., & Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. Operations research, 33(5), 1050-1073.
- [14] Fukasawa, R., Lysgaard, J., de Aragão, M. P., Reis, M., Uchoa, E., & Werneck, R. F. (2004). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. In Integer programming and combinatorial optimization (pp. 1-15). Springer Berlin Heidelberg.
- [15] Two-commodity network flow formulation
- [16] Toth, P., & Tramontani, A. (2008). An integer linear programming local search for capacitated vehicle routing problems. In The Vehicle Routing Problem: Latest Advances and New Challenges (pp. 275-295). Springer US.
- [17] Clarke, G. U., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations research, 12(4), 568-581.
- [18] Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. Networks, 11(2), 109-124.
- [19] Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. Management science, 40(10), 1276-1290.
- [20] Rochat, Y., & Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. Journal of heuristics, 1(1), 147-167.
- [21] Toth, P., & Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. Informs Journal on computing, 15(4), 333-346.
- [22] Ergun, Ö., Orlin, J. B., & Steele-Feldman, A. (2006). Creating very large scale neighborhoods out of smaller ones by compounding moves. Journal of Heuristics, 12(1-2), 115-140.
- [23] Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31(12), 1985-2002.
- [24] Reimann, M., Doerner, K., & Hartl, R. F. (2004). D-ants: Savings based ants divide and conquer the vehicle routing problem. Computers & Operations Research, 31(4), 563-591.
- [25] Kalmbach, A. (2002). Evolutionary Algorithms in Vehicle Routing. Natural computing, 1(1), 3-52.
- [26] Doerner, K. F., & Schmid, V. (2010). Survey: matheuristics for rich vehicle routing problems (pp. 206-221). Springer Berlin Heidelberg.
- [27] Anbuudayasankar, S. P., Ganesh, K., & Mohapatra, S. (2014). Survey of Methodologies for TSP and VRP. In Models for Practical Routing Problems in Logistics (pp. 11-42). Springer International Publishing.
- [28] Toth, P., & Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. Informs Journal on computing, 15(4), 333-346.
- [29] Schneider, M., Schwahn, F., Vigo, D. Designing Granular Solution Methods for Routing Problems with Time Windows.

On Solving the Quadratic Three-Dimensional Assignment Problem with a PSO Algorithm

I.Ait Abderrahim

LAPECI laboratory, Computer science department, University of Oran 1 Ahmed Ben bella, Algeria
aitabderrahim_i@hotmail.fr

Keywords: Quadratic Assignment Problem (QAP), Quadratic Three-Dimensional Assignment Problem (Q3AP), Particle Swarm Optimization (PSO).

1 Introduction

The quadratic three-dimensional assignment problem (Q3AP) is a generalization of the well-known quadratic assignment problem (QAP). It is one of the hardest permutation problems and has recently gained interest for its application in the hybrid automatic repeat request (Hybrid ARQ) protocol in wireless communication systems and whose solution can significantly increase throughput and reduce the cost for providing reliable digital transmission over noisy fading channels. For optimizing a Hybrid ARQ protocol, solving the Q3AP consists of finding an optimal symbol mapping over two vectors. Whereas, a single vector protocol can be solved with the QAP. See [5]

The Quadratic Assignment Problem (QAP) is one in which *N* units have to be assigned to *N* sites in such a way that the cost of the assignment, depending on the distances between the sites and the flows between the units, is minimal.

The QAP can be formulated as follows: Given two *N* X *N* matrices, $F=[f_{ik}]$ where f_{ik} is the flow between units i and k; and $D=[d_{jn}]$ where d_{jn} the distance between sites j and n. The problem consists of finding a permutation p over the set S= {1, 2,..., *N*} that minimizes the following cost function:

$$\text{Cost } (p) = \sum_{i=1,\ldots,N} \sum_{k=1,\ldots,N} f_{ik} d_{p(i)p(k)}.$$

The QAP is one of the most difficult NP-hard combinatorial optimization problems. Solving large size instances of the QAP is still computationally challenging [1].

2 The Quadratic Three-Dimensional Assignment Problem

Pierskalla [2] introduced the Quadratic 3-dimensional Assignment Problem (Q3AP) in a technical memorandum. Since then, little on the subject has appeared.

Hahn et al. [3] re-discovered the Q3AP while working on a problem arising in data transmission system design. The problem consists of finding bit-to-symbol mappings for two (re)transmissions in the Hybrid Automatic Repeat Request protocol (HARQ) that minimizes the BER (Bit Error Rate) of the two transmissions. According to the explanation given in [4] and [3], in data transmission systems, when a data packet is sent through variably fading channels, some errors may remain in some bits (i.e., binary digits) of the received message, even after error correction mechanisms are used. The HARQ mechanism consists of sending to the transmitter a retransmission request for the same packet.

The Q3AP is an extension of two NP-hard problems, the QAP and the 3-index assignment problem (3AP). Thus it is easy to see that the Q3AP is also NP-hard.

If we use the facility location example, a Q3AP problem can be seen as a facility to location to managers assignment. Indeed, as illustrated in:

$$min \left\{ \begin{array}{l} \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{p=1}^{N} b_{ijp}\, x_{ijp} \\ + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{p=1}^{N}\sum_{k=1}^{N}\sum_{n=1}^{N}\sum_{q=1}^{N} C_{ijpknq}\, x_{ijp}\, x_{knq} \end{array} \right\} \quad (1)$$

$$x \in I \cap J \cap P,\ x = 0, 1 \quad (2)$$

With: I, J, P sets of the same cardinality N, and:

$$I = \left\{ x \geq 0 \sum_{j=1}^{N}\sum_{p=1}^{N} x_{ijp} = 1\ for\ i = 1, ..., N \right\} \quad (3)$$

$$J = \left\{ x \geq 0 \sum_{i=1}^{N}\sum_{p=1}^{N} x_{ijp} = 1\ for\ j = 1, ..., N \right\} \quad (4)$$

$$P = \left\{ x \geq 0 \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ijp} = 1\ for\ p = 1, ..., N \right\} \quad (5)$$

Equation 1, Q3AP is formulated in [3] as an extension of the formulation of the QAP. If we consider a diversity mapping problem in which a mapping should be used between *N* modulation symbols and *N* segments of the message to send, the cost coefficients C_{ijpknq} in Equation 1 refer to the simultaneous assignment of the symbols ***j*** and ***n*** to the bit strings ***i*** and ***k*** in the first transmission and the assignment of symbols ***p*** and ***q*** in the second retransmission. Equations 3, 4 and 5 express the uniqueness constraints of the two assignments (one symbol to one segment and vice versa in the two bit-to-symbol mappings)

In [3], artificial Q3AP instances are obtained from QAP using Equation 6. F and D are the two matrices representing the flow and the distances between factories/locations in QAP instances.

$$C_{ijpknq} = F_{ik} \star D_{jn} \star F_{ik} \star D_{pq}; (i, j, p, k, n, q = 1..N) \quad (6)$$

If we consider the assignment of *N* message segments to *N* adjacent modulation symbols during two transmissions, a solution for the Q3AP can be represented by two permutations π and ψ of numbers in the set {1,2, ..., *N*} referring to the labels of the *N* message segments to send during the two transmissions. The permutation formulation of Q3AP is derived from the permutation formulation of QAP (see Equation 7). ***π(i)*** and ***ψ(j)*** respectively indicate the bit string assigned to the modulation symbol ***i*** and ***j*** in the first and the second transmissions.

$$min f(\pi, \psi) = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{i\pi(i)\psi(j)j\pi(i)\psi(j)} \quad (7)$$

If among exact algorithms, the branch-and-bound method is the most successful one, the lack of a sharp lower bound technique in this algorithm is one of the major difficulties. The fact that the Q3AP is NP-hard is not sufficient to explain its difficulty, as we can now solve exactly very large instances of a great number of NP-hard problems. The homogeneity of the values of the solutions for most of the applications, due to the structure of the problem (scalar product of the matrices) is a more convincing explanation. Indeed, first, we have many solutions whose value is close to the optimum. So, even when the best solution is obtained, it is very hard to prove its optimality.

3 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) algorithm is inspired by social behavior patterns of organisms that live and interact within large groups. In particular, PSO incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior. This method was proposed by Kennedy and Eberhart, 1995[6].

The classical PSO model consists of a swarm of (solution) particles, which are initialized with a population of random candidate solutions. The particle swarm moves iteratively through the *d*-dimension problem space to search for new solutions, where the fitness ***f***, can be calculated as the certain qualities measure. Each particle has a position represented by a position-vector x_i (***i*** is the index of the particle), and a velocity represented by a velocity-vector v_i. Each particle remembers its own best position so far in a vector p_i. The best position-vector among the swarm so far is then stored in a vector p_g. During the iteration time ***t***, the update of the velocity from the previous velocity to the new velocity is determined by Eq.(8). The new position is then determined by the sum of the previous position and the new velocity by Eq.(9).

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{gj}(t) - x_{ij}(t)) \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

where ***r*1** and ***r*2** are the random numbers in the interval [0,1], ***c*1** is a positive constant, called as coefficient of the self-recognition component, ***c*2** is a positive constant, called as coefficient of the social component. The variable ***w*** is called as the inertia factor, whose value is typically setup to vary linearly from 1 to near-0 during the iterative processing. From Eq.8, a particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of its most successful solution particle in the swarm, it is also a parameter to control the impact of the previous velocities on the current velocity [7].

4 PSO for the Q3AP

In order to guide the particles effectively in the search space, the maximum moving distance during one iteration is clamped in between the maximally different velocities $[-v_{max}, v_{max}]$.

Evaluation of each particle in the swarm requires the determination of the permutation of numbers ***1...n*** since he value of function z in Q3AP problem is a result of the sequence.

In this paper, we use a heuristic rule called Smallest Position Value (SPV) [8] to enable the continuous PSO algorithm to be applied to all class of sequencing problems, which are NP-hard in the literature. By using the SPV rule, the permutation can be determined through the position values of the particle so that the fitness value of the particle can then be computed with that permutation.

The pseudo-code for particle swarm optimization algorithm is illustrated in Algorithm 1

Algorithm 1: Particle Swarm Optimization (PSO)
Initialize parameters Initialize population Find sequence Evaluate Do { Find the personal best Find the global best Update velocity Update position Find sequence Evaluate } While (Termination)

A population of particles is constructed randomly for the PSO algorithm of the QAP problem. The continuous values of positions are established randomly. The following formula is used to construct the initial continuous position values of the particle**:** $x_{ij}=x_{min}(x_{max} - x_{min}) \times U(0,1)$**,** where x_{min} = 0.0 , x_{max} = 4.0 and $U(0,1)$ a uniform random number between 0 and 1. similar formula as follows $v_{ij}=v_{min}(v_{max} -v_{min})\times U(0,1)$**,** where v_{min} = -4.0 , v_{max} = 4.0

Step 1: Initialization

• Set ***k***=0, ***m***=size of swarm.

• Generate *m* particles randomly as explained.

• Generate initial velocities of particles randomly.

• Apply the SPV rule to find the sequence with the best fitness value.

• Evaluate each particle ***i*** in the swarm using the objective function Z_i for ***i=1,…, m***.

• For each particle i in the swarm, set best personal position along with its best fitness value.

• Find the best fitness value $Z_l = min\{Z_i\}$ for ***i=1,…, m*** with its corresponding position.

• Set global best with its fitness value.

Step 2: Update iteration counter.

• ***k = k +1.***

Step3: Update inertia weight.

• $w=w^* \alpha$, where α is the damping ratio.

Step 4: Update velocity (Eq 8).

Step 5: Update position (Eq 9).

Step 6: Find Sequence.

•Apply the SPV rule to find the sequence with the best fitness value.

Step 7: Update personal best.

• Each particle is evaluated by using its sequence to see if personal best will improve. That is, if $f(x) < f(PB)$ for ***i=1,...,m*** , then personal best is updated.

Step 8: Update global best.

• Find the minimum value of personal best.

• If $f(x) < f(GB)$, then the global best is updated.

Step 9: Stopping criterion.

• If the number of iteration exceeds the maximum number of iterations, then stop, otherwise go to step 2.

5 Experimental results

In this section the results of the implementation of the algorithm is described. A PSO algorithm presented for the Q3AP problem was coded in the C++ programming language.

We used the following parameters for the PSO. Social and cognitive parameters, and uniform random numbers are taken as $c_1 = c_2 = 2$ and $r1, r2$ respectively uniform numbers are taken from $(0,1)$. Initial inertia weight is set to $w_0 = 1.0$. Finally, the decrement factor α is taken as 0.99.

It is clear that the performance of the approximated algorithms is affected by parameter tuning. So, at first we do tuning process, to obtain good values for the key parameters. We use nug-8 instance of QAPLIB to test the effect of PSO parameters.

We increase the number of iterations of the algorithm ranging to 1000. The figure below shows the results. The first point is that as the number of iterations increases, the quality of solutions will improve. There is another point noticeable in which the cost decreased; it is the choice of the lower and upper bound of variables as well as the choice of the learning coefficients (c_1, c_2).

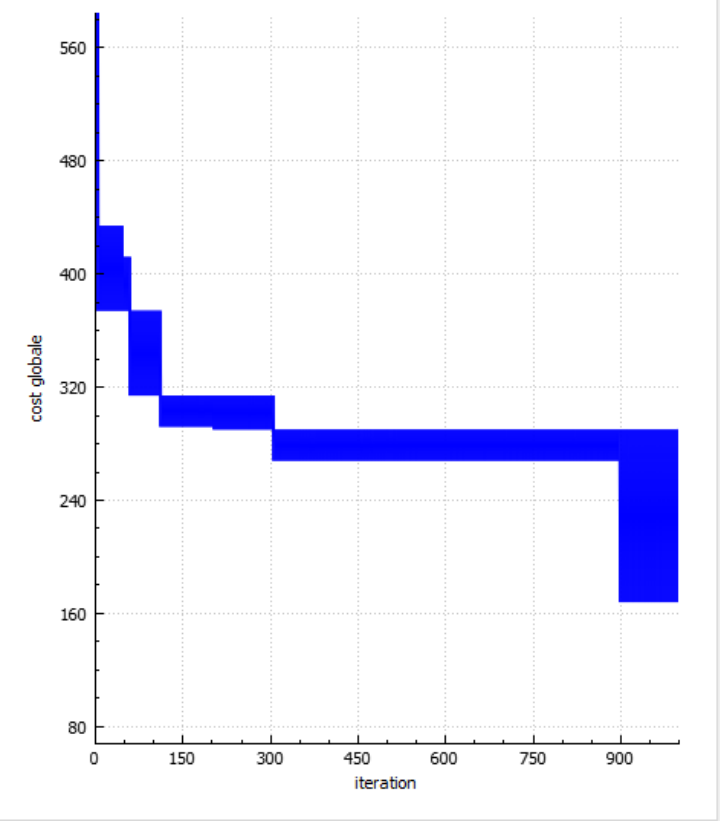


Figure1: the impact of changing PSO parameters.

We choose the size of swarm from the set {80}, the lower one is the best results we found 168 using the values parameters given before, and the upper one is one of simulations when we changed one or more than one of the parameters.

6 Conclusion

The quadratic 3-dimensional assignment problem (Q3AP) is a very difficult combinatorial optimization problem and has reviewed nowadays existing methods for solving such problem included.

In this paper we defined the approach based on a particle swarm optimization for the quadratic three dimensional problem. This work is in-progress, we evaluated the performance of our approach and compare it with what existed to tell that the result that the proposed approach returned was a close value from the optimal which is 134 for nug-8 for the quadratic three dimensional assignment problem but it still a poor result, and we have plans to improve that result with additional experiments.

References

[1] P.M. Hahn, Y.R. Zhu, M. Guignard and J. M-G. Smith Exact Solution of Emerging Quadratic Assignment Problems.2013

[2] W.P. Pierskalla, The multi-dimensional assignment problem. Technical Memorandum No. 93, Operations Research Department, CASE Institute of Technology, 1967.

[3] P. M. Hahn et al. The Quadratic Three-dimensional Assignment Problem : Exact and Approximate Solution Methods. European Journal of Operational Research, 2008.

[4] Butong Zhang, Xin Zhang, Jiancun Hu, and Dacheng Yang. A hybrid ARQ scheme with constellation rearrangement and power adjustment. In Proceedings of the International Conference of Wireless Communications, Networking and Mobile Computing, pages 464–468, 2005. ISBN 0-7803-9335-X.

[5] Samra, H., Ding, Z., Hahn, P.M., 2005. Symbol mapping diversity design for multiple packet transmissions. IEEE Transactions on Communications 53 (5), 810–817.

[6] J. Kennedy et R. Eberhart. Particle swarm optimization, proceedings of IEEE international conference on neural networks. doi :10.1109/ICNN.1995.488968 :1942–1948, 1995

[7] Y. Shi et R. C. Eberhart. A modified particle swarm optimizer. In Proceedings of the IEEE international conference on evolutionary computation, pages 69–73, 1998

[8] A.S.Mamaghani et M.R.Meybodi. Solving the Quadratic Assignment Problem with the Modified Hybrid PSO Algorithm, Application of Information and Communication Technologies (AICT), 2012 6th International Conference on. IEEE, pp. 1-6

[9] Liu H., Abraham A., Zhang J.: “A Particle Swarm Approach to Quadratic Assignment Problems”; Proceedings of The Eleventh Online World Conference on Soft Computing in Industrial Applications (WSC11), Springer (in press).

[10] Szwed, P., Chmiel, W.: Multi-swarm PSO algorithm for the quadratic assignment problem: a massive parallel implementation on the OpenCL platform. Comput. Res. Repos. 1504.05158 (2015)

Genetic Algorithm Guided By Pretreatment Information For The 0-1 Multidimensional Knapsack Problem

Abdellah Rezoug · Mohamed Bader-El-Den · Dalila Boughaci

Abstract This paper presents an improved version of the Genetic Algorithm (GA) for solving the well-known 0-1 Multidimensional Knapsack Problem (MKP01). The Genetic Algorithm Guided by Pretreatment information (GAGP) is composed of two steps, in the first, a greedy-based simple pretreatment extracts the subset of items that likely to be contained in the best solutions. In the second, the population initialisation and the fitness function of a standard GA are based on the pretreatment information, in addition to an efficiency update operator. The pretreatment information has been investigated using the CPLEX deterministic optimiser. In addition, GAGP has been examined on the most used MKP01 data-sets, and compared to several other approaches. The obtained results showed that the pretreatment succeeded to extract the most part of the important information. It has been shown, that GAGP is a simple but very competitive solution.

Keywords Greedy algorithm · genetic algorithm · 0-1 multidimensional knapsack problem · core concept · guided genetic algorithm

A. Rezoug
Department of Computer Science
University Mhamed Bougara, Boumerdes, Algeria
E-mail: abdellah.rezoug@gmail.com

M. Bader-El-Den
School of Computing
University of Portsmouth, Portsmouth, United Kingdom
E-mail: mohamed.bader@port.ac.uk

D. Boughaci
Department of Computer Science
University of Science and Technology Houari Boumediene, Algiers, Algeria
E-mail: dalila_info@yahoo.fr

1 Introduction

The MKP01 is composed of N items and a knapsack with m different capacities c_i where $i \in \{1, \ldots, m\}$. Each item j where $j \in \{1, \ldots, n\}$ has a profit p_j and can take w_{ij} of the capacity i of the knapsack. The goal is to pack the items in the knapsack so as to maximize the profits of items without exceeding the capacities of the knapsack. The MKP01 can be represented as the following integer program:

$$\textbf{Maximize}: \quad \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\textbf{Subject to}: \sum_{j=1}^{n} w_{ij} x_j \leq c_i \qquad i \in \{1 \ldots m\} \tag{2}$$

$$x_j \in \{0, 1\} \qquad j \in \{1 \ldots n\} \tag{3}$$

During the past few decades several variants of GA have been proposed, all of them aim to increase the performance of GA and boost its convergence. Most of these ideas are either based on changing the GA operators such as: crossover and mutation (e.g. one-point, two-point, cut and splice, three parents, uniform, flip bit, Boundary, non-uniform, uniform, etc.), or based on modifying the GA's evolutionary behaviour, such as: Hybrid GA (Rezoug et al. 2015; Valls et al. 2008), Distributed GA (Adeli and Kumar 1995), Adaptive GA (Alavidoost et al. 2015), Parallel GA (Sudholt 2015), Genetic Programming (Poli and Koza 2014), etc. An extended overview of the GA variations is available in (Castro et al. 2013). The focus here is only on the methods that are related to the guided GA concept.

The concept of *proximate optimality* suggests that, in most cases, the best solutions have a similar structure. Senju and Toyoda (1968) presented a primal greedy gradient algorithm for the MKP01 that establishes a decreasing sort of the items such as the most priority is given to those most likely to form the best solution. The sort is calculated according to an *efficiency measurement* that try to find the compromise between the profit and the weight. Latter Puchinger et al. (2006) applied the principal of *efficiency measurement* in addition to the *core concept* for reducing the size of the problem data to only the most relevant items. On the other hand, the process of GA is stochastic, this leads to an important useless work. Our aim in this this paper is to reinforce the GA process using the useful information about the items. To this purpose, the Genetic Algorithm Guided by Pretreatment information (GAGP) is proposed. Firstly, GAGP applies the primal greedy with the core concept decomposition to extract a useful information about the subset of important items. Secondly, specific population initialisation, fitness function and update efficiency measurement operators augment a standard GA by exploiting the pretreatment information. In GAGP, an important rang of solutions are avoided and the process does not consider the non relevant solutions.

The paper is structured as follows: Section 2 gives an overview of the literature review related to the GA with guidance. The proposed algorithm GAGP is introduced in Section 3. Section 4 presents the conducted experiments and the obtained results. The conclusions and final remarks are drawn in Section 5.

2 Related Works

There are several methods related to the guided GA concept in the literature, that has been applied to a wide range of applications. For solving the Course Timetabling Problem, the approaches by (Jat and Yang 2009; Yang and Jat 2011) use a memory denoted MEM to record useful information to guide the GA process and improve its performance. MEM is a list of limited size, in which a list of room and time slot pairs is recorded. This information is integrated into the crossover operator of the proposed guided GA. Other researchers used an external structure to guide GA such as (Acan and Tekol 2003; Louis and Li 1997). Another approach for guiding the GA is through the use of approximate probabilistic models.

In (Chen et al. 2012; Zhang 2009) The GA is augmented with an approximate probabilistic model to guide the crossover and mutation operators. The probabilistic model is used to estimate the quality of candidate solutions generated by the traditional crossover and mutation operators. It also evaluates the quality of candidate solutions. This estimation enables the crossover and mutation operators to generate more promising solutions.

A subset of the genetic operators is guided. The proximate optimality principle assumes that good solutions have a similar structure. Based on this principle, the guided mutation proposed by (Zhang et al. 2005) uses a probability model inspired by estimation of distribution algorithms EDA mutation operator. The generated offspring by this operator is constructed based on the best parent so far and a dynamic probability model and a probability β. This allows conducting the searching process in promising areas. A guided crossover operator has been proposed by (Rasheed 1999). The crossover operator works by using guidance from all members of the GA population to select a direction for exploration. The first parent is selected by the selection operator. To select the second parent, a metric named $Mutual\_fitness$ is calculated for all the other chromosomes. The chromosome which has the maximum value is selected. One offspring is generated by crossing the parents in a point chosen randomly such that the offspring resulting is the best.

3 Genetic Algorithm Guided by pretreatment information for The MKP01

The algorithm in this paper is motivated by the observation that in may optimisation real-world problem, we may have some prior information about the components/patterns that are likely to appear in the good solutions. For example, in MKP01, it is possible using linear relaxation or the "optimal fractional

solution" (Dantzig 1957; Shih 1979) to predict some of the items that are likely or unlikely to appear in the good solutions. This study proposes a method for using such prior information as an additional guide for the GA evolutionary process for the MKP01 problem. By guide, we mean any structure external to GA, which maintains its original composition and is used to drive its search process. This can be through a subset of operators, in order to accelerate the search process and improve the speed of convergence. This section aims to describe the GAGP components.

3.1 Pretreatment

The guiding information is based on the work by (Puchinger et al. (2006)). The items are sorted in decreasing order according to a statistical efficiency e_j based on the profit and the cost. In simple words, the items are sorted based on how likely each item to appear in high performing individuals, the item at the top of this list are the items that are likely to be selected while the items at the bottom of the list are the items that are unlikely appear in good solutions. However, it is important to note here that this list is just an estimate and not a predefined part of the solution. It should be noted also that the Greedy heuristic as by Senju and Toyoda (1968) only based on the efficiency sorting is not an effective solution for the strongly correlated problem instances of the MKP01 (Huston et al. 2008).

$$e_j^{st} = \frac{p_j}{\sum_{j=1}^{m} w_{ij}(\sum_{l=1}^{n} w_{il} - c_i)} \tag{4}$$

The sorting operation allows favouring items that have a good compromise (i.e. efficiency) between the average profit and overall capacity. The efficiency of an item is high if its profit is high while its required global capacity is low. The sorted items are split into three sets where the value of each variable is assigned as follows:

- $X_1 : x_j = 1$ The variables have the best efficiency e_j. These variables are most likely to build the best solutions even the optimal solution.
- $Core : x_j =?$ The variables have the values of the efficiency e_j very close. In this group, it is difficult to determine the best.
- $X_0 : x_j = 0$ The variables have a very low efficiency e_j, in other words, the profit is low or the capacity is large or both.

The guide is represented by the items of $X_1 \cup Core \cup X_0$. The sizes of X_1, $Core$ and X_0 are determined as follows: Construct a feasible solution by adding the items in the order. The item that makes the solution infeasible represents the center of $Core$. The size of each part of the guide depends on the size of $Core$. Set the size of $Core$ defines the size of the other parts.

3.2 Guided Genetic Algorithm Optimisation

The GAGP chromosome consists of the set of the items to be added to the knapsack. GAGP uses the integer representation, where each gene presents an item ID. The items are coded as integer numbers. A chromosome is formed only by the number of items that it contains. This representation allows reducing the size of the processed data.

1. *Initial population.* GAGP algorithm uses a special initialization process which allows the GA to make use of the prior information available about the items, and in the same time generates a diverse initial population to ensure exploration of the search space. A chromosome is generated from the items of X_1 completed by items generated randomly. In each chromosome, X_1 is integrated with a probability α. If α is set to zero this means that all the items in each individual are selected randomly, while $\alpha = 1$ means that each individual in the initial population contains all the items in X_1. This method allows having an initial population of good quality by integrating X_1 and ensures the diversification by adding the rest randomly.
2. *Fitness evaluation.* Besides the population initialisation, the guidance by the pretreatment information is integrated in the GA by this operator. The fitness function $f(j)$ is evaluated according to Eq. 1. The efficiency e_j is introduced in its evaluation according to Eq. 5. Each generation, the fitness value of each chromosome is calculated. The fitness formula allows giving more chance to the chromosome that has a high efficiency to be selected more than the others.
$$f(j) = \sum_{j=0}^{n} e_j p_j x_j \tag{5}$$
3. *Genetic operators.* GAGP uses standard genetic crossover and mutation operators. A tournament selection of size 5 is used as the selection method, and the random single point method is applied with a probability p_c as a crossover method. For the mutation operator, the random multiple point bit flip with the probability p_m is adopted. And finally, a reproduction operator copies a subset of individuals with the probability p_r such as $p_c + p_m + p_r = 1$.
4. *Update efficiency.* The Sorting efficiency is not always efficient especially for the problems with strong correlation. A step of efficiency update is proposed that aims to make a perturbation in the items efficiencies. Two items $j, j', j \in X_1$ and are selected and their efficiency is permuted. Rather than maintaining the same guidance, the search process diversify the guide with the items in X_1 and $Core$. This modification has an impact on the fitness evaluation and so on the whole process of GA.
5. *Stopping condition* The process of optimisation is repeated until a specific number of iteration is reached.

The algorithm could be described by the following pseudo-code (Algorithm 1).

Algorithm 1 The GAGP pseudo-code.

Require: MKP01 instance
Ensure: a feasible solution S
calculate the efficiency e_j for each variable
sort the items according to the efficiency measurement
calculate X_1, $Core$ and X_0 of the guide
initialize the population pop with X_1 and α
for $ctr = 1$ to ng **do**
    evaluate the fitness for each chromosome in pop according to the fitness equation
    crossover with (pc)
    mutation with (pm)
    reproduction with (pr)
    select randomly items j, j' such as $j \in X_1$, $j' \in Core$ and permute their efficiencies
end for
return the best solution S^*.

4 Experimental Results

The experiments aims to compare the proposed GAGP with the state-of-art results reported in the literature (Section 4.2). For an experimental purpose, and because the chosen sorting method concerns MKP01, it is natural to use data from this problem. The test platform is a Toshiba laptop with 4GB RAM capacity and an Intel Core (TM) i5-4200 M 2.5 Ghz CPU. The Java language is used to implement the approach. As for the test data, two well known benchmarks from the OR-Library[1] are used.

4.1 Analytical study of the guidance

This analytical study compares firstly, between the items of the optimal solution and the two main parts of the guide (i.e. X_1 and $Core$) and secondly, between the optimal solution and the solution obtained by GAGP. The aim is on the one hand, for understanding how significant is the sort and measure its effective impact on the GA; On the other hand, to find if the GAGP does effectively follow the guide and what happens to drop the GA in the wrong solution whether because of the guide or because of the optimisation process itself.

The composition of the optimal solution S^*, calculated using the deterministic CPLEX optimizer 12.5, is compared to a feasible solution S obtained by GAGP. Also, the items of the X_1 and $Core$ and the placement of the items of the S^* in the three parts of the guide are given (where +, * and - corresponds to item in X_1, in $Core$ and in X_0 respectively). The first four instances (OR5x100.0.25 1-4) are used to conduct this analysis. Finally, the Distance From the Optimum D.F.O of the solution calculated by GAGP is given. The obtained results of the comparison are reported in Table 1.

[1] http://people.brunel.ac.uk/ mastjjb/jeb/orlib/files/

Table 1 Analytical comparison of GAGP feasible solution composition to the composition of the optimal solution obtained by CPLEX and the composition of the parts of the guidance information using the OR5x100.0.25 1-4 instances.

S[a]	S*[b]	G[c]	X_1	S	S*	G	X_1	S	S*	G	X_1	S	S*	G	X_1
1	1	+	1	1	3	+	3	7	4	+	4	0	0	*	3
3	3	*	4	3	10	+	10	10	11	+	11	1	1	*	5
6	6	+	6	10	18	*	27	11	13	+	13	3	3	+	11
8	8	+	8	20	20	*	28	13	18	+	18	5	5	+	12
10	10	-	23	27	27	+	34	18	19	-	21	6	8	-	22
12	18	*	26	34	28	+	45	21	21	+	28	8	11	+	27
17	23	+	31	36	34	+	49	28	26	-	37	11	13	-	28
23	25	-	43	39	36	-	56	32	28	+	44	12	22	+	30
26	26	+	49	41	41	*	57	34	32	*	48	22	24	-	34
28	28	*	56	42	42	*	61	36	34	-	55	24	27	+	35
29	29	*	62	45	45	+	62	37	37	+	69	26	30	+	42
31	31	+	65	48	48	*	73	42	42	*	72	27	34	+	53
41	43	+	68	49	49	+	90	44	44	+	74	30	35	+	63
43	49	+	76	53	53	-	91	48	48	+	84	34	42	+	69
49	56	+	78	56	56	+	93	51	51	-	87	35	49	*	70
56	61	-	85	57	57	+	95	53	55	+	92	42	53	+	78
62	62	+	92	58	58	-	99	55	59	*		49	54	-	86
65	65	+		61	61	+		59	60	-		53	55	-	94
68	68	+	Core	62	62	+	Core	60	64	*	Core	54	56	*	
76	70	-	3	70	64	*	18	64	72	+	10	55	58	*	Core
78	73	*	15	73	73	+	20	72	74	+	32	56	61	*	0
84	76	+	18	74	74	*	41	74	78	*	36	61	63	+	1
85	78	+	28	81	81	*	42	78	79	-	42	63	74	-	15
91	84	-	29	88	88	*	48	79	84	+	59	68	78	+	49
92	85	+	34	90	90	+	64	84	87	+	64	70	79	-	56
94	91	-	66	91	91	+	74	87	92	+	78	76	86	+	58
95	92	+	73	92	93	+	81	92	93	*	93	79	94	+	61
98	95	-	81	93	95	+	88	93	96	*	96	86	95	+	65
	98	+	98	95	99	+	92	96	99	*	99	95			68
				99											95
D.F.O[d] = 0.82				D.F.O[d] = 0.51				D.F.O[d] = 0.23				D.F.O[d] = 0.50			

[a] S : Items of the solution obtained by GAGP
[b] S* : Items of the optimal solution obtained by CPLEX
[c] G : Group of S* item in the guide (item is : + $\in X_1$, - $\in X_0$ or * $\in Core$)
[d] D.F.O : Distance Form the Optimum in % of the GAGP solution

A percentage of $75-90\%$ of the items in S^* are included in X_1 or $Core$. Similarly, S contained $75-90\%$ of the items of S^*. Almost the same items initially contained in X_1 and $Core$ are maintained in S. Some (3 to 7) items of the excluded part X_0 appear in S^* at the same time some were introduced in S by the mutation operator. In the first three instances, at most one item form X_1 has not been contained in S^*. GAGP could be more effective by introducing a better mutation operator. The efficiency measurement function would be more effective if $Core$ contained a slightly more items of X_0. Most items of $Core$ were components of S^*, that supports the efficiency update operator proposed in GAGP.

4.2 Comparison with the literature

As with most optimisation problems, MKP01 heuristics could be classified in two groups: the first is*constructive* heuristics, that aim to construct a solution. The second is *improvement* heuristics which aim to improve a given initial solution normally generated first by a *constructive* heuristic. The proposed method is considered as a *constructive* heuristic. However, in order to demonstrate the performance of the proposed method, the performance of the GAGP is compared with both *constrictive* and *improvement* approaches. The following is short description of the methods (*constructive* and *improvement*) used in the comparison presented in this section. GAGP is compared with the standard

GA algorithm and other state-of-the-art optimisation methods reported in the literature. GAGP is compared to the following constructive approaches : PECH (Primal Effective Capacity Heuristic) (Akcay et al. 2007); MAG (Magazine and Oguz 1984); VZ (Volgenant and Zoon 1990); PIR (Pirkul 1987) and SCE (Shuffled Complex Evolution) (Baroni and Varejão 2015). GAGP is also compared to the following improvement approaches : CB (Chu and Beasley 1998); NR (P) (New Reduction (Pirkul)) (Hill et al. 2012) and MCF (Modified Choice Function - Late Acceptance Strategy) (Drake et al. 2015). The comparison is shown in Table 2. As shown in table 2, GAGP is competitive with both construction and improvement methods and has managed to outperform both group of methods on a few instances.

Table 2 Comparison of results obtained by GAGP with GA, constructive and improvement heuristics

				Constructive						Improvement		
n	m	α	GAGP	GA	PECH	MAG	VZ	PIR	SCE	CB	NR(P)	MCF
5	100	0.25	**0.35***	2.17	7.3	13.6	10.3	1.6	3.5	0.99	0.94	1.09
		0.50	**0.48**	0.86	3.4	6.7	6.9	0.77	2.6	0.45	**0.44***	0.57
		0.75	**0.21***	0.42	2.02	5.1	5.6	0.48	1.1	0.32	**0.22**	0.38
	250	0.25	0.58	4.03	7.1	6.6	5.8	**0.53**	4.3	**0.23***	0.46	0.41
		0.50	0.36	1.15	3.2	5.2	4.4	**0.24**	3.3	**0.12***	0.17	0.22
		0.75	0.23	0.58	1.8	3.5	3.5	**0.16**	1.5	**0.08***	0.1	0.14
	500	0.25	0.51	4.27	6.4	4.9	4.1	**0.22**	4.6	1.56	**0.15***	0.21
		0.50	0.36	1.45	3.4	2.9	2.5	**0.08**	3.6	0.79	**0.06***	0.1
		0.75	0.22	0.65	1.7	2.3	2.41	**0.06**	1.8	0.48	**0.03***	0.06
10	100	0.25	**1.0**	2.40	8.2	15.8	15.5	3.4	6.8	**0.09***	2.05	1.87
		0.50	**0.53**	1.53	3.7	10.4	10.7	1.8	5.1	**0.04***	0.81	0.95
		0.75	**0.27**	0.53	1.8	6.1	5.67	1.1	2.4	**0.03***	0.44	0.53
	250	0.25	**0.75**	3.56	5.8	11.7	10.5	1.1	6.9	**0.51***	0.88	0.79
		0.50	0.48	1.35	2.5	6.8	5.9	**0.57**	5.4	**0.25***	0.39	0.41
		0.75	**0.27**	0.66	1.5	4.4	3.7	0.33	2.8	**0.15***	0.19	0.24
	500	0.25	0.71	3.61	5.1	8.8	7.9	**0.52**	6.8	**0.24***	0.34	0.44
		0.50	0.4	1.44	2.4	5.7	4.1	**0.22**	5.8	**0.11***	0.14	0.2
		0.75	0.29	0.71	1.2	3.6	2.9	**0.14**	3.4	**0.07***	0.1	0.13
30	100	0.25	**1.56***	2.27	6.8	17.3	17.2	9.1	8.6	2.91	2.24	3.61
		0.50	**1.07***	1.72	3.2	11.8	10.1	3.51	6.6	1.34	1.32	1.6
		0.75	**0.36***	0.78	1.9	6.58	5.9	2.03	3.6	0.83	0.8	0.97
	250	0.25	**1.66**	3.20	4.8	13.5	12.4	3.7	8.3	**1.19***	1.27	1.75
		0.50	**1.0**	1.46	2.1	8.6	7.1	1.5	6.9	**0.53***	0.75	0.79
		0.75	**0.5**	0.73	1.2	4.4	3.9	0.84	3.8	**0.31***	0.38	0.43
	500	0.25	4.07	3.50	3.7	9.8	9.6	**1.89**	8.6	**0.61***	0.89	1.05
		0.50	2.14	1.45	1.7	7.1	5.7	**0.73**	7.4	**0.26***	0.36	0.44
		0.75	0.51	0.69	0.9	3.7	3.5	**0.48**	4	**0.17***	0.23	0.27

5 Conclusion

This paper aims to present a modified version of GA. Extracted information about the variables likely to appear in the best solutions are used to guide the search process of GA. The approach called Genetic Algorithm Guided by Pretreatment information (GAGP) begins by analysing the problem data

using a gradient greedy sorting method which sorts the variables according to an efficiency value expressed by profit and cost. These information are used to drive the GA search process by its integration in the generation of the initial population and for measuring the fitness function. Some experiments were conducted using a set of well-known MKP01 data. It has been shown that the information improves the performance of GA. The pretreatment allows to reduce the size of the problem to only the most relevant space of solutions, this allows the search process to avoid the areas of worst solutions. In addition, the results obtained in the resolution of MKP01 are competitive. As prospects for the next step, we expect to apply the method to other optimisation problems.

References

Acan A, Tekol Y (2003, January) Chromosome reuse in genetic algorithms. In *Genetic and Evolutionary Computation-GECCO 2003* (pp. 695-705).

Adeli H, Kumar S (1995) Distributed genetic algorithm for structural optimization. *Journal of Aerospace Engineering, 8* (3), (pp. 156-163).

Akcay Y, Li H, Xu SH (2007) Greedy algorithm for the general multidimensional knapsack problem. *Annals of Operations Research, 150*(1), 17-29.

Alavidoost MH, Tarimoradi M, Zarandi MF (2015) Fuzzy adaptive genetic algorithm for multi-objective assembly line balancing problems. *Applied Soft Computing, 34*, (pp. 655-677).

Baroni MDV, Varejão FM (2015) A Shuffled Complex Evolution Algorithm For the Multidimensional Knapsack Problem. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (pp. 768-775). Springer International Publishing.

Castro F, Gelbukh A, & González M (Eds.) (2013) *Advances in soft computing and its applications: 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, Mexico City, Mexico, November 24-30, 2013 : Proceedings, part II* (Vol. 8266). Mexico City:

Chen SH, Chang PC, Cheng TCE, Zhang Q (2012) A self-guided genetic algorithm for permutation flowshop scheduling problems. *Computers & Operations Research, 39* (7), (pp. 1450-1457).

Chu PC, Beasley JE (1998) A genetic algorithm for the multidimensional knapsack problem. *Journal of heuristics, 4*(1), 63-86.

Dantzig GB (1957) Discrete-Variable Extremum Problems. *Operations Research, 5* (2), 266-277. Retrieved from http://www.jstor.org/stable/167356

Drake JH, Ozcan E, Burke EK (2015) Modified choice function heuristic selection for the multidimensional knapsack problem. In *Genetic and Evolutionary Computing* (pp. 225-234). Springer International Publishing

Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Simulation, 76* (2), (pp. 60-68).

Hill RR, Cho YK, Moore JT (2012) Problem reduction heuristic for the 0-1 multidimensional knapsack problem. *Computers & Operations Research, 39*(1), 19-26.

Huston S, Puchinger J, Stuckey P (2008, January) The core concept for 0/1 integer programming. In *Proceedings of the fourteenth symposium on Computing: the Australasian theory-Volume 77* (pp. 39-47). Australian Computer Society.

Jat SN, Yang S (2009) A guided search genetic algorithm for the university course timetabling problem. In *The 4th Multidisciplinary International Scheduling Conference: Theory and Applications*(MISTA 2009), Dublin, Ireland: 180-191, 10-12 Aug 2009.

Louis S, Li G (1997) Augmenting genetic algorithms with memory to solve traveling salesman problems. In *Proceedings of the Joint Conference on Information Sciences* (pp. 108-111).

Magazine MJ, Oguz O (1984) A Heuristic Algorithm for the Multidimensional Zero-One Knapsack Problem. *European Journal of Operational Research, 16*, 319-326.

Poli R, Koza J (2014) *Genetic programming* (pp. 143-185). Springer US.

Puchinger J, Raidl GR, Pferschy U (2006) The core concept for the multidimensional knapsack problem. In *Evolutionary Computation in Combinatorial Optimization* (pp. 195-208).

Rasheed K (1999) Guided crossover: A new operator for genetic algorithm based optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 2). IEEE.

Rezoug A, Boughaci D, Badr-El-Den M (2015) Memetic Algorithm for Solving the 0-1 Multidimensional Knapsack Problem. In *Progress in Artificial Intelligence* (pp. 298-304).

Senju S, Toyoda Y (1968) An approach to linear programming with 0-1 variables. *Management Science*, B196-B207.

Shih W (1979) A Branch and Bound Method for the Multiconstraint Zero-One Knapsack Problem. *Journal of the Operational Research Society, 30*, 369-378. doi:10.1057/jors.1979.78

Sudholt D (2015) Parallel evolutionary algorithms. In *Springer Handbook of Computational Intelligence* (pp. 929-959).

Valls V, Ballestin F, Quintanilla S (2008) A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research, 185* (2), (pp. 495-508).

Volgenant A, Zoon JA (1990) An Improved Heuristic for Multidimensional 0-1 Knapsack Problems. *Journal of Operational Research Society, 41*, 963-970.

Yang S, Jat SN (2011) Genetic algorithms with guided and local search strategies for university course timetabling. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 41* (1), (pp. 93-106).

Zhang Q, Sun J, Tsang E (2005) An evolutionary algorithm with guided mutation for the maximum clique problem. *Evolutionary Computation, IEEE Transactions on, 9* (2), (pp. 192-200).

Zhang Q (2009) A guided memetic algorithm with probabilistic models. *International Journal of Innovative Computing, Information and Control, 5* (12 (B)), (pp. 4753-4764).

SimGA: a simulating approach to fuzzy job shop problems

J. Puente[1], C. R. Vela[1], I. Gonzalez-Rodriguez[2], and E-G. Talbi[3]

[1] Dep. de Informática, University of Oviedo (Spain)
puente@uniovi.es, crvela@uniovi.es
[2] Dep. de Matemáticas, Estadística y Computación, University of Cantabria (Spain)
gonzalezri@unican.es
[3] Univ. Lille, CNRS, INRIA (France)
el-ghazali.talbi@lifl.fr

Abstract. We consider the fuzzy job shop, a job shop scheduling problem with uncertain processing times modelled as triangular fuzzy numbers. While the usual approaches to solving this problem involve adapting existing metaheuristics to the fuzzy setting, we propose instead to follow the framework of simheuristics from stochastic optimisation. More precisely, we integrate the simulation of possible realisations of the fuzzy problem with a genetic algorithm that solves the deterministic job shop. We test the resulting method, simGA, on a testbed of 23 benchmark instances and obtain results that suggest that this is a promising approach to solving problems with uncertainty by means of metaheuristics.

1 Introduction

Scheduling is with no doubt a research field of great importance, involving complex combinatorial constraint-satisfaction and optimisation problems and with relevant applications in industry, finance, welfare, education, etc [1, 2]. In particular, the job shop problem in its numerous variants is a model for many real problems which has posed and still poses a challenge to the research community due to its complexity. It is this complexity that has led many researchers to resort to metaheuristic techniques in order to find approximate good solutions for the problem, as is the case with many hard optimisation problems [3, 4].

Traditionally, it has been assumed that problems are static and certain: all the variables in the problem are precisely known in advance and do not change as the solution is being executed. However, for many real-world scheduling problems design variables are subject to perturbations or changes, causing optimal solutions to the original deterministic problem to be of little or no use in practice. This has motivated an increasing effort to take incorporate uncertainty and variability in the scheduling model, in order to find useful solutions and bridge the gap between theory and practice.

A source of changes in scheduling problems is the uncertainty in activity durations. There exists great diversity of approaches to dealing with this kind of uncertainty [5]. Perhaps the best-known is stochastic scheduling [1], although fuzzy sets and possibility theory provide an interesting alternative, with a tradeoff between the expressive power of probability and their associated computational complexity and knowledge demands [6, 7]. In the particular case of the job shop, there are numerous research papers where uncertain durations are modelled as fuzzy numbers, mostly triangular fuzzy numbers. These range from the seminal papers [8–10] to the most recent [11, 12] to mention but a few. An extensive review of the different proposals can be found in [13], which highlights the relevance of this topic.

So far, most methods used for solving the fuzzy job shop consist in adapting existing metaheuristics to the fuzzy setting. This is not without considerable effort, given the added complexity that results from handling and propagating uncertain information. It is also necessary in some cases to approximate certain operations for the sake of computational tractability, causing different models and approximation criteria to co-exist in the literature. An alternative would be to use an approach based on simulations, following the methodology for dealing with stochastic combinatorial optimisation problems known as "Simheuristics".

Simheuristics constitute a methodology that allows for extending metaheuristics through simulation to solve stochastic combinatorial optimisation problems [14]. The underlying idea is to handle real-life uncertainty by integrating simulation (in any of its variants) into a metaheuristic-driven

framework. This approach takes the stance that there exist efficient metaheuristic for deterministic versions of a combinatorial optimisation problem and, in scenarios with moderate uncertainty, high-quality solutions for the deterministic version of the problem might as well be of high quality in the corresponding uncertain setting.

All the above motivates this work, where we propose to adopt the simheuristics framework to solve the fuzzy job shop problem. The remaining of this paper is organised as follows: after describing in Section 2 the fuzzy job shop problem, Section 3 presents our proposal to solve the problem using a genetic algorithm (GA) to solve a deterministic counterpart of the problem combined with simulations. Finally, Section 4 presents some experimental results to assess the proposal and Section 5 presents some conclusions and ideas for future work.

2 Fuzzy Job Shop Problem

The *job shop scheduling problem*, also denoted *JSP*, consists in scheduling a set of jobs $\{J_1, \ldots, J_n\}$ on a set of physical resources or machines $\{M_1, \ldots, M_m\}$, subject to a set of constraints. There are *precedence constraints*, so each job J_i, $i = 1, \ldots, n$, consists of m tasks $\{\theta_{i1}, \ldots, \theta_{im}\}$ to be sequentially scheduled. Also, there are *capacity constraints*, whereby each task θ_{ij} requires the uninterrupted and exclusive use of one of the machines for its whole processing time. A solution to this problem is a schedule (an allocation of starting times for all tasks) which, besides being *feasible*, in the sense that precedence and capacity constraints hold, is optimal according to some criteria, for instance, that the makespan is minimal or its robustness is maximal.

2.1 Uncertain Durations

In real-life applications, it is often the case that the exact duration of a task, i.e. the time it takes to be processed, is not known in advance, and only some uncertain knowledge is available. Such knowledge can be modelled using a *triangular fuzzy number* or TFN, given by an interval $[a^1, a^3]$ of possible values and a modal value a^2 in it. For a TFN A, denoted $A = (a^1, a^2, a^3)$, the membership function takes the following triangular shape:

$$\mu_A(x) = \begin{cases} \frac{x-a^1}{a^2-a^1} & : a^1 \leq x \leq a^2 \\ \frac{x-a^3}{a^2-a^3} & : a^2 < x \leq a^3 \\ 0 & : x < a^1 \text{ or } a^3 < x \end{cases} \tag{1}$$

In the job shop, we essentially need two operations on fuzzy numbers, the sum and the maximum, which should extend the corresponding operations on real numbers using the *Extension Principle*. While it is straightforward to extend the sum in this manner, computing the maximum can become cumbersome and, most importantly, the set of TFNs is not closed under the resulting operation. For the sake of simplicity and tractability of numerical calculations, it is usual to approximate the maximum by a TFN which is relatively easy to compute, although there is no total consensus among authors on which approximation should be used. Another issue that must be taken into account is that no natural total ordering exists in the set of TFNs, making it necessary to adopt a ranking method to compare different makespan values. Which ranking method should be chosen is a controversial matter, with several proposals co-existing in the literature [13, 15].

Finally, the membership function of a fuzzy number can be interpreted as a possibility distribution on the real numbers, which allows to define its expected value [16]. It also constitutes an alternative view of the TFN as an upper envelope encoding a family of probability distributions [6], providing a framework to select a probability distribution consistent with the TFN (an essential step in the simheuristics approach). In particular, we will consider the uniform probability distribution that is bounded by the support of the TFN. This transformation is motivated by several results from the literature (see [17, 18]) that justify the use of TFNs as fuzzy counterparts to uniform probability distributions and model-free approximations of probability distributions with bounded support.

2.2 Fuzzy schedules

A fuzzy schedule does not provide exact starting times for each task. Instead, it gives a fuzzy interval (a TFN) of possible values for each starting time, provided that tasks are executed in the order determined by the schedule. In fact, it is impossible to predict what the exact time-schedule will be, because it depends on the realisation of the task's durations, which is not known yet. It is however possible —provided the right choice for TFN arithmetic approximations is made— to guarantee that for each possible configuration of task durations, the starting and completion times and the makespan of the executed schedule will lie in the corresponding fuzzy intervals. These fuzzy times also provide information on which starting and completion times are more likely to occur.

This is the basis for the semantics for fuzzy schedules proposed in [19] by which solutions to the fuzzy job shop should be understood as a-priori solutions, also called baseline or predictive schedules in the literature [5]. When tasks are executed according to the ordering provided by the fuzzy schedule we shall know their real duration and, hence, obtain a real (executed) schedule, the a-posteriori solution with deterministic times. Clearly, it is desirable that a fuzzy solution yields reasonably good executed schedules at the moment of its practical use.

3 A Simulated Genetic Algorithm for the FJSP

According to the semantics of fuzzy schedules described above, a solution to the FJSP provides an order for task execution together with estimates of the starting and completion times of all tasks. Alternatively, the task ordering could be obtained by solving a deterministic counterpart of the problem (with identical precedence and resource constraints but deterministic durations). In both cases, the actual performance of a given task ordering once it is executed will depend on the exact realisation of each task's duration. If the assumption of simheuristics is right, in scenarios with moderate uncertainty, orderings of high-quality in the deterministic setting should also result in high quality solutions in the original uncertain setting.

This is the idea underlying our proposal: solve a deterministic counterpart of a given FJSP instance using a metaheuristic from the literature (a genetic algorithm, GA, in our case) and evaluate the obtained solution in the fuzzy setting. To this end, we simulate different configurations for task durations, so each simulation provides a deterministic instance of JSP. The task ordering provided by the GA can be applied to this deterministic instance, simulating the execution of a schedule with a deterministic makespan value. The average makespan across a set of simulations will allow to assess the actual quality of the solution provided by the GA.

The general scheme of our simulated algorithm, called *simGA* in the following, is inspired by the scheme for simheurists from [14] and can be seen in Figure 1. Given an instance of FJSP, its deterministic counterpart is obtained via defuzzification, substituting every uncertain duration by its expected value. The resulting deterministic JSP is then solved with a GA. In fact, the GA is run several times. Each run of the GA provides us with a feasible solution to the deterministic problem. This solution will be included in a pool of solutions if it satisfies an acceptance criterion that compares the solution's fitness, i.e., the deterministic makespan, to the fitness of the solutions (if any) previously generated and already in the pool. If the obtained solution is accepted, its quality as solution to the original fuzzy problem is estimated by its average performance across a small set of deterministic instances that represent possible realisations of the FJSP, obtained after a fast simulation process. The process iterates until a finishing condition is met (here, after MaxRuns runs of the GA), providing us with a set of "accepted" solutions. with a fast estimate of their performance in the fuzzy setting. These accepted solutions are then ranked according to their estimated performance in the fuzzy setting and filtered (in our case, we keep only the MaxElite best solutions). For this set of filtered elite solutions we refine the estimated performance by testing them on a larger set of simulated instances (obtained after a more intensive simulation process). Solutions are then re-ranked according to their updated performance, thus obtaining an ordered set of elite solutions to the FJSP.

As solving method for the deterministic JSP we use a standard GA with chromosomes codified as permutations with repetitions [20]. This GA has already been adapted to the fuzzy problem in [21]. Also, in combination with a local search procedure, it constitutes a state-of-the-art method for the FJSP [22].

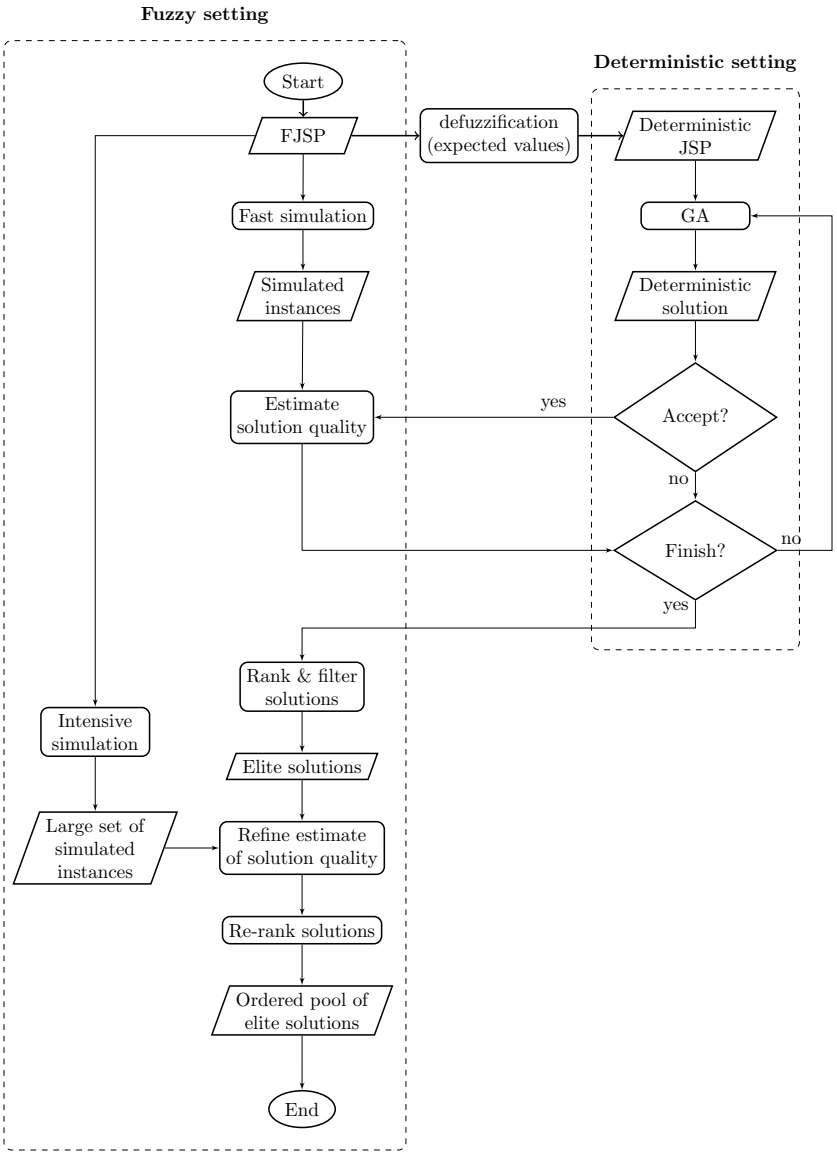


Fig. 1. General scheme of simGA.

Once a solution is obtained by the GA, it is accepted depending on how it compares to the best solution accepted so far in terms of makespan. If the new solution improves the best solution, it is accepted, but it can also be accepted if it is only worse by ϵ%, where ϵ is an acceptance parameter.

To simulate possible realisations of the FJSP we generate a real duration for each tasks following a probability distribution that is consistent with the possibility distribution defined by its fuzzy duration. Such probability distribution is obtained by applying the pignistic transformation as proposed in [23] based on the work on possibility-probability transformations from [24]. Each simulation provides an instance of deterministic JSP where a solution can be tested. Indeed, the task ordering established by that solution serves as a priority vector in an appending schedule generation scheme [25, 26], so each task is scheduled at the earliest possible time after all its predecessors both in its job and its machine, where all starting and completion times are calculated according to the simulated durations. Hence, the same ordering results in different makespan values across all simulations; the average is used as an estimate of the solution's performance.

4 Experimental Results

To test the proposed algorithm, simGA, we present some experiments on 23 benchmark instances for the FJSP from [15]. These are instances $\mathsf{La_F29}$, $\mathsf{ABZ_F8}$, $\mathsf{ABZ_F9}$, $\mathsf{Ta_F21–30}$ and $\mathsf{Ta_F41–50}$, fuzzified versions of well-known benchmarks for JSP which have been identified as challenging in [15]. We have run simGA on these instances, as well as its fuzzy equivalent, fGA, the GA from [21] with the same operators than the GA in simGA, but modified where necessary to handle TFNs using fuzzy arithmetic so it solves the fuzzy instance directly.

For both algorithms, the population size has been set to 100 individuals that evolve for 500 generations. The results below correspond to MaxRuns=30 runs of the GA in simGA and 24 runs of fGA. By doing so, we give fGA a total running time similar to simGA. simGA generates 10 simulated instances in the initial fast simulation and 100 instances in the intensive simulation phase. Finally, simGA keeps MaxElite=10 elite solutions. Accordingly, for fGA we keep the 10 best solutions in the population according to the ranking of fuzzy makespan values.

For each problem instance, each of the 10 solutions obtained with simGA (resp. fGA) has been tested on the 100 simulations to obtain 100 makespan values, so the average makespan across the 100 simulations is taken as a quality measure for a given solution. To compare simGA and fGA, for each instance we have run statistical tests with a significance level of 0.05. After checking for normality, a t-test concludes that there are no significant differences between simGA and fGA in 15 of the 23 instances, while simGA is significantly better than fGA in the other 8 instances.

To illustrate and compare the behaviour of both algorithms, Figures 2 and 3 show results obtained with fGA and simGA on instances $\mathsf{Ta_F45}$ and $\mathsf{Ta_F46}$ respectively. The figures correspond to the boxplots for the makespan values obtained on the large simulation set with the 10 best solutions of each algorithm (fGA in pink, simGA in green). They are representative of the cases where no significant difference exists between fGA and simGA (Figure 2) and the cases where simGA is signicantly better than fGA (Figure 3).

More detailed results for those instances where the algorithms present statistically significant differences can be found in Table 1. The first three columns refer to the GA that is used by simGA on the deterministic problem obtained after substituting each fuzzy duration by its expected value. Column "Best" contains the fitness value of the best elite solution, while column "Avg." contains the average fitness value across the 10 elite solutions. Notice that the fitness value is calculated on the deterministic counterpart of the FJSP instance, i.e., assuming all durations take their expected values, and it is the value used to guide the GA in the optimisation process and to accept (or reject) solutions in the "Accept?" step of simGA. The third column "RD" is a count of the elite solutions that are ranked differently depending on whether the fitness value or the simulated makespan is used. The fourth and fifth columns show the best ("Best") and average ("Avg.") simulated makespan value across the ten elite solutions provided by simGA. Finally, the last two columns show the best and average expected makespan values of the ten best solutions obtained by fGA. In this case, the makespan of the solutions (fuzzy schedules) is a TFN and its expected value is used as fitness value to guide fGA in the optimisation process.

Notice that for these instaces, where simGA outperforms fGA, the expected makespan values obtained with fGA are closer to the simulated makespan values of simGA than the deterministic

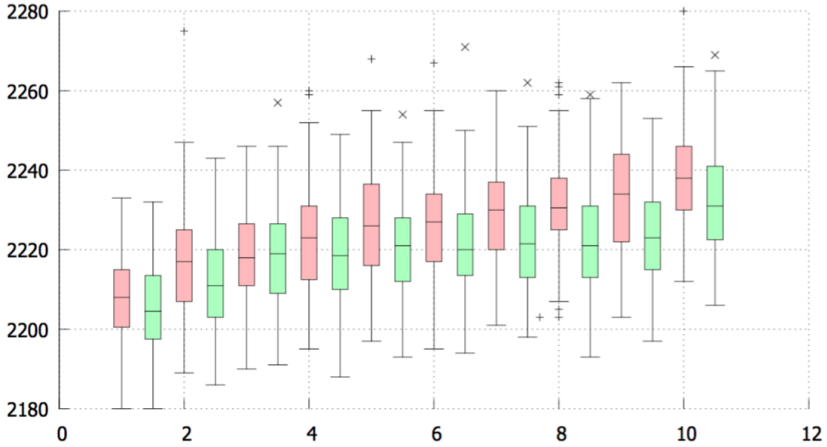


Fig. 2. Boxplot of makespan values obtained with the 10 best solutions of fGA and simGA on the simulation set for instance $\mathsf{Ta_F45}$, with no significant different between both algorithms.

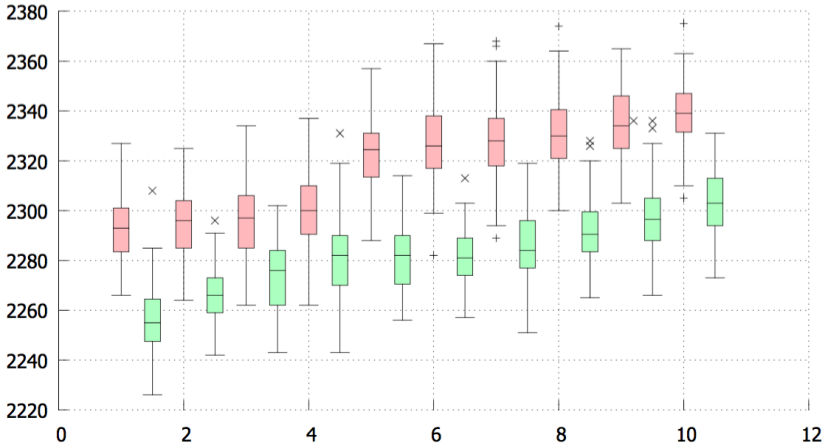


Fig. 3. Boxplot of makespan values obtained with the 10 best solutions of fGA and simGA on the simulation set for instance $\mathsf{Ta_F46}$, where simGA is significantly better than fGA.

Table 1. Results on instances for which simGA and fGA are statistically different

Problem	GA			simGA		fGA	
	Best	Avg.	RD	Best	Avg.	Best	Avg.
$\mathsf{La_F29}$	1217	1227.00	5	1233.48	1241.23	1222	1243.62
$\mathsf{Ta_F24}$	1736	1758.70	6	1760.34	1781.27	1752	1787.83
$\mathsf{Ta_F30}$	1661	1648.30	5	1684.83	1708.33	1677	1721.75
$\mathsf{Ta_F41}$	2262	2271.00	9	2277.68	2295.50	2260	2306.25
$\mathsf{Ta_F42}$	2170	2197.70	6	2197.64	2221.35	2205	2239.96
$\mathsf{Ta_F46}$	2231	2255.10	2	2256.74	2281.75	2263	2317.29
$\mathsf{Ta_F47}$	2121	2147.80	4	2146.74	2172.39	2136	2189.33
$\mathsf{Ta_F48}$	2140	2169.50	6	2164.59	2191.76	2158	2202.50

ones from GA. In fact, the deterministic makespan values from GA are always more "optimistic" (clearly smaller) than the simulated ones. This, together with the difference in ranking, suggests that the simulation phase in simGA does make a difference when evaluating and selecting solutions.

5 Conclusions and Future Work

We have tackled a fuzzy scheduling problem, the fuzzy job shop, with wide presence in the literature. The usual approach in the existing works consists in proposing metaheuristic methods that deal with the uncertainty in the problem in order to find a solution. This is however not done without a considerable effort, given the peculiarities of the fuzzy setting, and with little consensus on how to handle fuzzy numbers. We have proposed instead to adopt the simheuristics framework from stochastic optimisation, by which we handle uncertainty by integrating simulation with a metaheuristic method (a GA) for the deterministic job shop.

The resulting algorithm, simGA ,outperforms fGA, a GA that solves directly the fuzzy problem, in approximately 35% of the benchmark instances, with no significant difference between both algorithms in the remaining cases.

Future work would involve improving the sim-algorithm scheme, considering alternative ways of handling elite solutions and generating the simulations, as well as incorporating a more powerful state-of-the-art metaheuristic to solve the JSP. It might also be interesting how the simAlgorithms behave in terms of solution robustness, as previous works [26, 23] suggest that working directly on the fuzzy problem results in robust solutions.

Acknowledgements

This research has been supported by the Spanish Government under Grants FEDER TIN2013-46511-C2-2-P and MTM2014-55262-P and by the Principality of Asturias Government under FICYT Grant FC-13-COF13-035.

References

1. Pinedo, M.L.: Scheduling. Theory, Algorithms, and Systems. Fourth edn. Springer (2012)
2. Teoh, C.K., Wibovo, A., Ngadiman, M.S.: Review of state of the art for metaheuristic techniques in academic scheduling problems. Artificial Intelligence Review (2013)
3. Talbi, E.G.: Metaheuristics. From Design to Implementation. Wiley (2009)
4. Boussaïd, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. Information Sciences **237** (2013) 82–117
5. Herroelen, W., Leus, R.: Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research **165** (2005) 289–306
6. Dubois, D., Fargier, H., Fortemps, P.: Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. European Journal of Operational Research **147** (2003) 231–252
7. Wong, B.K., Lai, V.S.: A survey of the application of fuzzy set theory in production and operations management: 1998–2009. International Journal of Production Economics **129** (2011) 157–168
8. Dubois, D., Fargier, H., Prade, H.: Fuzzy constraints in job-shop scheduling. Journal of Intelligent Manufacturing **6** (1995) 215–234
9. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. IEEE Transactions of Fuzzy Systems **7** (1997) 557–569
10. Sakawa, M., Kubota, R.: Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. European Journal of Operational Research **120** (2000) 393–407
11. Zheng, Y.L., Li, Y.X.: Artificial bee colony algorithm for fuzzy job shop scheduling. International Journal of Computer Applications in Technology **44 (2)** (2012) 124–129
12. Li, J.q., Pan, Y.x.: A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem. International Journal of Advanced Manufacturing Technology **66** (2013) 583–596
13. Abdullah, S., Abdolrazzagh-Nezhad, M.: Fuzzy job-shop scheduling problems: A review. Information Sciences **278** (2014) 380–407
14. Juan, A.A., Faulin, J., Grasman, S.E., Rabe, M., Figueira, G.: A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Operations Research Perspectives **2** (2015) 62–72

15. Palacios, J.J., Puente, J., Vela, C.R., González-Rodríguez, I.: Benchmarks for fuzzy job shop problems. Information Sciences **329** (2016) 736–752
16. Heilpern, S.: The expected value of a fuzzy number. Fuzzy Sets and Systems **47** (1992) 81–86
17. Dubois, D., Foulloy, L., Mauris, G., Prade, H.: Probability-possibility transformations, triangular fuzzy sets and probabilistic inequalities. Reliable Computing **10** (2004) 273–297
18. Baudrit, C., Dubois, D.: Practical representations of incomplete probabilistic knowledge. Computational Statistics & Data Analysis **51** (2006) 86–108
19. González Rodríguez, I., Puente, J., Vela, C.R., Varela, R.: Semantics of schedules for the fuzzy job shop problem. IEEE Transactions on Systems, Man and Cybernetics, Part A **38** (2008) 655–666
20. Mattfeld, D.C.: Evolutionary Search and the Job Shop Investigations on Genetic Algorithms for Production Scheduling. Springer-Verlag (1995)
21. González Rodríguez, I., Vela, C.R., Puente, J.: A memetic approach to fuzzy job shop based on expectation model. In: Proceedings of IEEE International Conference on Fuzzy Systems, FUZZ-IEEE2007, London, IEEE (2007) 692–697
22. Puente, J., Vela, C.R., González-Rodríguez, I.: Fast local search for fuzzy job shop scheduling. In: Proceedings of ECAI 2010, IOS Press (2010) 739–744
23. Palacios, J.J., González-Rodríguez, I., Vela, C.R., Puente, J.: Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. Fuzzy Sets and Systems **278** (2015) 81–97
24. Dubois, D., Prade, H., Sandri, S.: On possibility/probability transformations. In: Fuzzy Logic. Volume 12 of Theory and Decision Library. Kluwer Academic (1993) 103–112
25. Sprecher, A., Kolisch, R., Drexl, A.: Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem. European Journal of Operational Research **80** (1995) 94–102
26. Palacios, J.J., González-Rodríguez, I., Vela, C.R., Puente, J.: Robust swarm optimisation for fuzzy open shop scheduling. Natural Computing **13** (2014) 145–156

Variable Neighbourhood Search for the Intermittent Traveling Salesman Problem

San Tu Pham and Patrick De Causmaecker

KU Leuven, Kulak
san.pham@kuleuven.be
patrick.decausmaecker@kuleuven.be

Abstract. In this paper a new variant of the Traveling Salesman Problem namely Intermittent Travelling Salesman Problem (ITSP) is presented. The ITSP originates from industrial polishing/drilling applications where the temperature of the work piece is taken into account. In this problem, the processing of a node must be divided into several periods and there is a certain time lap between two consecutive periods hence the name ITSP. We present a Mixed Integer Linear Programming (MILP) model for the problem and propose a Variable Neighborhood Search (VNS) to solve it. The best combination of VNS components for the ITSP is identified via experimental analysis. In addition, to understand the influence of the instance properties on the problem difficulty, analysis on various instance sets are conducted.

1 Introduction

In industrial polishing/drilling applications, heat generated during the machining plays an important role in the process. Consider a device, such as a laser, visiting a number of spots on a work piece, the aim of the visit is to machine the work piece at this particular spot. The heat generated from machining increases the temperature of the work piece locally, which might melt it down at a certain point. Consequently, machining cannot go on for ever and at some point in time the device has to leave the spot, to come back to continue only after a certain time lap.

Our problem consists of determining the best path to process the work piece which does not let the temperature at any point in the work piece exceed the maximum allowed value. Due to this temperature constraint, some spots have to be visited a number of times. In this work, we model and solve the problem as a variant of the Traveling Salesman Problem (TSP) [1] where some nodes have to be visited multiple times and there are time constraints between consecutive visits, hence the name Intermittent Traveling Salesman Problem (ITSP).

TSP is a well-studied problem in the field of combinatorial optimization with several variants. An extended review of TSP can be found in [2] [3]. However, to the best of our knowledge, the ITSP is the first problem tackling with constraints like temperature constraints. We hereby go through some TSP variants that appear to share some similarities to the ITSP and point out the differences with it. The first one is the TSP with multiple visits (TSPM) [3] - where a node must be visited a number of times. The TSPM has no time constraint between visits. The second one is the TSP with Time Windows [4] where each node is associated with a time window. No work in TSPTW has considered multi visits. A further scope takes us to the Inventory Routing Problem [5], in which each node (retailer) has a periodical (e.g. daily) requirement and an inventory capacity that should not be exceeded. The problem consists of finding the best shipping policy to deliver a product from a common supplier to several retailers, subject to the vehicle capacity constraints, product requirements and inventory capacities of the retailers. The shipping policy includes deciding which vehicle to serve which retailer, when to serve and vehicle traveling paths. Inventory constraints seem to be similar to temperature constraints. However, in the IRP, vehicle traveling routes are within one discrete period (e.g. a day). Therefore route scheduling doesn't affect retailer inventory levels, in constrast with the ITSP where traveling time directly affects node temperature.

Setting the number of visits of all nodes to 1 reduces the ITSP to a general TSP [6]. Hence, the ITSP is NP-hard, which makes it hard to solve large instances using exact methods. In this paper, a Mixed Integer Linear Programming (MILP) model for the ITSP are proposed. We then present a Variable Neighborhood Search (VNS) to solve the problem. The best components of the VNS are selected through statistical analysis of experimental results.

The rest of the paper is organised as follows. A MILP model is given in section 2. In section 3, a detailed description of the VNS is presented. The experimental results and analysis are reported in section 4 and finally conclusions and future work follow in section 5.

2 MILP model

Consider a graph $G = (V, E)$ with n nodes corresponding to n spots on the work piece. Each node has a processing time $p_i \geq 0$. Each arc $(i, j) \in E$ has a weight $c_{ij} > 0$ corresponding to the traveling time from node i to node j. To simplify the model, we assume that at the beginning of the process, the temperature of each node is 0. The node temperature increases during the processing linearly, i.e. one degree per time unit. After being processed for p seconds, the temperature of the spot is p degrees. While not being visited, spots cool down at the rate of one degree per time unit. Each material has a maximum temperature that must not be exceeded during the processing. We call this maximum temperature B. Due to the temperature constraints, node i with processing time $p_i \geq B$ must be visited multiple times and the processing time of each visit must be smaller than or equal to B. From now on, such nodes are mentioned as $multi-visit$ nodes while the other nodes are mentioned as $single-visit$ nodes. Two virtual nodes 0 and $n+1$ are created to represent the departing depot and returning depot, $c_{0,i} = c_{i,0} = c_{n+1,i} = c_{i,n+1} = 0 \quad \forall i \in V'$.

Consider a set of visits $Q = \{1, .., m\}$, each visit $j \in Q$ is a continuous process without interruption and is associated with a node $k \in V$, the model consists of the following variables:

- t_j represents the time at arrival of visit $j \in Q$.
- s_{jk} represents the processing time of visit $j \in Q$ at node $k \in V$.
- q_{jk} represents the temperature of node k at arrival of visit j.
- y_{jk} is a binary variable, $= 1$ iff node k is visited during visit j, 0 otherwise.
- x_{jkl} is a binary variable, $= 1$ iff node k is visited during visit j and node l is visited during visit $j + 1$, 0 otherwise.
- bl_{jk} and br_{jk} are binary variables used for linearizing the temperature constraint corresponding to q_{jk}.

The objective is to minimize the arrival time of the final visit. The model is as follows:

$$\text{Minimize } t_m$$

subject to

$$t_{j+1} \geq t_j + \sum_k s_{jk} + \sum_{k,l} d_{kl} * x_{jkl} \qquad \forall j \neq m \qquad (1)$$

$$\sum_j s_{jk} = p_k \qquad \forall k \qquad (2)$$

$$s_{jk} \leq B * y_{jk} \qquad \forall j, k \qquad (3)$$

$$y_{jk} = \sum_l x_{jkl} \qquad \forall j, k \qquad (4)$$

$$y_{jk} = \sum_l x_{j-1,lk} \qquad \forall j \neq 0, k \qquad (5)$$

$$y_{0,0} = y_{m,n+1} = 1 \qquad (6)$$

$$\sum_{jl} x_{jkl} \geq 1 \qquad \forall k \qquad (7)$$

$$\sum_{jl} x_{jlk} \geq 1 \qquad \forall k \qquad (8)$$

$$q_{0k} = 0 \qquad \forall k \qquad (9)$$

$$q_{j+1,k} = \max\{0, q_{jk} + 2 * s_{jk} - (t_{j+1} - t_j)\} \qquad \forall j, k \qquad (10)$$

$$q_{jk} + s_{jk} \leq B \qquad \forall j, k \qquad (11)$$

$$x_{jkl} \in \{0, 1\} \qquad \forall j, k, l \qquad (12)$$

$$y_{jk} \in \{0, 1\} \qquad \forall j, k \qquad (13)$$

where (1) represent the constraints between arriving time and traveling distance, and also play the role of subtour elimination constraints. (2) represent the total processing time of nodes. (3) are the linking constraints between s and y, also the upper bound constraints for the processing time of a single visit. (4, 5) are linking constraints between x and y. (6) are constraints for the first and final visits bound with the first and final node. (7) and (8) are flow constraints. (9) and (10) are temperature construction constraints. (11) are maximum temperature constraints.

Since (10) are not linear constraints, the binary variables bl and br are used for linearizing the constraints as follows:

$$-B * bl_{jk} \leq q_{j+1,k} \leq B * bl_{jk}$$
$$\forall j, k \tag{14}$$
$$-B * br_{jk} \leq q_{j+1,k} - q_{jk} - 2 * s_{jk} + t_{j+1} - t_j \leq B * br_{jk}$$
$$\forall j, k \tag{15}$$
$$-B * (1 - bl_{jk}) \leq q_{jk} + 2 * s_{jk} - (t_{j+1} - t_j) \leq B * (1 - br_{jk})$$
$$\forall j, k \tag{16}$$
$$br_{jk}, bl_{jk} \in \{0, 1\} \tag{17}$$

3 Variable Neighbourhood Search for the ITSP

Variable Neighborhood Search (VNS) [16] is a metaheuristic framework which is based on the systematic change of the neighborhood during the search in order to escape from the valleys containing local optimal. Two main components of VNS are local search and perturbation. From the current solution, a local search procedure is utilized to get to the local optimal of the search space valley containing it while perturbation (or shaking step) makes a big jump from this local optimal to a far distant neighbor, hopefully to escape from the valley. The general VNS framework can be found in figure 1.

VNS framework

Input: A set of neighborhood structures N_k, $k = 1..., k_{max}$
$x \leftarrow$ construct the initial solution
Repeat the following until the stopping condition is met
(1) Set k $k \leftarrow 1$
(2) Until $k = k_{max}$ repeat the following steps
(a) *Shaking*. Generate a point at random from the k^{th} neighborhood of x
(b) *Local search*. $x'' \leftarrow$ *Local search* (x')
(c) *Move or not*. If x'' satisfies the acceptance criteria, set $x \leftarrow x''$, and continue the search with N_1 ($k \leftarrow 1$); otherwise, set $k \leftarrow k + 1$

Fig. 1: VNS framework [16]

In order to apply VNS for the ITSP, the following components have to be defined: initial solution constructor, local search procedure, perturbation procedure and acceptance criterion. The initial solution is achieved by constructing a Hamiltonian path using a *Cheapest Insertion* algorithm [7]. Regarding to the three remaining components, we propose different options for each of them and aim to find the best combination for the ITSP. All components are presented in the following subsections.

3.1 Local search

Local search exploits the search space by moving from the current solution to a close neighbor if it satisfies some acceptance criteria. The goal of local search is to get to the local optimal of corresponding search space valley.

The main ingredient of local search is neighborhood move. Thanks to the similarity in TSP and ITSP solution structures, we use neighborhood classes from TSP: 2-opt [8], 3-opt [9] [10], or-opt, remove-and-reinsert [19]. For each class, a corresponding local search procedure is developed.

The ITSP moves are quite different from the original TSP moves. To present ITSP moves, we firstly present the ITSP solution encoding. An ITSP solution is encoded by an ordered set of visits $Q = \{1..m\}$. Each visit $j \in Q$ corresponds to a continuous process at a node $k \in V$. A multi-visit node has more than one corresponding visits in a solution. Each visit includes its position in the solution, the corresponding node id, the processing time of this visit (s_j), the temperature of the node at the beginning of the visit (q_j), the visited time (t_j) and the waiting time (w_j). The temperature constraint must be satisfied at all nodes, i.e. $q_j + s_j \leq B \quad \forall j \in Q$. The objective value of the solution is the total processing time or the visited time of the final depot.

Unlike the TSP, everytime an ITSP solution is modified, the temperature, the visited time and the waiting time must be recalculated, which is very costly. Therefore, we propose an estimation to check if the ITSP move is good or not before actually proceeding with the move. We illustrate this estimation using the 2-opt move as follows. Consider the TSP 2-opt move where edges AC and BD are replaced by AB and CD (Figure 2), the move yields benefit if $c_{AC} + c_{BC} - (c_{AB} + c_{CD}) > 0$. Therefore, checking if a move is beneficial in TSP is $O(1)$. In ITSP, due to the temperature constraints, the temperature, visited time and waiting time of all nodes starting from A have to be recalculated in order to check if a move yields benefit. We estimate the new objective value by recalculating the waiting time and visited time at node C, $C+1$, $C+2$ and D, $D+1$ and $D+2$ only since those nodes are most likely affected by the move. If the estimated objective value is smaller than the original one, the move might be beneficial, we then proceed with the move and recalculate the whole solution to get the exact objective value. A similar scheme is applied for other moves as well.

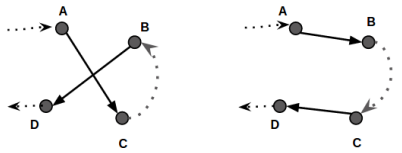


Fig. 2: 2-opt move

After each move, we do an "ITSP post processing" as follows: if visit j has the waiting time $w_j > 0$, we consider moving a part of its processing time s_j to other visits of the same node to see if the total waiting time can be reduced. To speed up the process, for each node, a list of corresponding visits is kept.

Based on those neighborhood moves, we propose two local search algorithms for VNS:

Variable Neighborhood Descent (VND) The VND local search [16] makes use of many different neighborhoods so that they can compensate each other. For each neighborhood move, a corresponding hill climbing local search is developed. The VND iteratively selects a neighbour structure, calls the corresponding local search to obtain its local optimum and updates the best solution correspondingly. The neighbour structures used in the VND are 2-opt, 3-opt, or-opt and remove-and-reinsert.

Late Acceptance Hill Climbing (LAHC) [17] In this strategy, a list of best solutions is kept and in each iteration, a non-improving solution might be accepted if it is better than one solution in the list.

3.2 Perturbation

Perturbation plays an important role in guiding the search to exploit new regions of the search space. Starting at the local optimal given by the local search procedure, perturbation makes a big move to get to a distant neighbor in order to escape from the corresponding valley. Therefore, in the perturbation step, a big neighborhood move is required. We propose two strategies for perturbation including hyperedge exchange [16] [15] and ruin-and-recreate [18].

Ruin-and-Recreate: A Ruin-and-Recreate ($R\&R$) move [18] includes two phases: the ruining phase removes some nodes from the tour while the recreating phase reconnects those nodes into the tour. The R & R for the ITSP is conducted as follows: In the ruining phase, the first node to be removed is chosen randomly, while the other ones are chosen from the list of its closest neighbour. Afterward, the recreating phase reinserts those nodes back into the tour using best insertion strategy. The strength of R & R move is defined by the number of nodes being removed. The VNS framework increases the strength of R & R move at each iteration.

Hyperedge exchange [13] An hyperedge $H = (p_0, h)$ includes h consecutive edges and starts from node p_0. In hypereged exchange, two hyperedges $H_1 = (p_0, h)$ and $H_2 = (q_0, h)$ are removed from the tour leaving some nodes isolated. Those nodes are then reconnected to the tour randomly. A hyperedge exchange move is illustrated in figure 3.

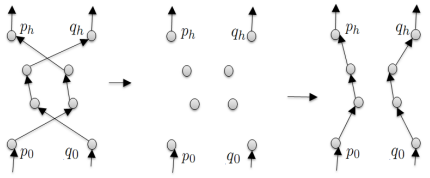


Fig. 3: Hyperedges exchange

Guided perturbation

In the first version of hyperedge perturbation, two exchanged hyperedges are chosen randomly. As an attempt to guide the algorithm to a promising search space, we adopt the guided perturbation of Burke et. al. [13] to the ITSP as follows. Instead of random shaking, we exploit the information of the current tour to guide the perturbation to promising areas. The quality of a hyperedge $H = (p_0, p_1, .., p_h)$ is evaluated by:

$$\sum_{i=0}^{h-1} c_{p_j,p_{j+1}} - min_{q\in\{1,2,...,n\}} c_{p_j,q} + \mu * \sum_{j=0}^{h} w_j$$

with w_j is the waiting time at visit j and μ is a normalizing factor.

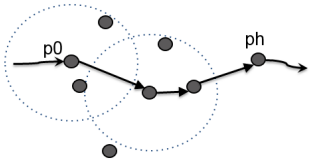


Fig. 4: Hyperedge evaluation

The first part of the formula measures the difference between the total distance from each visit to its successor in the hyperedge and to its closest neighbor. The large difference indicates that there might be a chance to obtain a shorter path by destroying and re-routing that path. The second part of the formula calculates the total waiting time along the hyperedge. In total, high value of the meric function implies poor quality hyperedge.

In the guided perturbation, the first hyperedge is chosen randomly from the list of $n/3$ worst hyperedges, n is the instance size. The second one is chosen randomly so that its first node belongs to the list of $n/4$ closest neighbors of the first node of the first hyperedge.

3.3 Acceptance criteria

Acceptance criteria strategy is an important component of the VNS which controls the balance between diversification and intensification of the search. The move from a solution to a better one

is call "downhill move". A search which only accepts "downhill moves" is an intense search and usually ends up in a local optimal. However, to diversify the search and explore potential search space, sometimes, a worse solution than the current one is also accepted. Such move is called "uphill move". Our acceptance criteria is based on Simulated Annealing (SA) algorithm [11]. SA is a simple yet effecient heuristic for combinatorial optimization problems, especially for TSP [12] since it controls the trade-off between diversification and intensification efficiently. In SA's acceptance criteria, "uphill moves" to solutions having higher costs are also be accepted occasionally. This acceptance is doing under the guidance of a control parameter called the temperature. High temperature means the algorithm is more likely to accept "uphill moves". By the beginning of the algorithm, the temperature is set to a high value, which causes more "uphill moves" to be accepted. At each iteration, the temperature is reduced gradually, leads to less "uphill moves" to be accepted.

4 Experimental results and analysis

4.1 Test instances and experimental settings

The datasets used in the following experiments are generated systematically with different property settings. An ITSP instance has three properties: size, density and bound ratio. Instance sizes are varied based on the purpose of the test. Density property implies the proportion of multi visit nodes in an instance. There are three density settings: I - dense, II - average and III - sparse. In dense instances, more than 90% nodes are multi-visit, while sparse instances contain single-visit nodes only. Bound ratio indicates the difference between maximum temperature B with the average traveling time $\bar{C}$ between nodes. There are two bound ratio settings: RI where $B = 0.5 * \bar{C}$ and RII where $B = 4 * \bar{C}$. In setting RI, in which the average traveling time between nodes is considerably smaller than the maximum temperature B, waiting time is likely to contribute for a larger portion of total time compared to setting RII.

All algorithms are implemented in Java. The MIP model is solved by Gurobi 6.0.5 [14]. All tests are performed on a Core i7, 2.9GHz Intel machine with 7.7 GB of RAM.

4.2 Selecting VNS best components

In this section, we investigate the VNS performance with different component combinations. Detailed options for two components local search (L), perturbation (P) are listed below:

- Local search: VND (L1), LAHC (L2)
- Perturbation type: VNS random (P1), VNS guided (P2), Ruin and Reinsert (P3)

The combination of the two above components forms 6 algorithms which are named by their components. E.g. algorithm $L1\_P1$ uses VNS random as perturbation and VND as the local search procedure.

In this experiment, generated instances with the size of 100 nodes are used. To compare the performances of the algorithms, we calculate the deviation from their best solutions to the solution generated by 3-opt local search. For each instance, each algorithm is run 10 times then the average objective value of the best solution and the time taken to achieve it are reported.

The first remark from the result (figure 5) is that algorithms using VND (L1) as local search component perform better the ones using LAHC (L2) in general. The second remark is that the VNS guided (P2) outperform the others perturbations. It proves that the perturbation process makes a good use of the information of the current tour. Since $L1\_P2$ is the best amongst those combinations, it is used in all latter experimental investigations.

4.3 Comparision between MILP and VNS

To check how close the VNS can get to the optimal solutions, we generate a set of random instances which size ranges from 10 to 50 nodes. Time limit for Gurobi is 800 seconds and for VNS algorithms is 100 seconds. For MILP, the upper bounds (best incumbents) and lower bounds of optimal solutions are reported.

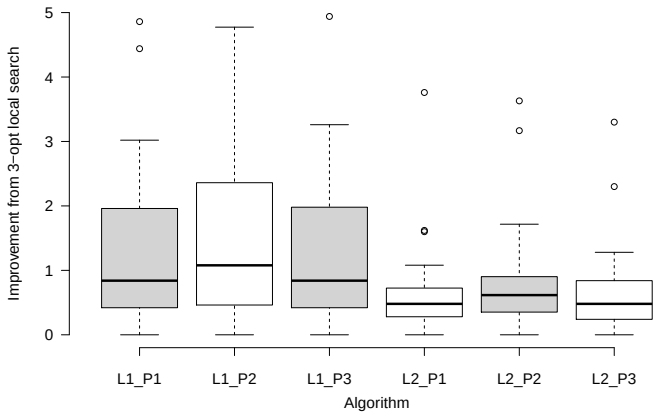
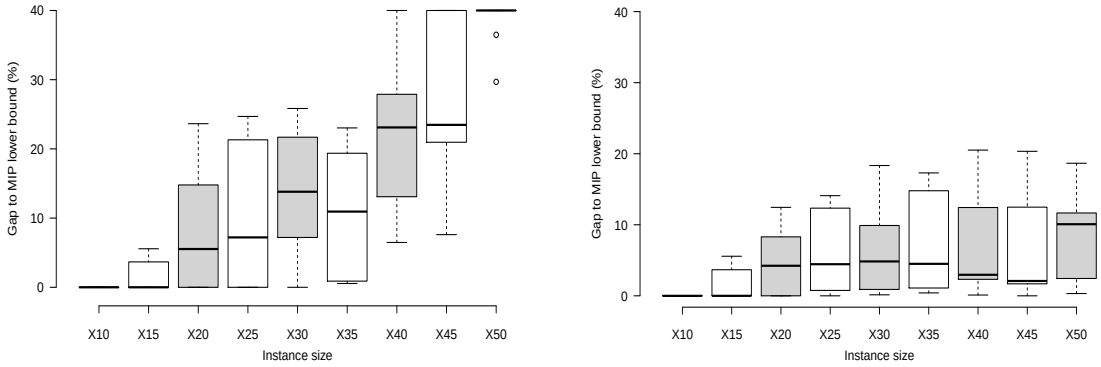


Fig. 5: Performance of VNS with different component combinations



(a) Gurobi (b) VNS

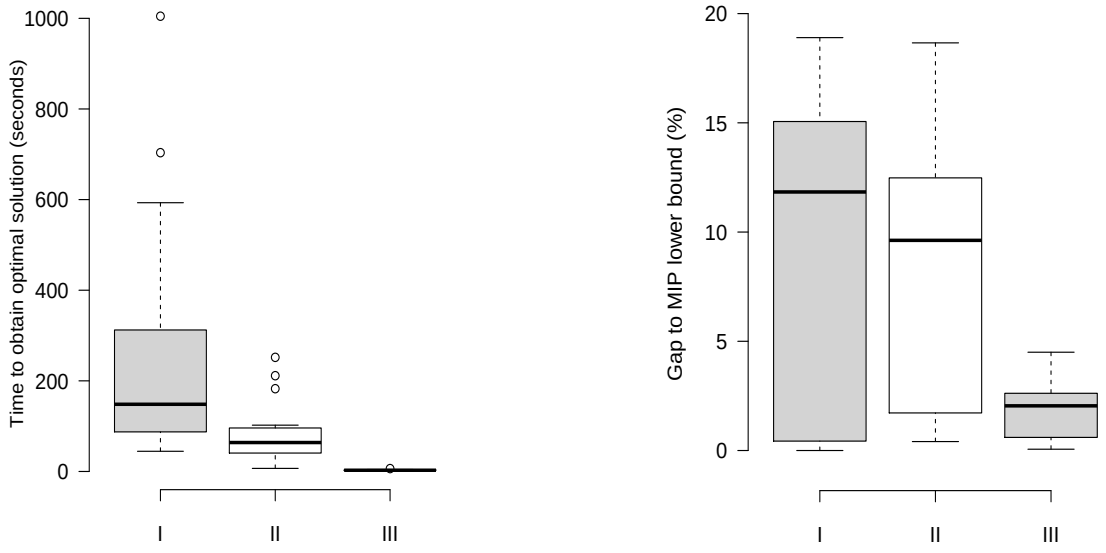
Fig. 6: Gap from Gurobi and VNS best objective solution to MIP lower bound

Figure 6 shows the gap between (a)the best solutions of VNS and (b)the best solutions of Gurobi to the MIP lower bounds given by Gurobi. As shown in the figure, Gurobi closes the gap for small instances only (less than 15 nodes), which is predictable. In those instances, VNS gets to the optimal solution as well. For large instances, MIP is not able to produce good solutions. For some instances, no solution is given (which is represented by 40% gap in the figure). In contrast, the mean value of VNS gap is less than 5% at instances up to 45 nodes and less than 10% at larger instances. This shows that VNS is suitable for large instances, hence is suitable for real world applications. This also suggests that the MILP is quite good at obtaining lower bounds for the ITSP.

4.4 Instance difficulty

In this section, we analyse the difficulty of test instances regarding to their properties as presented in section 4.1. Since MIP can only obtain optimal solutions for small instances on which VNS close the gap easily, in this section we perform the experiment on two datasets. On the first set containing small instances (13 nodes), we record the time taken by Gurobi to obtain the optimal solution. On the second set containing big instances (40 to 50 nodes), the gap between VNS best solutions and MIP lower bounds given by Gurobi are reported. In each set, 60 instances are generated with different density and bound ratio settings. The result of the VNS is the average of 10 runs. The time limit for Gurobi is set as 2000 seconds in this experiment.

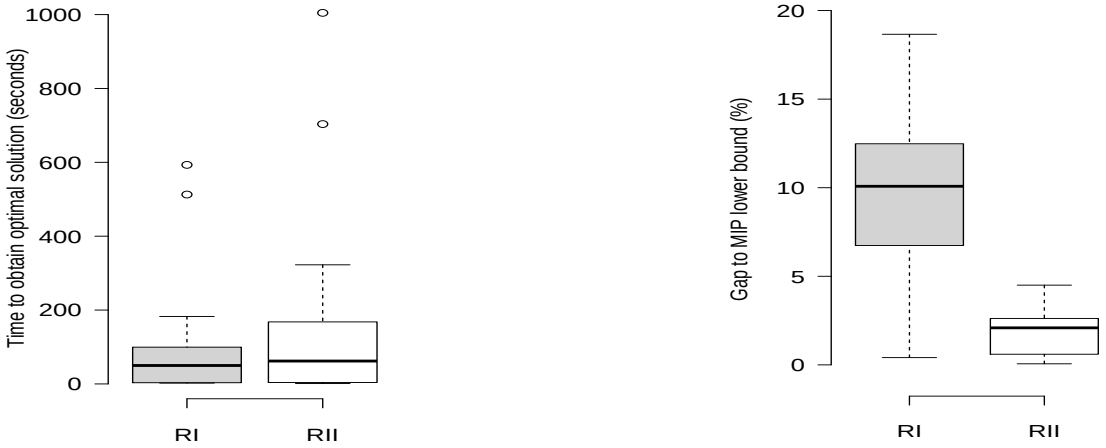
Figure 7 shows the comparison on different density settings. From the figure 7a we can see that density of instance affects MIP performance significantly. In details, the more multi-visit nodes an instance has, the more difficult it is to be solved. Especially for density setting III, which reduces



(a) MILP - time to obtain optimal solution

(b) VNS gap to MIP lower bound

Fig. 7: Comparison between different density settings



(a) MILP - time to obtain optimal solution

(b) VNS gap to MIP lower bound

Fig. 8: Comparison between different bound ratio settings

the ITSP to the TSP, Gurobi solves all instances in less than 1 second. Similarly, the NS encounters difficulties in closing the gap in dense instance as well (figure 7b).

Figure 8 shows the comparison regarding to the bound ratio property (the ratio between the maximum temperature B and the average traveling time between nodes $\bar{C}$). Solutions of instances with RI where $B = 0.5 * \bar{C}$ have more waiting time than solutions of instances with RII setting where $B = 4 * \bar{C}$. From the figure we can see that there are no significant difference in Guribo performance at different bound ratio setting. However, VNS performance relies heavily on this setting. The VNS performs better in setting RII.

5 Conclusion

In this paper, we presented a new variant of the TSP which raises from industrial polishing/drilling applications where the processing of a node has to be divided into multiple visits. A MILP model

was presented. We proposed a Variable Neighborhood Search approach with several options for each component. Through experimental result analysis, the best combination of the VNS components for the ITSP were identified. The experimental results showed that our algorithm can get close to optimal values on small instances. For instances up to 70 nodes, the gap from our best solutions to the MILP lower bounds are less than 5%. An experimental analysis on instance properties were carried out which indicated that the density and bound ratio of instances affect the difficulty of the problem.

Since this is a new problem, no benchmark is available currently. The building of a standard benchmark with different instance settings and difficulty levels is necessary for further investigation of the problem. However, to evaluate the difficulty of instances, we need a good algorithm to obtain the optimal solutions or good lower bounds for big instances. Since the MILP presented in this paper seems to be promising in producing good lower bounds, we plan to elaborate this approach in the future work. Based on the result, we will build a standard benchmark for the ITSP.

Acknowledgment

Work supported by the Belgian Science Policy Office (BELSPO) in the Interuniversity Attraction Pole COMEX. (http://comex.ulb.ac.be), Research Foundation Flanders (FWO) and the Marie Curie ITN STEEP (Grant Agreement no. 316560, http://www.steep-itn.eu/steep/index.aspx).

References

1. Dantzig, George, Ray Fulkerson, and Selmer Johnson. "Solution of a large-scale traveling-salesman problem." Journal of the operations research society of America 2.4 (1954): 393-410.
2. Johnson, David S., and Lyle A. McGeoch. "The traveling salesman problem: A case study in local optimization." Local search in combinatorial optimization 1 (1997): 215-310.
3. Gutin, Gregory, and Abraham P. Punnen, eds. The traveling salesman problem and its variations. Vol. 12. Springer Science & Business Media, 2006.
4. Dumas, Yvan, et al. "An optimal algorithm for the traveling salesman problem with time windows." Operations research 43.2 (1995): 367-371.
5. Campbell, Ann, et al. "The inventory routing problem." Fleet management and logistics. Springer US, 1998. 95-113.
6. Garey, Michael R., and David S. Johnson. "Computers and intractability: a guide to the theory of NP-completeness. 1979." San Francisco, LA: Freeman (1979).
7. D.S. Johnson and C.H. Papadimitriou, Performance guarantees for heuristics. Ch. 5 in The Traveling Salesman Problem, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, (editors), Wiley & Sons, Chichester (1985)
8. G. A. Croes, "A method for solving traveling salesman problems" Operations Res. 6 (1958), 791-812
9. F. Bock, "An algorithm for solving traveling-salesman and related network optimization problems" unpublished manuscript associated with talk presented at the 14th ORSA National Meeting, 1958.
10. S. Lin, "Computer solutions of the traveling salesman problem" Bell Syst. Tech. J. 44 (1965), 2245-2269
11. Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. "Optimization by simulated annealing." science 220.4598 (1983): 671-680.
12. Laporte, Gilbert. "The traveling salesman problem: An overview of exact and approximate algorithms." European Journal of Operational Research 59.2 (1992): 231-247.
13. Burke, Edmund K., Peter I. Cowling, and Ralf Keuthen. "Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem." Applications of Evolutionary Computing. Springer Berlin Heidelberg, 2001. 203-212.
14. Gurobi Optimization, Inc., "Gurobi Optimizer Reference Manual", 2015, http://www.gurobi.com
15. Hansen, P., Mladenovíc, N.: An introduction to variable neighborhood search. In: S. Voss, S. Martello, I.H. Osman and C. Roucairol (eds.): Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, pages 433-458, Kluwer Academic Publishers, Boston, MA (1999)
16. P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. European Journal of Operational Research, 130(3):449?467, 2001
17. Burke, Edmund K., and Yuri Bykov. "The late acceptance hill-climbing heuristic." University of Stirling, Tech. Rep (2012).
18. Schrimpf, Gerhard, et al. "Record breaking optimization results using the ruin and recreate principle." Journal of Computational Physics 159.2 (2000): 139-171.
19. Nilsson, Christian. Heuristics for the traveling salesman problem. Tech. Report, Linköping University, Sweden, 2003.

Design day selection based on Pareto Multiobjective Optimization

S. Khelil[1], N. Khelil[2] and N. Zemmouri[3]

[1, 3] *LACOMOFA* Laboratory – Biskra Algeria

[1]*sara.khelil@yahoo.fr,* [3] *pr.zemmouri@gmail.com*

[2]. AM Laboratory – Biskra Algeria

[2] *khelilna@yahoo.fr*

Keywords: Design day, Multiobjective optimization, Pareto approach, NSGA-II, AMY, Building comfort.

1 Introduction

The architectural simulation plays an essential role in providing performance and energetic capability because it enables the rapid quantitative exploration of designs. The building's behavior simulation is an important phase in the design. It becomes valuable technique to understand and optimize enormous challenges. It depends on the values of yearly meteorological parameters variations. The difficulty is intensified by the fact that annual weather data are not easily available and building designers involved in performance simulations are not the ones responsible for weather information gathering and recording. Therefore, we need an alternative reduced weather data as "the design day", extracted from yearly weather information that can ensure shorter time and less complex simulation [1].

The design day is a real historical day, which reflects the natural hourly variations of meteorological parameters [2]. We select it from a complete set of weather data for a single day chosen from the AMY "Actual Meteorological Year" file. Theoretically, the design day is the day having the most adverse set of weather conditions to enable the design to meet the indoor comfort criterion all over the year when performing at their maximum capacity. It consists of 24 hourly values of climatic parameters [3].

In this work, we are aiming to select a design day in order to study the natural ventilation in a building located in the city of Biskra, which is chosen for its representativeness of the hot and arid environments in Algeria. It has a rigorous climate characterized by very hot, dry summer and cold winter. Its characteristics are unfavorable for achieving thermal comfort.

In the table Table.1, we present two daily weather parameters, temperature and wind speed, from the 2015 AMY weather file, such as for each parameter, the annual maximum, minimum and average are described. These weather parameters have an influence on the energy and ventilation performance of a building [4]. Besides acting as control criteria in the selection of a design day, these criteria offer clues for interventions to reduce discomfort in occupied zones.

Weather parameter	Annual maximum	Annual minimum	Annual average
Temperature T [°c]	44,9	-2,1	19,87
Wind speed V[m/s]	10,6	0,1	1,87

Table .1. Annual statistics of the Biskra 2015 AMY

The design day is selected from the 365 days in 2015. The selected design day weather file consists of detailed data of 24 hourly values of climatic criteria parameters: temperature, wind velocity.

According to our objective, the design day is selected from the 365 days in 2015 where the selection is based only on daily averages of temperature and wind speed parameters, thus the selected day is the day having the maximum temperature (t) and the maximum wind speed (ws). In this context, the problem is defined as research, from a set of possible year days, the day (d^*) that makes these two criteria in their maximum values. To optimize (maximize) simultaneously these criteria, the multiobjective optimization (MO) techniques are used in the problem of the design day selection.

In multiobjective optimization problems (MOP), we have two or more objective functions to be optimized at the same time, instead of having only one. As a consequence, there is no unique solution to multiobjective optimization problems, but instead, we are aiming to find all of the good trade-off solutions available (the so-called Pareto optimal set). A solution x_p is a Pareto optimal solution if no objective function can be improved without worsening at least one other objective function. Such solution is not unique, and the set of the Pareto optimal solutions are known as the Pareto front.

Several bio-inspired optimization techniques have been developed for MO problems, the most known are genetic algorithms (AGs). The nondominated sorting genetic algorithm II "NSGA-II" [5] is the most popular genetic algorithm for solving MOP. This algorithm can find multiple Pareto-optimal solutions in a multi-objective optimization problem and has the following three features: It uses an elitist principle, it uses an explicit diversity preserving mechanism, and it emphasizes non-dominated solutions.

In this paper, we have adopted NSGA-II [5] for the design day selection using two weather criteria (temperature, wind speed) simultaneously, in order to find good compromises (or trade-offs) instead of a single solution (global optimization). This has the advantage of providing to architects more choices at the step of decision making.

The figure Fig. 1 illustrates, in two dimensions, the Pareto fronts in the case of the design day selection. The design day is one of the days presented in the table Table .2.

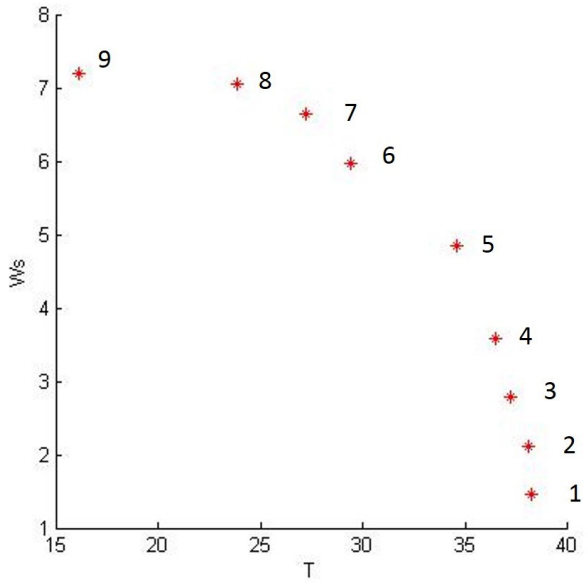


Fig .1. Pareto front found by NSGA-II, in the case of the maximization of two criteria: temperature (t) wind speed (ws).

The day (d^*)	1	2	3	4	5	6	7	8	9
	13 July	19 July	1 August	21 July	4 July	31 May	4 November	15 March	22 April
Temperature (daily average)	38.21	38.1	37.24	36.49	34.59	29.38	27.19	23.81	16.1
Wind speed (daily average)	1.47	2.12	2.8	3.58	4.85	5.99	6.65	7.07	7.22

Table .2. Set of the Pareto optimal solutions

References

[1] H. Tianzhen, K. Chou, Y. Bong (1999). A *design day for building load and energy estimation*. Building and Environment. pp 358-366.

[2] C. Iolanda, A. Florinela & al (2008). *Annual energetic behavior of buildings and the typical meteorological year*. 1st WSEAS International Conference on environmental and geological science and engineering.

[3] N. Piotr, J. Marcin & al (2013). *Annual Comparison of Untypical Meteorological Years (UMY) and Their Influence on Building Energy Performance Simulations*. 13th Conference of International Building Performance Simulation Association.

[4] H. Ishino & al (2005). *Proposal of seven-day design weather data for HVAC peak load calculation.* Ninth International IBPSA Conference.

[5] K.Deb, A.Pratap, S.Agarwal, and T.Meyarivan. (2002). *A fast and elitist multi-objective genetic algorithm: NSGA-II.* IEEE Transactions On Evolutionary Computation, 6 (2, 182–197).

Experimental Study of the Stability of Artificial Bees Colony When Solving Automotive Safety Integrity Levels Allocation

Youcef Gheraibia, Habiba Krimou, and Khaoula Djafri

Department of Computer Science, University of Souk-Ahras, Souk-Ahras, Algeria
{ youcef.gheraibia,H.Krimou,D.Khaoula }@univ-soukahras.dz

Abstract. The new functional safety standard ISO 26262 uses the concept of Automotive Safety Integrity Levels (ASILs) to classify the strictness of safety requirements and to reduce the risk provided by the malfunction of systems to a residual level. ASILs are five levels (QM, A, B, C, D) from the least strict A to the most strict D where QM means no special requirement. The main objective of the ISO 26262 is to assign the most appropriate ASILs to the failure modes of the system in order to reduce any potential risk to an acceptable level. But since ASILs allocation is a complex, constrained problem, it is quite hard to find the best allocation among the huge number of possible allocations especially if it was the case of a huge number of failure modes. In this paper, we present an experimental study that explores the impact of the failure modes' frequency on the stability of an ASILs allocation problem solver. The experiments were applied to an approach that uses the Artificial Bee Colony algorithm which is a nature inspired meta-heuristic and it explores the impact of the failure modes' frequency on its convergence and stability.

Mots-Clefs. Automotive Safety Integrity Levels, ABC, ISO26262, constrained Optimization.

1 Introduction

ISO 26262 is the international standard for the functional safety in the automotive industry and titled "Road Vehicle–Functional Safety". ISO 26262 is applied to the safety-related systems, these systems must comply with a safety life cycle defined in the same standard by going through an overall process [1]. The key component of this standards is Automotive Safety Integrity Levels (ASILs) which are five levels **QM, A, B, C, D** where **D** implies the most strict requirement while **A** implies the least strict and**QM** means no special safety requirements. The standard introduces an algebra concerning ASILs, and to formulate this algebra, ASILs were assigned integer values as follows: ASIL(QM)=0, ASIL(A)=1, ASIL(B)=2, ASIL(C)=3, ASIL(D)=4 [1]. ISO 26262 uses ASILs to classify the strictness of safety requirements to be allocated to the hazardous components of the automotive system. The failure of these hazardous components can cause the malfunction of the main system and subsequently represent a potential harm to the driver and road users. In order to design and develop reliant and safe products, ISO 26262 makes sure that the risk provided by the failure modes is reduced to a residual level. To do so, it requires a process that aims to ensure that the most appropriate allocation is assigned to these components based on the failure that they can provide.

The process begins with a hazard analysis to estimate any possible risk that may rise and subsequently, identify all the possibles hazardous failure modes that may provide that risk in order to allocate the most appropriate ASILs to those components. This allocation is based directly on the safety goals which represent the result of the hazard analysis and risk estimation. The safety goal is determined via a combination that gathers the probability of exposure, controllability and severity. Exposure is the state of being in an operational situation that can be hazardous if coincident with the failure mode under analysis[2]. Controllability is the ability to avoid a specified harm or damage through the timely reactions of the persons involved, possibly with support from external measures[2]. While severity is the estimation of the extent of harm to one or more individuals that can occur in a potentially hazardous situation[2]. Throughout the hazard analysis, for each failure mode of the automotive system, a safety goal is determined and an ASIL is assigned to that particular failure mode according to the safety goal determined earlier.

Assigning an ASIL to a hazard means that each failure mode that can possibly cause that hazard is required to meet that ASIL. But this leads to a serious development cost problem, since it would be expensive to make each component meet the highest safety requirements. To find a way out of this situation, ISO 26262 defined and introduced ASILs decomposition. The later allows the ASIL to be decomposed over the failure modes that provide the same hazard. The concept of ASILs decomposition reduced the cost of developing products considerably. ASILs allocation problem has numerous solutions that varies in safety requirements allocated to the the failure modes and subsequently varies in costs. For these reason, ASILs allocation has merited its hard nature and therefore it can be presented as hard combinatorial optimization problem of finding the most appropriate ASILs allocation that comes with the highest safety requirements and the least cost, among a set of feasible possible allocations. So finding the best ASILs allocation in a reasonable time is quite hard, especially if the number of failure modes was huge.

In this paper, we explore the impact of failure mode's frequency on the convergence and the stability of an ASIL allocation problem solver(see section 7). To do so, we present an novel approach that uses the swarm intelligence of Artificial Bee Colony (ABC) and solves the ASILs allocation problem (see section 6). To the best of our knowledge this is the first work that explores and presents an experimental study of the impact of different failure mode's frequencies on an ASILs allocation solver.

2 Background Study on ASILs Allocation Algorithms

Many contributions and techniques with different nature were proposed, to allocate ASILs to components of the automotive system under development, techniques can be categorized into exact solvers and search meta-heuristics. This section covers approaches which have been previously used to solve ASILs allocation problem. Starting with the exact solvers, the contribution presented in [3] introduces three off-the-shelf solvers to find all exact ASILs allocation solutions. Choco solver 3.2 [4] is the first solver proposed in the approach i.e. a java library for Constraint Satisfaction Problems (CSP) and Constraints Programming (CP). The approach uses Clafer 0.3.6 [5] which is a lightweight modeling language to improve the comprehension of the problem in the early stages of software development. Choco 3.2 interprets ASILs allocation problem into a CSP and cooperates with Clafer 0.3.6 to solve it using CP techniques. Developed at Microsoft, Z3 2.0 [6] is the second solver described in the approach, where ASILs allocation problem is solved after transforming it into Satisfeasability Modulo Theories (SMT) [7] problem. The last solver proposed in the approach was an Integer Linear Programming (ILP) solver that uses CPLEX [8] i.e. a file format that formulates a ASILs allocation problem in a natural, algebraic formulation, which allows the solver to read and therefore allocate ASILs to components of the system. The next exact approach described in [9] transforms ASILs allocation problem into a system of linear equations. The idea was to extract constraints from Fault Tree Analysis (FTA) [10] results, and use them to build the system of linear equations. The next step is to solve the system and there are many methods to do so, one drawback of using any of them is that the augmented matrix of the system is not always square. To overcome this problem, the Row Reduced Echelon Form (RREF) [11] of the augmented matrix is considered, the system is then solved using Gauss-Jordan elimination. The exact solvers provided the exact optimal assignments of ASILs to the components of the automotive system, however, due to the hard nature of the problem, scalability remains an issue in large systems. Therefore researches were directed to meta-heuristics.

In [1] a penalty based genetic algorithm is used to identify the optimal ASILs allocation. The search starts with a population of candidate solutions, the algorithm checks the feasibility of each candidate, and uses a penalty method to penalize the infeasible candidates. Fittest candidates are selected to go through mutation and uniform crossover to create new offspring, the offspring candidate is added to the population, if the new candidate is feasible and better than the current best solution. Then the algorithm replaces the current best solution with the offspring candidate, else the process re-starts all over again. In [12] the problem of allocating ASILs was converted into an Integer Linear Programming (ILP) problem, where ASILs are assigned integer values, the contribution uses a modeling framework equipped with an ILP solver. An EAST-ADL [13] model is created by the framework and used to generate Fault Trees. The constraints extracted from Fault Trees Analysis results and identified during the preliminary hazard analysis, are used in

addition to the engineer preferences as an input to the ILP solver. Another search meta-heuristic is presented in [14], where authors made use of an adopted version of Steepest Ascent Mildest Decent method (SAMD), which is a member of Tabu Search [15] family. The algorithms starts with a feasible solution and follows the steepest descent i.e. decrement the ASILs at each iteration until a local minimum is reached. In this case, the mildest ascent is followed to avoid getting trapped in the local minimum. The algorithms uses Tabu lists to prevent the search from re-visiting the previously visited solutions and stops when a termination criteria is met. In [16] a new member of swarm intelligence algorithms, Penguin Search Optimization Algorithm (PSOA)[17] is used to find an economic allocation to ASILs. The algorithm employs the concept of priority classes to assign low cost combinations a high priority. The process starts with a population of penguins, each penguin generates a set of neighboring solutions, for each neighboring solution, the algorithm checks it's feasibility and calculates its fitness. If the neighboring solution is better than the current best solution, it replaces the current best solution. The approach also uses the concept of oxygen reserve, which increments whenever a new good solution is found, and reset to 1 otherwise, in this case the penguin will have to explore another area of the search space.

3 ASILs Allocation Problem

ISO 26262 defines and uses the concept of ASILs to classify the strictness of safety requirements to be allocated to the failures modes of automotive systems. According to the standard, an ASIL is one of four levels to specify the item's or element's necessary requirements of ISO 26262 and safety measures to apply for avoiding an unreasonable residual risk, with D representing the most stringent and A the least stringent level [2]. ASILs are considered as an adaptation of Safety Integrity Levels (SILs) that were defined and used in IEC 61058 standard (international standard for Functional Safety of Electrical, Electronic, and Programmable Electronic (E/E/PE) Safety-Related Systems) [18]. ASILs are five levels QM, A, B, C, D where, as previously mentioned, A is the least strict level and D is the most strict and QM implies no special requirement. In ISO 26262 standards, an ASILs algebra is defined, and to formulate this algebra, levels from QM to D were assigned integer values as follows: QM=0, A=1, B=2, C=3, D=4. Each ASILs has a cost that differs on the used cost function, linear, logarithmic or experiential (See Table 1).

ASILs are assigned to the hazardous components and subsystems of the automotive system in order to ensure that the risk is reduced to a residual level. ASILs then are allocated to these components in a way that the highest safety requirements are guaranteed regarding the safety goal and the risk estimation. Therefore, each of the failure components of the system will be assigned the appropriate ASIL that fulfills the required safety. If the later is fulfilled, the risk will be reduced to an acceptable level which is the main goal of this whole process. So can a system meet the highest safety requirements, it would be fairly logical to assign the highest safety levels to safety goals of each failure mode. But, doing so, leads to a serious cost problem, because higher ASILs implies higher costs. For this purpose, ISO 26262 introduced ASILs decomposition which allows a safety critical system to meet a particular ASIL target without all its components having to individually meet that target[18]. Which means, ASILs decomposition allows the components that cause the hazard by failing simultaneously together to share that hazard's ASIL. Therefore, the ASIL will be decomposed over these components, instead of being assigned to each of them. As an example, consider a hazard provided by two components (C_1, C_2) that must meet ASIL C in order to avoid this hazard. By virtue of ASILs decomposition concept, these two components to do not have to meet ASIL C individually, instead, (C_1, C_2) might respectively meet (ASIL A, ASIL B) or (ASIL B, ASIL A) because 1+2=3 which is the same required ASIL. It is worth pointing that ASIL decomposition, participates in reducing the development cost as it is shown in the previous example, but reducing the cost still imposes a problem and researchers are still seeking for the allocation that decreases the cost even more.

ASILs allocation can be presented and formulated as a combinatorial optimization problem of finding the best allocation of ASILs to the hazardous components among a set of possible allocations. The best allocation must be found in the most reasonable time and comes with the highest safety requirements and the least development cost. ASILs allocation is then formulated

as follows:

$$\min f = \sum_{i=1}^{n} C(ASIL[i]). \tag{1}$$

Subject to: $\left(\sum_{j=1}^{mk} ASIL[j]\right) \geq k_{ASIL} \qquad 1 \leq k \leq l$
$ASIL_i, k_{ASIL} \in Z \cap ASIL_i, k_{ASIL} \in [0, 4]$
where:
n: number of the hazardous events.
l: number of ASIL allocation constraints.
mk: number of hazardous events in the k^{th} ASIL allocation constraint.
$C(ASIL_i)$: cost of the ASIL allocated to the i^{th} Hazardous event.
k_{ASIL}: ASIL requirement for the k^{th} ASIL allocation constraint.

4 Solution Representation and Quality Measurement

ASILs allocation problem belongs to the combinatorial type of problems, where the number of feasible solutions is huge and the goal is to find the optimal solution by maximizing or minimizing an objective function regarding a set of constraints. Constraints come in the form of minimal combinations of failure modes that if appeared simultaneously may lead to a hazard; these combinations are known as a Minimal Cut Set (MCS) [19]. The later is extracted and computed from Fault Trees Analysis (FTA) [10] results, using HIPHOPS (Hierarchically Performed Hazard Origin and Propagation Studies) [19] which is a safety tool that allows the automatic FTA. Solving ASILs allocation problem, means finding the most economic assignment of ASILs to the components of the system that reduces the development cost, while ensuring that the safety requirements are respected. Feasibility of the solution is determined by whether the provided ASILs allocation complies with the MCSs. The fitness of the solution is computed using the cost function and it simply equals the sum of the cost of the solution ASILs. As formerly stated, there are several cost scales, each scale implies different cost for each ASILs as follows:

Table 1. ASIL cost functions

Cost function	QM	A	B	C	D
Linear	0	10	20	30	40
Logarithmic	0	10	100	1000	10000
Experiential-I	0	10	20	40	50
Experiential-II	0	5	30	35	50

As an example, consider the following feasible allocation of ASILs S to a set of hazardous components where $S = 0, 0, 1, 0, 2, 1$ which implies $S = QM, QM, A, QM, B, A$. The cost of this allocation equals the sum of all ASILs' costs, which is based on the chosen cost function, in this example, the linear is applied. The cost is then calculated as follows: Cost (QM) + Cost (QM)+ Cost (A)+ Cost (QM)+ Cost (B)+ Cost (A), means 0+ 0+ 10+ 0+ 20+ 10= 40. Therefore the cost or the fitness of this allocation is 40.

5 Artificial Bee Colony (ABC) for ASIL allocation

5.1 ABC general description

Artificial Bee Colony is a member of swarm intelligence algorithms. It is a nature inspired, population-based, optimization meta-heuristic that mimics the behavior of real bees. The model of honey bee swarms consists of three essential components: food sources, employed foragers and unemployed foragers [20]. The two leading modes of behavior are the recruitment of nectar and the abandonment of nectar [20].

– **Food Sources:** defined by their profitability, its distance from the nest, richness, ease of extraction.

– **Employed foragers:** bees associated with a food source, and employed at. They exploit the food source to evaluate it and then carry information about this food source. This information will be shared later with other bees with a certain probability.

– **Unemployed foragers:** are bees on a continuous search for a food source. There are two types of unemployed foragers:
  - **Scouts:** bees searching and exploring their surrounding environment for food sources randomly.
  - **Onlookers:** bees waiting in the nest to choose a food source based on the information shared.

The most important part in the collective intelligence of bees is the exchange of information among them. This information which represents the profitability of the food source is shared in a very important part of the hive called "dancing area"[20]. Employed bees share the information gathered while exploiting the food source with the probability of profitability of that source by performing in the dance area. The dance which is called the "waggle dance" is performed is the dance area, hence, the onlookers will be able to watch a lot of dances and will get the opportunity to choose food sources. Employed foragers share their information with a probability proportional to the profitability of the food source, and the sharing of this information through waggle dancing is longer in duration [21]. An onlooker then can decide which food source to employ her self in. Hence, the information circulating about the food source is proportional to its profitability and the later is proportional to its recruitment [21]. The search process starts with unemployed bees called "scouts", exploring the environment randomly without any information about any food source. When a scout finds a food source, it employs her self in that source so it will be an "employed" in order to provide information about it. This information will be shared later with other bees in the dance area through the waggle dance. Onlookers are waiting in the nest to be informed through the waggle dance about all the possible food sources already found. The dance duration will determine the profitability of that food source. According that, onlookers will get to decide which food source to be recruited and which to abandon.

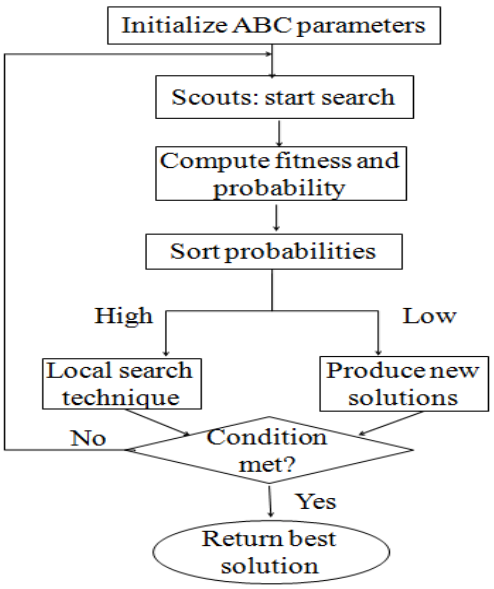


Fig. 1. ABC algorithm.

5.2 ABC for ASILs allocation

ABC has proven itself as a powerful nature inspired meta-heuristic when it comes to solve hard optimization problems. Through this work, a novel approach that used ABC algorithm to solve ASILs allocation is introduced. A food source represents a possible solution to the ASILs allocation problem. Therefore, the best food source represents the optimal ASILs allocation. The search process starts with hiring n number of artificial explorer bees called "scouts" that aim to explore the global search space for food sources (possible solutions). Each scout will randomly generate a possible feasible solution to ASILs allocation problem. The feasibility of the solution is checked by verifying the compliance of each solution with MCS. To this point, n feasible solution are generated, each scout then will employ her self in the food source (solution) she has generated and becomes an employee. Employees will provide information about the solution which is represented by its fitness, and then by its probability. Each employee will calculate the fitness (cost) of her solution, which equals the sum of the cost of ASILs allocated to the failure components of the system. This fitness will be used to calculate the probability of each solution according to the following equation.

$$p_i = \frac{fit_i}{\sum_{n=1}^{N} fit_n} \tag{2}$$

N is the number of food sources (and the number of employed bees), fit is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i.

The solution with the best fitness (least cost) is picked and saved in the memory as *"best"*. After calculating the probability of her solution, each employee will share this probability in the dance area. The dance area will be represents via an array that contains the probabilities of all solutions where each column corresponds to a solution. probabilities will be in a descending order. In the case of ASILs allocation problem, the cost of a particular solution represents its profitability, therefore, higher probabilities means lower costs. Hence, onlookers will choose p higher probability solutions, so each onlooker will employ her self in one. The rest of the solutions will be abandoned and their employees will be scouts and generate new random solutions. After picking the p good solutions, onlookers will try to improve these solutions through the local search technique. Local search is limited with a number of iterations called "limit". If a solution exceeds this limit without getting improved it will be abandoned. However, if a solution, did actually got improved by the local search and gets better than *"best"* it will replace it. Local search consist of creating a neighborhood of each solution by applying some modification over it. A good solution to ASILs allocation problem from p is firstly compared with the other good solutions. If an ASIL is present in these solutions and not in that particular solution, it will replace another ASIL that was picked randomly in that solution. In this case, feasibility will be checked again after the modification. The whole algorithm will be executed a number of iteration until the optimal ASIL allocation is reached.

6 Experimentation and Results

6.1 Data Description

The contribution presented in this paper, was evaluated by simulated data, that is, a set of generated cut sets. The idea was to use a binary matrix to simulate the cut sets extracted from Fault Tree Analysis (FTA) results, the generated matrix is made of 100 rows and 50 columns, which means 100 cut sets and 50 failure modes respectively. The presence of a failure mode F_i where $i = \{1, ..., 50\}$ in a cut set C_j where $j = \{1, ..., 100\}$, is represented by 1, while the absence of the same failure mode is represented by 0. The density of failure modes i.e. the frequency of failure modes in the cut sets is manipulated according to a probability that indicates how frequent a failure mode in a certain number of cut sets. For more details an example is provided (see table 2).

The example shows that the cut set CS1 is an order 1 cut set with only one failure mode FM2, the CS3 is an order 2 cut set with 2 failure modes FM1 and FM3, while the CS5 is an order 3 with 3 failure modes FM1, FM2, FM3. The density of failure modes in CS1 is 33.33%, the density of failure modes in CS3 is 66.66% and the density of failure modes in CS5 is 100%. Once the binary matrix is generated, the ABC algorithm inspects an appropriate assignment of ASILs that minimizes the cost regarding safety requirements to be met, trough out 1000 iterations. Linear cost heuristic is considered here to calculate the fitness of each allocation

Table 2. An example of simulated data

	FM1	FM2	FM3
CS1	0	1	0
CS2	0	1	0
CS3	1	0	1
CS4	1	0	0
CS5	1	1	1
CS6	1	1	0

6.2 Results and Discussion

The results show that the application of our approach to the simulated data was very promising. The density of failure modes affected both of the quality of allocation and the convergence time of the ABC algorithm. Figure 2 shows a considerable decrease of the cost value of allocations found. The cost was reduced from 110 to 10 when applied to ASILs 1 cut set, and from 360 to 40 when applied to ASIL 4 cut set.

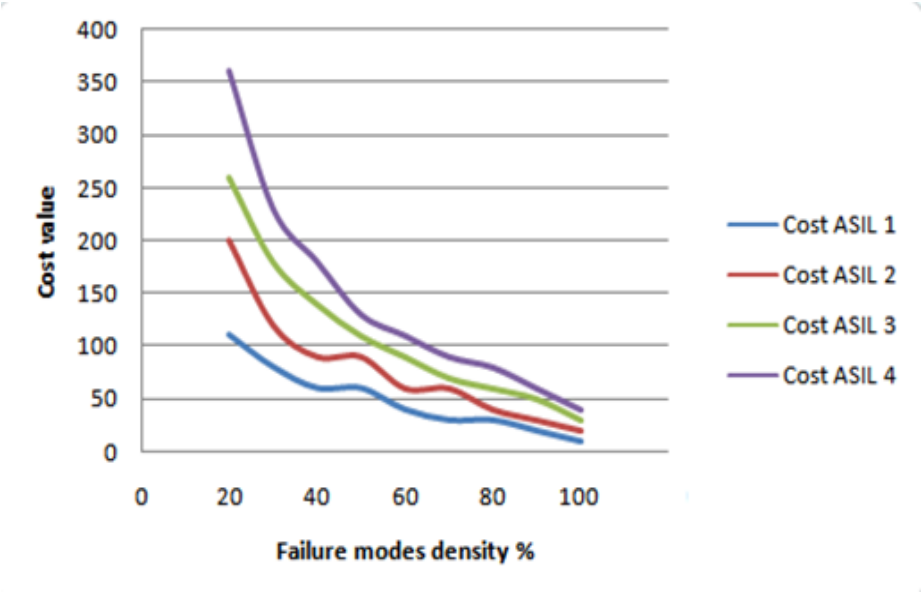


Fig. 2. Impact of failure modes frequency on the solution's quality

Figure 2 shows a considerable decrease of the cost value of allocations found, the cost was reduced from 110 to 10 when applied to ASILs 1 cut set, and from 360 to 40 when applied to ASIL 4 cut set.

Table 3, shows that the frequency of failure modes in a cut set of ASIL (D), has an impact on the convergence time of the ABC algorithm, which remains under a second in each and every case. The convergence time increases according to the density of failure modes, until it reaches the 40% density, then it decreases until 100% density is reached. The reason behind that is the algorithm faces a difficulty to find the best allocation with low density (from 20% to 40%), low density implies few combinations to find the appropriate assignment of ASILs. E.g. let $S = \{F1, F2, F3, F4\}$ be a set of failure modes of an ASIL(D) cut set, 20% density means only one failure mode out 4 failure modes is present in the cut set , which means it can only be assigned a 4 in order to find a feasible solution, this value may violet the cut sets which will extend the convergence time. While for 60% density, 3 failure modes out 4 are present in the cut set, which means there are more combinations to find feasible solution, more combinations implies low probability of cut set violation, therefore less time is required for the ABC algorithm to converge. for 100% density 4 failure modes out of 4

are present in the cut set, this also means that there is possible combinations to find an appropriate allocation without violating the cut set.

Table 3. Impact of failure modes frequency on the convergence time of ABC algorithm

Density (%)	20	30	40	50	60	70	80	90	100
Convergence time (MS)	387	431	575	508	435	494	386	365	357

7 Conclusion

ISO 26262 the functional safety standard in the automotive industry uses the concept of Automotive Safety Integrity Levels (ASILs) to represent the strictness of safety requirements to be assigned to the components of the system. ASILs allocation is a hard complex problem, and it was presented in this work as an optimization problem of finding the most appropriate ASILs allocation to the system's. The best ASILs allocation must be found in the most reasonable time, fulfills the highest safety requirements and guarantee the least development cost .In this paper, we presented an experimental study that explores the impact of failure mode's frequency on stability of an ASILs allocation solver and the case study was an approach that uses the Artificial Bee Colony algorithm. In our future work, we intend to extend the experimental study to include the other existing solvers and explore the impact of the failure modes' frequency on their convergence and performance.

References

1. D. Parker, M. Walker, L. S. Azevedo, Y. Papadopoulos, and R. E. Araújo, "Automatic decomposition and allocation of safety integrity levels using a penalty-based genetic algorithm.," in *IEA/AIE*, pp. 449–459, Springer, 2013.
2. I. ISO, "26262–road vehicles-functional safety," *International Standard ISO/FDIS*, vol. 26262, 2011.
3. A. Murashkin, L. S. Azevedo, J. Guo, E. Zulkoski, J. H. Liang, K. Czarnecki, and D. Parker, "Automated decomposition and allocation of automotive safety integrity levels using exact solvers," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 8, no. 1, pp. 70–78, 2015.
4. N. Jussien, G. Rochart, and X. Lorca, "Choco: an open source java constraint programming library," in *CPAIOR'08 Workshop on Open-Source Software for Integer and Contraint Programming (OSSICP'08)*, pp. 1–10, 2008.
5. K. Bak, Z. Diskin, M. Antkiewicz, K. Czarnecki, and A. Wasowski, "Clafer: unifying class and feature modeling," *Software & Systems Modeling*, pp. 1–35, 2015.
6. L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 337–340, Springer, 2008.
7. C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli, "Satisfiability modulo theories.," *Handbook of satisfiability*, vol. 185, pp. 825–885, 2009.
8. I. I. CPLEX, "V12. 1: Users manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.
9. M. S. Dhouibi, J.-M. Perquis, L. Saintis, and M. Barreau, "Automatic decomposition and allocation of safety integrity level using system of linear equations," *Complex Syst*, pp. 1–5, 2014.
10. W.-S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, "Fault tree analysis, methods, and applications a review," *Reliability, IEEE Transactions on*, vol. 34, no. 3, pp. 194–203, 1985.
11. S. J. Leon, *Linear algebra with applications*. Macmillan New York, 1980.
12. R. Mader, E. Armengaud, A. Leitner, and C. Steger, "Automatic and optimal allocation of safety integrity levels," in *Reliability and Maintainability Symposium (RAMS), 2012 Proceedings-Annual*, pp. 1–6, IEEE, 2012.
13. A. Consortium *et al.*, "East-adl domain model specification," *ATESST2, Deliverable D*, vol. 4, 2010.
14. L. S. Azevedo, D. Parker, M. Walker, Y. Papadopoulos, and R. E. Araujo, "Automatic decomposition of safety integrity levels: Optimization by tabu search," in *SAFECOMP 2013-Workshop CARS (2nd Workshop on Critical Automotive applications: Robustness & Safety) of the 32nd International Conference on Computer Safety, Reliability and Security*, p. NA, 2013.
15. F. Glover and M. Laguna, *Tabu Search*. Springer, 2013.

16. Y. Gheraibia, A. Moussaoui, L. S. Azevedo, D. Parker, Y. Papadopoulos, and M. Walker, "Can aquatic flightless birds allocate automotive safety requirements?," in *Intelligent Computing and Information Systems (ICICIS), 2015 IEEE Seventh International Conference on*, pp. 1–6, Dec 2015.
17. Y. Gheraibia and A. Moussaoui, "Penguins search optimization algorithm (pesoa)," in *26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Recent Trends in Applied Artificial Intelligence*, 2013.
18. L. da Silva Azevedo, D. Parker, M. Walker, Y. Papadopoulos, and R. Esteves Araujo, "Assisted assignment of automotive safety requirements," *Software, IEEE*, vol. 31, no. 1, pp. 62–68, 2014.
19. Y. Papadopoulos, M. Walker, M.-O. Reiser, M. Weber, D. Chen, M. Törngren, D. Servat, A. Abele, F. Stappert, H. Lonn, *et al.*, "Automatic allocation of safety integrity levels," in *Proceedings of the 1st workshop on critical automotive applications: robustness & safety*, pp. 7–10, ACM, 2010.
20. D. Karaboga, "An idea based on honey bee swarm for numerical optimization," tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.
21. V. Tereshko and A. Loengarov, "Collective decision making in honey-bee foraging dynamics," *Computing and Information Systems*, vol. 9, no. 3, p. 1, 2005.

Solving dynamic multi-objective problems using a copula-based estimation of distribution algorithm

A. Cheriet and F. Cherif

LESIA Labratory,Biskra University, Algeria
ahcheriet@gmail.com
foud_cherif@yahoo.fr

1 Introduction

The most real world problems are multi-objective, we have to add that many real world problems are also dynamic or continuously changing over a period. Changes may affect the object function, the problem instance, and/or constraints. In the literature of optimization in dynamic environments,researchers usually define optimization problems that change over time as dynamic problems or time-dependent problems [1].

A dynamic multi-objective problem can be represented as the following multi-objective optimization problem [2]. Let t be the time variable,$\mathbf{V}$ and $\mathbf{W}$ be n-dimensional and M-dimensional continuous or discrete vector spaces,$\mathbf{g}$ and $\mathbf{g}$ be two functions defining inequalities and constraint equalities, and $\mathbf{f}$ be a function from $\mathbf{V} * t$ to $\mathbf{W}$. A dynamic multi-objective minimization problem with M objectives is defined as:

$$\begin{cases} \min\limits_{v \in V} f = \{(f_1(v,t), \ldots, f_M(v,t)\} \\ s.t \quad g(v,t) \leq 0, h(v,t) = 0 \end{cases} \tag{1}$$

In this work, we present a new memory-based algorithm to solve a class of dynamic multi-objective problems. We use a copula-based estimation of distribution algorithm as a method of optimisation to get Pareto solutions in every generation of the algorithm. The obtained estimation model created using copulas will be utilized as a memory. We employ the memory created when a modification occurs during the execution of the optimisation process. The proposed EDA estimate in every generation the distribution of the best Pareto solutions obtained so far by creating a copula that describes the dependency between those solutions. We suppose that the created copulas can be used as a memory because they, not only save the last obtained best solutions but, save an explicit representation of the best solutions. The generated individuals from this explicit representation will be used as an initial population when a change occurs to the problem. We tested our proposal on the CEC2015 [3] benchmarks and we find that our algorithm gives good results.

An Estimation of Distribution Algorithm (EDA) is a class of the evolutionary algorithms that aims to estimate a distribution of a set of solutions usually the best ones, and use this estimation to generate new ones in every generation. The main difference between an EDA to another optimization algorithm is the manner of the estimation and the fashion of the algorithm implementation.

In Mathematics, a Copula is used to describe the dependencies between random variables. The proposed Copula-based EDA helps to create the estimator of the EDA. After finding the optimal solutions - like any classical optimization Algorithm - the generated Copula Model can be used when a change is detected in the problem. To validate our proposal, the proposed algorithm is performed to find the optimal solutions of a set of benchmark problems using the MOEA/D [4] as selection method.

2 The Proposed Copula-based Estimation of Distribution Algorithms

The Estimation of Distribution Algorithms uses many ways to estimate the distribution of the best solutions, we can find in [5] a good description of the used methods of estimation, however the use of Copula to estimate the distribution in EDA is a very strong idea to optimize complex problems [6–8]. We referred to copulas as "functions that join or couple multivariate distribution functions to their one-dimensional marginal distribution functions and as distribution functions whose one-dimensional margins are uniform." [9]. Many types of Copula heve been applied in various research

studies such as [10] and [11–13], in this paper, we will use an Archimedeon copula to find the best estimation.

Algorithm 1 Dynamic Copula-based EDA

$Q_0 \leftarrow$ Initialization(N_0)
$NDSet_0 \leftarrow$ Sorting(Q_0)
$P_0 \leftarrow$ SelectFromNDS(N)
$t \leftarrow 1$
while Not termination criteria **do**
  if Change Detected **then**
    $C_t \leftarrow$ EstimateMarginal(P_{t-1})
    $P_{tmp} \leftarrow$ GenerateSolutions(C_t)
  end if
  $NDSet_t \leftarrow$ ApplyMOEAD($P_{tmp} \cup NDSet_{t-1}$)
  $P_t \leftarrow$ SelectFromNDS(N)
  $NDSet_t \leftarrow P_t$
  $t \leftarrow t+1$
end while
Return $NDSet_t$,C

Like any evolutionary algorithm, our proposed method has two main steps; the *Selection* and the *Reproduction*, in the first step, we use the MOEA/D to select the best solution which will be used in the second step called the Reproduction, in this second step, the Copula is applied to estimate, then to regenerate new individuals. When a change occurs in the problem our proposal uses the obtained Copula model to generate individuals and use them as an initial population in the next generation. A pseudo-code of the Estimation of distribution algorithm using a Copula for a dynamic multi-objective problem illustrated in Algorithms 1.

3 Experimentation

To proof the efficiency of the proposed algorithm, a set of tests has been conducted using a set of benchmarks which are usually used in this kind of problems trying to test new solving algorithms in the area of dynamic multiobjective optimization. The work CEC2015 [3] provide a set of test benchmarks to compare the new algorithms with the classical algorithms. We used especially FDA4, FD5, HE2 and DMOP2 benchmarks in experimentations to proof the results given by our proposal.

4 Conclusion

In this work, we have proposed an Estimation of distribution Algorithm. The proposed algorithm used an estimation method which is the Copula and a very famous type which is the Archimedean one. Then we made an application of the new Copula-based EDA algorithm to solve Dynamic Multiobjective Optimization Problems. We used the obtained Pareto Solutions Estimated Model to generate new Pareto Solutions when the problem change. The Copula Model is viewed as a memory that conserves the characteristics of the PS, this vision is the motivation of using this algorithm in the Dynamic Multiobjective algorithm. A future work can be the use of the Copula Model as a memory in class of Real World Dynamic Multiobjective optimization problems.

References

1. Nguyen, T.T., Yang, S., Branke, J.: Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation **6** (2012) 1–24
2. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: test cases, approximations, and applications. Evolutionary Computation, IEEE Transactions on **8** (2004) 425–442
3. Helbig, M., Engelbrecht, A.: Benchmark functions for cec 2015 special session and competition on dynamic multi-objective optimization. Technical report, (Technical report)

4. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. Evolutionary Computation, IEEE Transactions on **11** (2007) 712–731
5. Hauschild, M., Pelikan, M.: An introduction and survey of estimation of distribution algorithms. Swarm and Evolutionary Computation **1** (2011) 111–128
6. Salinas-Gutiérrez, R., Hernández-Aguirre, A., Villa-Diharce, E.R.: Estimation of distribution algorithms based on copula functions. In: Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation. GECCO '11, New York, NY, USA, ACM (2011) 795–798
7. Wang, L.F., Zeng, J.C.: Estimation of distribution algorithm based on copula theory. In: Exploitation of linkage learning in evolutionary algorithms. Springer (2010) 139–162
8. Salinas-Gutiérrez, R., Hernández-Aguirre, A., Villa-Diharce, E.R.: Using copulas in estimation of distribution algorithms. In: MICAI 2009: Advances in Artificial Intelligence. Springer (2009) 658–668
9. Nelsen, R.B.: An introduction to copulas. Springer (2006)
10. Gao, Y., Peng, L., Li, F., Liu, M., Hu, X.: Eda-based multi-objective optimization using preference order ranking and multivariate gaussian copula. In Guo, C., Hou, Z.G., Zeng, Z., eds.: Advances in Neural Networks ISNN 2013. Volume 7952 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 341–350
11. Wang, X., Gao, H., Zeng, J.: Estimation of distribution algorithms based on two copula selection methods. Int. J. Comput. Sci. Math. **3** (2012) 317–331
12. Chang, C., Wang, L.: A multi-population parallel estimation of distribution algorithms based on clayton and gumbel copulas. In: Artificial Intelligence and Computational Intelligence. Springer (2011) 634–643
13. Wang, L., Guo, X., Zeng, J., Hong, Y.: Using gumbel copula and empirical marginal distribution in estimation of distribution algorithm. In: Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on, IEEE (2010) 583–587

The robust uncapacitated multiple allocation *p*-hub median problem

El-Ghazali Talbi[1] and Raca Todosijevic[1]

INRIA Lille-Nord Europe, 40 Avenue Halley 59650 Villeneuve d'Ascq, France
el-ghazali.talbi@univ-lille1.fr
racatodosijevic@gmail.com

1 Introduction

In this paper, we focus on the uncapacitated multiple allocation p-hub median problem (UMApHMP). The problem is defined defined on a complete symmetric graph $G = (N, E)$, where $N = \{1, 2, ..., n\}$ represents a set of nodes, while $E = \{(i, j) : i, j \in N\}$ denotes a set of arcs. No capacity restrictions on arcs $(i, j) \in E$ are imposed. Transportation cost per unit of flow on arc $(i, j) \in E$ is denoted as c_{ij}. For each O–D pair $i - j$, $i, j \in N$ the demand t_{ij} that has to be transferred from node i to node j is given. So, UMApHMCP consists of choosing exactly p nodes from the set N to be hubs, where p is given in advance, so that the total transportation cost is minimized assuming that any non-hub node may use any hub node to communicate with other nodes and the flows can be sent and received through more than one hub (multiple allocation scheme). The transportation cost from node $i \in N$, assigned to hub h_i, to node $j \in N$, assigned to hub h_j is calculated as:

$$d_{ij} = \gamma c_{ih_i} + \alpha c_{h_i h_j} + \delta c_{h_j j},$$

where parameters γ, α and δ are unit rates for collection (origin-hub), transfer (hub-hub) and distribution (hub-destination), respectively. In general, parameter α is used as a discount factor to provide reduced unit costs on arcs between hubs, so $\alpha < \gamma$ and $\alpha < \delta$. For solving UMApHMP many methods have been proposed in the literature (see e.g., [6] and references therein)

Most of the literature studies deal with the deterministic formulation of the problem. However, a disadvantage of a such deterministic problem is the fact that it requires that flows to be transferred must be known in advance. However, in practice the flow may vary during the time and thus an optimal solution for certain realization of flow need not to be so for the another realization of flow. For example, the flow (passengers) in the airlines transportation depends on the period of year, holidays time and so on. The similar is also true for the postal and cargo services, i.e., the demands may vary from day to day. Thus, the purpose of this paper is to provide a decision maker with one possible way how to deal with the uncertainty that obviously appear in the hub networks and also to propose an efficient way to create a solution which is robust for any realization of the flow.

1.1 Modelling uncertainity in UMApHMP

Since in the hub networks it is unlikely that the change in flow follows certain probability distribution, from our point of view, the uncertainity will be modeled more accurately by means of robust optimization. Hence, we model uncertainity in the following way. We assume that the input parameters t_{ij}, representing the flow to be transferred from i to j are subject to uncertainty. Further, we assume that t_{ij}, is bounded random variable with unknown distribution which takes values from $[t_{ij} - \hat{t}_{ij}, t_{ij} + \hat{t}_{ij}]$, where $\hat{t}_{ij} \geq 0$ represents a deviation from nominal coefficient value. Without loss of generality, we may assume that $t_{ij} \in [t_{ij}, t_{ij} + \hat{t}_{ij}]$.

Since UMApHMP may be modelled as a mixed integer programming (MIP), where flow parameters t_{ij} participate only in the objective function, in the rest of section we consider a mixed integer programming (MIP) optimization problem in its general form:

$$\min\{cx | x \in X\} \tag{1}$$

and present several robust measures.

We assume that each objective coefficient in (1) may take value from the interval $[c_j, c_j + d_j]$ and therefore we seek a solution which minimizes the maximum cost that may occur under such

assumption. Since that it is unlikely that all coefficients will change at once, Bertsimas and Sim [3] introduced a parameter Γ which expresses the level of conservatism of the solution, i.e., the number of coefficients that may change. Under such assumption the robustness of some solution $x \in X$ is calculated as the maximum regret:

$$G(x, \Gamma) = \max_{\{S|S \subset J, |S| \leq \Gamma\}} \sum_{j \in S} d_j |x_j| \tag{2}$$

where $J = \{j|d_j > 0\}$. Now, the robust counterpart of (1) is given (according to Bertsimas and Sim) as:

$$\textbf{(BS\_}\Gamma\textbf{)} \ \min\{f(x, \Gamma) = cx + G(x, \Gamma)|x \in X\} \tag{3}$$

Note that in the case of $\Gamma = 0$, we discard the changes in the objective coefficients, unlike the case of $\Gamma = |J|$, when all possible cost deviations are considered.

The main drawback of this approach is that a decision maker obtains a solution just for a given value of Γ and in reality he/she cannot predict accurately the value of Γ in advance. For example, he/she can solve the above model for $\Gamma = \Gamma_0$, but in practice Γ may receive any value between 0 and $|J|$. So, the question that naturally arises is whether the solution he/she obtained for the certain value of Γ would be also robust solution for any realization of parameter Γ. Moreover, whether the robust solution obtained for $\Gamma = \Gamma_0 > 0$ will be also robust also for $\Gamma < \Gamma_0$? Intuitively, this is unlikely the case. So, in order to resolve this issue we propose the approach explained below.

In order to find solution that performs well regardless of the number of coefficients allowed to change it is desirable to consider another problem

$$\textbf{(GBS\_1)} \ \min\{\sum_{\Gamma=0}^{|J|} f(x, \Gamma) - f(x^*_\Gamma, \Gamma)|x \in X\} \tag{4}$$

where x^*_Γ denotes the optimal solution of the problem (3). More precisely, the problem (GBS_1) seeks a solution for which total sum of deviations from the optimal solution values of problems (BS_Γ) is minimal. Note that in problem (4), it is not necessary to calculate in advance optimal solutions of each problem $\{f(x^*_\Gamma, \Gamma)|x \in X\}$ since the problem $\min\{\sum_{\Gamma=0}^{|J|} f(x, \Gamma)|x \in X\}$ has the same optimal solution as the problem (4). So, in what follows (GBS_1) will refer to $\min\{\sum_{\Gamma=0}^{|J|} f(x, \Gamma)|x \in X\}$.

The problem (GBS_1) may be generalized assigning the weight $p(\Gamma)$ to each function $f(x, \Gamma)$. In this case, the resulting problem denoted as (GBS_p) may be stated as:

$$\textbf{(GBS\_p)} \ \min\{\sum_{\Gamma=0}^{|J|} p(\Gamma) f(x, \Gamma)|x \in X\} \tag{5}$$

In this model each weight $p(\Gamma)$ may be interpreted as the probability that the change of Γ coefficients will occur. Note that the model (GBS_p) includes the model (GBS_1) as a special case. Indeed, setting $p(\Gamma) = 1$ for $\Gamma = \Gamma_0$ and 0 otherwise, the model (5) becomes $\min\{f(x, \Gamma_0)|x \in X\}$.

2 Solution approaches for robust UMApHMP

The presented models are suitable for solving just small instances by using general purpose MIP solver. So, in order to tackle large size instances we develop a basic variable neighborhood heuristic which can be applied for robust and standard UMApHMP.

A basic variant of Variable neighborhood search (called Basic VNS) [5] consists of executing alternately, one local search procedure (used to improve a solution) and one so-called shaking procedure (used to hopefully resolve local minima traps) together with the neighborhood change step. The whole process is iterated until a predefined stopping condition (e.g., maximum number of iterations or maximum CPU time) is met.

Before providing more details of our heuristic, we describe the solution representation for UMApHMP. Keeping in mind that once the set of hubs is known, the optimal node to hub allocation for each origin-destination pair can be found by determining the shortest path between them via hubs, we represent a solution of UMApHMP as a set H containing p hubs, i.e. $H = \{h_1, h_2, \ldots, h_p\}$.

The proposed Basic VNS starts the search from a solution built in a greedy manner. Namely, the initial p hubs are chosen as those whose maximum distances from any other node are the p smallest. More precisely, let $g(h)$ be the maximum distance of a node h from any other node, i.e.,

$$g(h) = \max\{c_{ih} | i \in N, i \neq h\},\ h \in N; \tag{6}$$

Then the nodes with the p smallest values of function g are taken as the initial p hubs.

The local search used within basic VNS is based on the exploration of the neighborhood

$$Interchange\_hub(H) = \{H' | H' \subset N,\ |H' \cap H| = p - 1\}.$$

This neighborhood structure contains all solutions obtained by replacing one hub from the set H by a non-hub node. The objective function value of a resulting solution H' is calculated from scratch, i.e., in $O(n^2 p)$ operations [1, 4]. The search for an improving solution is performed using the first improvement strategy, i.e., as soon as a better solution than the incumbent is detected, it is accepted as the new incumbent solution and the search is resumed starting from there. Such a local search procedure will eventually get stuck at a local minimum. Therefore, to hopefully resolve the encountered local minima traps, a shaking procedure is used. For the input, the Shaking procedure requires a solution H and a parameter k. At each of k subsequent iterations, the last solution H is replaced by a randomly chosen one from its $Interchange\_hub$ neighborhood. At the end, the solution H obtained in the kth iteration is returned as the output of our shaking procedure. The maximum value of k is specified by parameter k_{max}.

3 Computational results

The proposed Basic VNS algorithm is coded in C/C++ and executed on an Intel Core I7 with 2.8 GHz CPU and 16GB of RAM. For solving the mathematical formulations we use commercial CPLEX 12.6 MIP solver. The CPLEX 12.6 was also run on an Intel Core I7 with 2.8 GHz CPU and 16GB of RAM. For testing purposes the benchmark instances from the literature that are used i.e., the CAB and AP data sets from the ORLIB library [2]. We compare solutions obtained solving the nominal problem, the robust model based on robustness measure from [3] and the robust model based on the robustness measure we propose in order to detect how uncertainity and different robustness measures may affect a solution. The obtained results, as expected, confirms that an optimal solution of the nominal problem may be very poor quality solution of a robust problem and vice versa that an optimal solution of a robust problem may be very far from the optimal solution of the nominal problem. In addition, the results reveal the need for the general robust measure introduced in this paper.

4 Conclusions

In this paper we study uncapacitated multiple allocation p-hub median problem (UMApHMP) and propose several ways how to deal with the uncertainty that may occur. More precisely, to deal with uncertainty we use the robustness measure proposed in [3] and the robustness measure we proposed here which generalizes the robustness measure from [3]. The main advantage, of newly proposed measure over the previous one is that it is able to provide a solution which is robust regardless of the maximal number of parameters allowed to change.

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network flows: Theory, applications and algorithms. *Prentice-Hall, Englewood Cliffs,*, 1993.
2. J. E. Beasley. Obtaining test problems via internet. *Journal of Global Optimization*, 8(4):429–433, 1996.
3. D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical programming*, 98(1-3):49–71, 2003.
4. A. T. Ernst and M. Krishnamoorthy. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, 104(1):100–112, 1998.
5. P. Hansen, N. Mladenović, and J. A. M. Pérez. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.
6. J. Kratica. An electromagnetism-like metaheuristic for the uncapacitated multiple allocation p-hub median problem. *Computers & Industrial Engineering*, 66(4):1015–1024, 2013.

A Metaheuristic Approach to computing (approximate) Pure Nash Equilibria

N. Elgers[1], N. Dang[2], and P. De Causmaecker[3]

[1] Dept. of Computer Science
KU Leuven KULAK
niels.elgers@student.kuleuven.be
[2] Dept. of Computer Science
KU Leuven KULAK
nguyenthithanh.dang@kuleuven.be
[3] Dept. of Computer Science
KU Leuven KULAK
patrick.decausmaecker@kuleuven.be

Abstract. A pure Nash equilibrium is a famous concept in the field of game theory and has a wide range of applications. In the last decade, a lot of progress has been made in determining the computational complexity of finding equilibria in games. Deciding if a pure Nash equilibrium exists in n-player normal form games and several subclasses has been shown to be NP-complete. Current exact approaches are impractical and only able to solve small instances. In this paper, we apply three local search-based metaheuristics for solving the problem. Results on 280 randomly generated instances with different sizes show the clear outperformance of the metaheuristic approaches over the exact method based on Mixed Integer Linear Programming.

1 Introduction

Game theory is the study of mathematical models (games) that represent situations where multiple rational agents (players) receive payoffs which depend on the choices (actions) they make. In most contexts, each player strives to achieve an as high as possible payoff. Problems are modeled as games in an effort to predict how the players will behave. There are multiple solution concepts (equilibria) that attempt to predict this behaviour. Applications of game theory include, but are not limited to: the modelling of psychological theories; resource allocation, networking and artificial intelligence in computer science; competition and coexistence between species in biology and the modelling of many economic interactions. We consider non-cooperative games in which players make decisions independent of each other as opposed to cooperative games in which players may form coalitions or reach consensus as to achieve the best possible outcome. Players have full information of the game. This means that they know what actions are available to the other players and what payoff values result from them. Computer scientists study game theory to describe the computational complexity of computing equilibria in games.

It is possible to define a probability distribution over the actions available to a player. A set containing one such probability distribution for each player is called a mixed strategy. The expected payoff for a player is the payoff value a player can expect from playing the game when all players sample actions from their distributions. A mixed strategy is called a mixed Nash equilibrium (MNE) if no player can alter his distribution as to increase his expected payoff, assuming the other players their distributions are held fixed.

A pure strategy is a special case of a mixed strategy. It is when all players limit themselves to playing only one action, i.e., that action has a probability of one of being sampled. A pure strategy is called a pure Nash equilibrium (PNE) if no player can change his action to another action as to increase his payoff, assuming that all other players their actions are held fixed.

John Forbes Nash, Jr. showed that in each finite game, there exists at least one MNE [7, 14]. Later, it has been shown that computing a MNE is PPAD-complete [4]. Games often have no PNE and it has been shown to be NP-complete to decide if there exists at least one [6]. PNE are appealing, however, because they are simple to describe and execute and thus are a more plausible explanation of how agents may behave in a game.

For computing MNE in two-player games, as far as we are aware, there exist exact methods [9] and local search metaheuristics [3]. For multi-player games that admit specific settings, which includes two-player games, there exists a collection of polynomial-time approximation algorithms [5, 16].

As far as we are aware, there exists only one exact method (other than brute force) to decide if a PNE exists in multi-player games. It is a Mixed Integer Linear Programming (MILP) approach [17]. Similarly to MNE, there exists a collection of polynomial-time approximation algorithms [2, 1] for games that admit specific settings. We propose a metaheuristic approach to computing approximate PNE for general games.

Section 2 goes over preliminaries. Section 3 describes three metaheuristic algorithms to compute pure Nash equilibria. In section 4 we report how the algorithms perform and discuss them. We end with concluding remarks and ideas for future work in section 5.

2 Preliminaries

We first describe a handful of definitions that are needed to understand the continuation of this paper in subsection 2.1. We then describe the normal form description in subsection 2.2 by which any game can be described by. Games described this way admit exponential size in the number of players. Lastly, we describe succinct games in subsection 2.3. These are games that can be represented in size polynomial in the number of players and actions by using an alternative description.

2.1 Definitions

A game $G = (N, S, U)$ is a tuple in which N represents the set of players, S represents the possible outcomes of the game and U a set of functions for each player that maps an outcome to a payoff value for the player.

The players are usually represented as unique integers; e.g. $N = \{1, 2, ..., n\}$ in which n is the number of players that participate in the game.

Each player $i \in N$ has an action set $S_i = \{s_i^1, s_i^2, ..., s_i^{m_i}\}$. It represents the choices available for player i in the game. m_i is the number of available actions to player i. We denote $m = \max_{i \in N} m_i$. An outcome is represented by $s = \{s_1, s_2, ..., s_n\}$, in which $s_i \in S_i$. The set of all possible outcomes s is denoted by S. In other words, $S = \Pi_{i \in N} S_i$.

$U = \{u_1(s), u_2(s), ..., u_n(s)\}$ is the set of payoff functions. That is, functions $u_i(s)$ determining the payoff value for player i when all players choose the action proposed for them in the outcome s. For ease of notation, $u_i(s)$ is sometimes written as $u_i(s_i, s_{-i})$ in which s_{-i} represents the set of actions played by the players that are not i. Each player strives to obtain an as high as possible payoff value from the game.

An outcome s is called a pure Nash equilibrium (PNE) if and only if no player i can improve his payoff $u_i(s)$ by solely altering his action. Formally, s is PNE if and only if:

$$\forall i \in N, \forall s' \in S_i : u_i(s_i, s_{-i}) \geq u_i(s', s_{-i}) \tag{1}$$

The number of constraints in the PNE definition is polynomial in the number of players and actions, i.e. $O(m.n)$. It is therefore efficiently verifiable if an outcome is a PNE or not, as aspected from a NP-complete problem.

We say that an outcome s is an approximation of a PNE according to some value $\epsilon \geq 0$ (ϵ-PNE) if and only if:

$$\forall i \in N, \forall s' \in S_i : u_i(s_i, s_{-i}) \geq u_i(s', s_{-i}) - \epsilon \tag{2}$$

It is clear that if s is an ϵ-PNE then s is a μ-PNE as well, with $\mu \geq \epsilon$. It follows from the definition that s is a PNE if and only if it is a 0-PNE.

2.2 Normal form games

A normal form description of a game describes the game by way of a multidimensional array. Each dimension of the array represents a player. The size of each dimension equals the number of actions

available to that dimension its player. Each element of the multidimensional array is an outcome which has a payoff value for each player.

The total number of payoff values in a normal form game is exponential in the number of players. More specifically, the number of payoff values is equal to $n.\prod_{i \in N} m_i$ or in asymptotic notation $O(m^n)$.

As an example of a normal form game, consider table 1 that represents a game in which $N = \{1, 2, 3\}$; $\forall i \in N : S_i = \{s_i^1, s_i^2\}$ and $U = \{u_1(s), u_2(s), u_3(s)\}$. The values of $u_1(s)$,$u_2(s)$ and $u_3(s)$ are represented by respectively the first, second and third value in table 1 in the relevant outcome.

Consider the outcome $s = \{s_1^1, s_2^2, s_3^1\}$. We can see that it is a pure Nash equilibrium since solely alternating a player's action will always lead to a worse payoff value for him/her.

	s_3^1		s_3^2	
	s_2^1	s_2^2	s_2^1	s_2^2
s_1^1	$(-1,-1,-1)$	$(2,1,2)$	$(2,2,1)$	$(0,1,1)$
s_1^2	$(1,2,2)$	$(1,1,0)$	$(1,0,1)$	$(1,1,1)$

Table 1: A three player normal form game.

2.3 Succinct games

By the definition of a game, all games can be described by the normal form description. Some games, however, can be described more efficiently using other representations.

As an example, consider a game in which players play two player games with a subset of the other players. Such a game is called a polymatrix game and can be represented by a graph. Each node of the graph represents a player. An edge is present between players if they play a two player game. An edge is absent if the relevant players do not play a game. The payoff for a player in a polymatrix game for some outcome is the sum of payoffs from each two player game the player is involved in if the players play the action proposed by the outcome in all two player games. The number of two player games is $O(n^2)$ and each game has $O(m^2)$ payoff values. Therefore, a polymatrix game described by the graph description has $O(n^2m^2)$ payoff values which is more efficient than the normal form description. Games, such as polymatrix games, that admit polynomial size are called succinct games. They are of interest to us as we can not generate non-succinct games with a lot of players and/or actions. To see this, consider a normal form game with 10 players and 10 actions. This game will have 10^{11} payoff values. If each payoff value is a 64 bit number, then the game would be 800 gigabytes big! A polymatrix game with the same amount of players and actions in which all players play games with all the other players will be 304 kilobytes big using the graph description. If, in the polymatrix game, the number of players and actions is changed to 100, we would still use *only* 792 megabytes.

For a comprehensive database of games and generators for them we refer to GAMUT [15].

3 Computing approximate Pure Nash Equilibria using local search-based metaheuristics

We convert the problem of finding a pure Nash equilibrium into an optimization problem by defining an objective function f:

$$f : S \to \mathbb{R} : s \mapsto f(s) = \max_{i \in N, s' \in S_i} (u_i(s', s_{-i}) - u_i(s_i, s_{-i})) \tag{3}$$

The function represents the highest increase that one of the players can gain by solely switching his action, and can be computed in polynomial time $O(m.n)$. A solution $s \in S$ is a PNE if and only if $f(s) = 0$. s is an approximation for the PNE otherwise. If $f(s)$ is small enough, the approximation

can prove to be useful, for example, when it would cost a lot of money and/or effort to change to a better action while it would only lead to a very small payoff increase.

We do not intend to build powerful metaheuristics but rather show that existing, well-described metaheuristics are a feasible method to approach the problem. We implement three local search-based metaheuristics for solving the problem:

- Random Restart Hill Climbing (RRHC), in which Hill Climbing repeatedly restarts from a random solution. The First Improvement strategy, which accepts the first improved solution in the neighborhood, is used inside the Hill Climbing. Pseudo-code of the algorithm is presented in Algorithm 1
- Simulated Annealing (SA) [8]: we implement a basic version of SA that allows reheating. The initial temperature is set as 2000. After t_{length} algorithm iterations, the temperature is decreased by a factor of α. When the temperature reaches a lower bound of ϵ, it is reset back to the initial temperature. In this work, we set $t_{length} = 100$, $\alpha = 0.9$ and $\epsilon = 10^{-6}$. Pseudo-code of the algorithm is presented in Algorithm 2.
- Chained Local Optimization (CLO) [12]: this algorithm is a combination idea of Simulated Annealing and Local Search. The algorithm resembles the Iterated Local Search framework [11] with the Hill Climbing algorithm as the local search component and the acceptance criteria taken from the Simulated Annealing. Algorithm 3 shows the pseudo-code of the algorithm implemented in this paper. Values of SA's parameters are the same as the SA described above.

All metaheuristics use the same neighborhood structure. The neighborhood $\eta_k(s)$ of an outcome s is the set of all outcomes in which at most k players altered their action. The size of a neighborhood is then $O(m^k)$. The parameter k is set differently among the three metaheuristics: $k = 1$ for the Hill Climbing component inside RRHC and CLO, $k = 3$ for SA and the step of generating a random neighbor solution in the while loop of CLO. The initial solutions used are random. That is, a random action for each player. The algorithms also share the same stopping criteria: either when a maximum running time limit is reached or when a PNE has been found.

Algorithm 1 Random Restart Hill Climbing

Require: Game $G = (N, S, U)$, desired amount of running time T.
  $s \leftarrow$ random outcome $\in S$.
  $best \leftarrow$ hillClimbing(s)
  while time elapsed $\leq T$ **do**
    $s \leftarrow$ hillClimbing(random outcome $\in S$)
    if s is PNE **then**
      return s
    end if
    if $f(s) < f(best)$ **then**
      $best \leftarrow s$
    end if
  end while
  return $best$

4 Results

In this section, the performance of the three metaheuristics on different problem instance sizes is presented. The size of a problem instance depends on two parameters: the number of players (n) and the number of possible actions for each players (m). For each pair of n and m, we generate ten random instances using GAMUT [15], an extensive library of different types of games and generators for them.

We first compare the performance of the metaheuristics with the MILP method on instances that are small enough so that MILP can solve them within a reasonable time and/or memory limit in subsection 4.1. We then show how the metaheuristics perform on instances of succinct games that MILP can not solve due to time and/or memory limitations in subsection 4.2.

Algorithm 2 Simulated Annealing

Require: Game $G = (N, S, U)$, desired running time T, initial temperature t_0, decreasing factor α, temperature stage length t_{length}, neighborhood k, approximation function $f : S \to R$.

```
s ← random solution ∈ S
best ← s
t ← t_0
while time elapsed ≤ T do
  for i = 1..t_length do
    s' ← random solution ∈ η_k(s)
    if s' is PNE then
      return s'
    end if
    Δ ← f(s') − f(s)
    if Δ ≤ 0 then
      s ← s'
      if f(s') < f(best) then
        best ← s'
      end if
    else if random value ∈ [0, 1] < e^{−Δ/t} then
      s ← s'
    end if
  end for
  t ← αt
  if t < 10^{−6} then
    t ← t_0
  end if
end while
return best
```

Algorithm 3 Chained Local Optimization

Require: Game $G = (N, S, U)$, desired running time T, initial temperature t_0, decreasing factor α, temperature stage length t_{length}, neighborhood k.

```
s ← random solution ∈ S
s ← hillClimbing(s)
best ← s
t ← t_0
while time elapsed ≤ T do
  for i = 1..t_length do
    s' ⇐ random solution ∈ η_k(s)
    s' ← hillClimbing(s')
    if s' is PNE then
      return s'
    end if
    Δ ← f(s') − f(s)
    if Δ ≤ 0 then
      s ← s'
      if f(s') < f(best) then
        best ← s'
      end if
    else if random value ∈ [0, 1] < e^{−Δ/t} then
      s ← s'
    end if
  end for
  t ← αt
  if t < 10^{−6} then
    t ← t_0
  end if
end while
return best
```

4.1 Comparison of the metaheuristics with a Mixed Integer Linear Programming approach

We first compare the metaheuristics with the exact method based on Mixed Integer Linear Programming (MILP) proposed in [17]. To the best of our knowledge, this MILP is the only method (except brute-force) that exists in the literature to solve the pure Nash equilibrium problem in normal form games. The comparison is done on a dataset of 28 problem instance sizes, leading to a total number of 280 instances. The payoff values are randomly selected from the range of $[-100, 100]$. Therefore, the objective function f (3) value belongs to the range of $[0, 200]$.

In this experiment, we set a maximum running time limit of 60 seconds for the metaheuristics. For the MILP, we use CPLEX 12.6 with the limits of 4GB memory and one hour of running time. Results are shown in Table 2. For each instance size, we report the number of instances in which a PNE has been found and the average running time (in milliseconds) for finding a PNE. For instances where a PNE has not been found, we normalize the objective function values of the approximate solutions returned by the metaheuristics to the range of $[0, 1]$ and report the mean over those instances. Additionally, the number of instances that the MILP can not solve within the given memory and time limits are also presented.

We can see the clear outperformance of the metaheuristic approaches over the MILP:

- For problem sizes whose every instance is solvable by the MILP, a PNE is always found by the metaheuristic for every instance that has at least one PNE. Moreover, the time required by the metaheuristics to find a PNE is smaller with an order of magnitude when compared to the MILP. For instances where a PNE does not exist, the small objective function values given by the metaheuristics suggest good approximate PNEs. In fact, since the values returned by all the metaheuristics for each instance are exactly the same (except the last problem size with $n = 9$ and $m = 5$), we suspect that these approximate solutions are actually the optimal ones, although we can not prove this.
- For problem sizes with $n \geq 5$ and $m \leq 7$ (except the case where $n = 7$ and $m = 3$), the MILP runs out of either memory or time in most instances. Those are the cases where the metaheuristics are dominant not only on the running time, but also on the ability of finding a PNE.

4.2 Results of the metaheuristic approaches on large games

We continue by running the metaheuristics on instances with a lot of players and/or actions. As motivated in section 2.3, we need to use instances of succinct games. We chose polymatrix games as a succinct game. We generate 10 instances of 25 different problem sizes resulting in a total of 250 instances. In our n-player polymatrix game instances, each player plays $n-1$ two-player games with the other players. The payoff values of the two-player games are randomly selected from the range $[-100, 100]$. Consequently, since the payoff for a player is the sum of all payoffs he gets in the subgame, the payoff for a player lies in the range of $[100(1-n),\ 100(n-1)]$. Therefore, the objective function f (3) value belongs to the range of $[0, 200(n-1)]$.

In this experiment, we set a maximum running time limit of 10 minutes for the metaheuristics. For each instance size, we only report the approximations. It was only the case for $n = 10$, $m = 10$ that a few PNE had been found. Similarly to the small instances, we normalize the approximations to the range of $[0, 1]$. We observed that for the largest instances, the metaheuristics did not perform well. That is, within the time limit, not even one iteration of RRHC or CLO was finished. In other words, both algorithms only performed Hill Climbing until the time limit was reached. We reran some of those instances with an increased time limit of one hour but even then not one iteration was finished.

The reported approximations in table 3 look similar to those for the small instances. While we can say that the approximations for the small instances are good, it would be too short-sighted to say the same about the large instances. In a game with a lot of actions, the difference between the smallest payoff value to which a player can switch that is bigger than the current payoff and the current payoff is generally smaller than in a game with fewer actions. Therefore, the optimal approximation value is expected to be smaller. We suspect that the reported approximations are no way near the optimal value, although we have no way to prove this.

5 Conclusion and future work

The main aim of the paper is to show that existing metaheuristics are promising methods to approach the problem of finding a pure Nash equilibrium in game theory. From the experimental results on normal form games, it is clear that the considered metaheuristics, although being implemented in a quite basic and general form, perform much better than the exact MILP method [17]. No other exact methods have been considered and there are none as far as we are aware other than enumerating all the possible outcomes (brute-force).

For larger instance sizes, the normal form description causes the memory usage of the game to be too big to be generated. Despite the fact that succinct games are polynomial in size in function of the number of players and actions, the MILP approach still reduces the problem to a linear program that is exponential in size. This means that not only the solving time of the MILP will take exponential time, but the reduction as well. We therefore deem the MILP approach as entirely unfit for these problem sizes. On the other hand, because our metaheuristic approaches do not require an exponential amount of memory, we can find ϵ-PNE in instances of succinct games that have a lot of players and or actions.

All the problem instances and detailed results of the solving approaches used in this paper are publicly available at `https://github.com/ElgersNiels/Metaheuristic-approach-to-PNE` . We hope that it could serve as a first benchmark for applying metaheuristics on the problem of finding a pure Nash equilibrium.

For future work, we plan to improve the current metaheuristic approaches by exploiting knowledge from specific game types, such as the graph structure of polymatrix games. In that sense, one may design game specific heuristics that could make metaheuristics more powerful. Moreover, in the current experiments, no real intensive work has been done to tune the parameters of the metaheuristics yet, so the application of an automated parameter tuning tool such as irace [10] will be interesting.

A second line of future work is to improve the current benchmark dataset. It should be noted that games generated by GAMUT are random and are not necessarily hard to find (approximate) PNE in. It might be the case that there exist specific generator settings that produce significantly harder problems than in the average case, which has been found in SAT community. In SAT, there exist specific settings so that the generated propositional formula has a 50% chance to be (un)satisfiable. These instances are shown to be the hardest to solve [13]. Perhaps such parameter settings exist for games as well so that finding a PNE, if it exists, is hard. It might prove to be fruitful to reduce hard SAT instances – or other hard instances of other problems – to games and evaluating their hardness in an attempt to create benchmarks.

6 Acknowledgements

This work is funded by COMEX (Project P7/36), a BELSPO/IAP Programme. The authors would like to thank Prof. Danny Weyns and Dr. Rahul Savani for providing comments on how to improve the quality of this paper.

Bibliography

[1] Anand Bhalgat, Tanmoy Chakraborty, and Sanjeev Khanna. Approximating pure nash equilibrium in cut, party affiliation, and satisfiability games. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 73–82. ACM, 2010.

[2] Ioannis Caragiannis, Angelo Fanelli, Nick Gravin, and Alexander Skopalik. Efficient computation of approximate pure nash equilibria in congestion games. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 532–541. IEEE, 2011.

[3] S. Ceppi, N. Gatti, G. Patrini, and M. Rocco. Local search methods for finding a nash equilibrium in two-player games. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 2, pages 335–342, Aug 2010.

[4] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[5] John Fearnley, Paul W. Goldberg, Rahul Savani, and Troels Bjerre Sørensen. *Algorithmic Game Theory: 5th International Symposium, SAGT 2012, Barcelona, Spain, October 22-23, 2012. Proceedings*, chapter Approximate Well-Supported Nash Equilibria Below Two-Thirds, pages 108–119. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[6] Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. Nash equilibria: hard and easy games. *Journal of Artificial Intelligence Research*, pages 357–406, 2005.

[7] Albert Xin Jiang and Kevin Leyton-Brown. A tutorial on the proof of the existence of nash equilibria. *University of British Columbia Technical Report TR-2007-25. pdf*, 2009.

[8] Scott Kirkpatrick, Mario P Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[9] Carlton E Lemke and Joseph T Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423, 1964.

[10] Manuel López-Ibánez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. The irace package, iterated race for automatic algorithm configuration. Technical report, Citeseer, 2011.

[11] Helena R Lourenço, Olivier C Martin, and Thomas Stützle. *Iterated local search*. Springer, 2003.

[12] Olivier C. Martin and Steve W. Otto. Combining simulated annealing with local search heuristics. *Annals of Operations Research*, 63(1):57–75.

[13] David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of sat problems. pages 459–465, 1992.

[14] John F. Nash. A dissertation on non-cooperative game theory. *Princeton university*, 1950.

[15] Eugene Nudelman, Jennifer Wortman, Yoav Shoham, and Kevin Leyton-Brown. Run the gamut: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 880–887. IEEE Computer Society, 2004.

[16] Haralampos Tsaknakis and Paul G. Spirakis. *Internet and Network Economics: Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007. Proceedings*, chapter An Optimization Approach for Approximate Nash Equilibria, pages 42–56. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[17] Zhengtian Wu, Chuangyin Dang, Hamid Reza Karimi, Changan Zhu, and Qing Gao. A mixed 0-1 linear programming approach to the computation of all pure-strategy nash equilibria of a finite n-person game in normal form. *Mathematical Problems in Engineering*, 2014.

Table 2: Random Normal Form Game results

n	m	Found				Time Found (ms)				Approximation			Unsolvable
		RRHC	SA	CLO	MILP	RRHC	SA	CLO	MILP	RRHC	SA	CLO	MILP
3	3	6	6	6	6	1.79	0.72	0.15	141.67	0.033	0.033	0.033	0
3	4	8	8	8	8	1.52	0.56	0.26	279.77	0.11	0.11	0.11	0
3	5	4	4	4	4	3.23	0.59	0.13	3696.24	0.057	0.057	0.057	0
3	6	6	6	6	6	2.60	0.389	0.44	2240.06	0.052	0.052	0.052	0
3	8	5	5	5	5	1.05	0.39	0.57	2654.57	0.026	0.026	0.026	0
3	9	5	5	5	5	0.58	1.013	0.68	8122.44	0.032	0.032	0.032	0
3	10	8	8	8	8	4.86	2.01	2.47	3731.40	0.011	0.011	0.011	0
3	15	6	6	6	6	16.34	5.82	12.05	1850.26	0.014	0.014	0.014	0
3	20	5	5	5	5	19.18	36.81	19.15	5929.46	0.011	0.011	0.011	0
3	25	6	6	6	6	67.70	29.78	133.63	42524.34	0.0061	0.0061	0.0061	0
3	30	3	3	3	3	108.31	55.99	68.93	49762.25	0.0065	0.0065	0.0065	0
3	35	5	5	5	5	432.00	92.37	285.10	215015.91	0.0066	0.0066	0.0066	0
3	40	6	6	6	6	670.20	173.30	165.02	324330.39	0.0066	0.0066	0.0066	0
3	45	7	7	7	7	409.89	247.53	293.42	469457.35	0.0060	0.0060	0.0060	0
3	50	8	8	8	8	921.45	297.51	788.01	818496.07	0.0012	0.0012	0.0012	0
5	3	8	8	8	8	1.92	0.82	0.51	10712.12	0.026	0.026	0.026	0
5	4	6	6	6	6	2.86	7.11	0.84	10423.76	0.021	0.021	0.021	0
5	5	7	7	7	7	3.16	2.27	5.64	87200.74	0.024	0.024	0.024	0
5	6	9	9	9	9	24.38	16.08	31.87	633587	0.012	0.012	0.012	0
5	7	5	5	5	4	55.80	129.75	50.72	1700474.00	0.23	0.23	0.23	6
5	8	6	6	6	2	78.73	34.41	110.84	2545067.13	0.013	0.013	0.013	8
5	9	4	4	4	0	348.69	82.07	352.03	/	0.009	0.009	0.009	10
5	10	6	6	6	0	592.63	362.79	1052.44	/	0.012	0.012	0.012	10
7	3	9	9	9	9	10.39	2.62	5.26	313190.35	0.028	0.028	0.028	0
7	4	9	9	9	1	50.40	37.74	298.16	3621348.80	0.022	0.022	0.022	9
7	5	4	4	4	0	452.33	395.58	467.50	/	0.018	0.018	0.018	10
9	3	5	5	5	0	57.58	42.95	45.19	/	0.015	0.015	0.015	10
9	5	7	8	7	0	9496.63	2863.37	10015.87	/	0.015	0.0082	0.020	10

Table 3: Random Polymatrix Game results

n	m	Approximation		
		RRHC	SA	CLO
10	10	0.0054	0.0040	0.0040
10	50	0.055	0.069	0.042
10	100	0.077	0.10	0.070
10	200	0.10	0.13	0.089
10	500	0.12	0.17	0.12
10	1000	0.15	0.20	0.15
10	2000	0.20	0.21	0.18
50	10	0.061	0.051	0.043
50	50	0.093	0.091	0.080
50	100	0.11	0.11	0.10
50	200	0.12	0.11	0.11
50	400	0.15	0.13	0.15
100	10	0.060	0.054	0.045
100	50	0.087	0.081	0.080
100	100	0.10	0.090	0.099
100	150	0.11	0.096	0.11
100	200	0.12	0.10	0.12
200	10	0.056	0.052	0.048
200	25	0.067	0.061	0.063
200	50	0.078	0.071	0.076
200	75	0.088	0.075	0.085
500	10	0.045	0.042	0.045
500	20	0.056	0.054	0.056
500	30	0.060	0.056	0.060
500	40	0.068	0.059	0.062

Co-evolutionary approach based on constraint decomposition

Emmanuel Kieffer[2], Grégoire Danoy[1]
, Pascal Bouvry[1], and Anass Nagih[3]

[1] SnT Interdisciplinary Centre, University of Luxembourg
emmmanuel.kieffer@ext.uni.lu
[2] CSC Research Unit, University of Luxembourg
{gregoire.danoy,pascal.bouvry}@uni.lu
[3] LCOMS, University of Lorraine
anass.nagih@univ-lorraine.fr

1 Introduction

Practical optimization problems are often large constrained problems in which the generation of feasible solutions still represent an important challenge. Population-based algorithms (e.g. genetic algorithm) are natured-inspired methods which experience a real success when solving free optimization problems[1]. Nevertheless when some decision variables are strongly linked through constraints, it may be very difficult to generate feasible solutions with standard evolutionary operators (e.g crossover, mutation). The initialization of the first population might also be a brainteaser and often rely on some random procedures. It is obvious that it is not possible to guaranty feasibility in these conditions. Penalty factors are thus added to the fitness function to disadvantage non-feasible solutions. Nevertheless, they are hard to define and strongly depends on the considered instance. A large penalty factor will definitely drive solutions to the feasible decision set while a small factor will not be enough to discriminate non-feasible solutions. Penalty factors do not solve the problem of generating feasible solution, they only penalize non-feasible one. If the evolutionary operators are not able to generation new valid solutions, the penalty factor will not help. In some cases, one can also observe that a feasible solution with poor fitness can be rejected in favor of a non-feasible one which are particularly closed to the feasible decision set. In this paper, we are going to describe a new approach to fix this issue. This method is based on two phases. The first one consists in ensuring a minimum rate of feasible solutions in the initial population while the second one adds a mechanism which is triggered when feasibility falls below this rate during the evolution. The remainder of this article is organized as follows. The next section will first describe some related works on constraint handling in evolutionary computing. Then, the new co-evolutionary approach will introduce as well as the decomposition scheme. Finally, we will conclude and propose some new perspectives.

2 Constraint handling in evolutionary computing

Constraint handling is a procedure of major importance in order to determine valid solutions. Feasibility is a mandatory characteristic which has to be achieve first. In this spirit, many approaches have been developed based on two main strategies. The first one is know as *Direct constraint handling* while the second one is referred as to *Indirect constraint handling*. A direct constraint handling procedure aims at driving algorithms to handle only feasible solutions. Among this strategy, a trivial approach consists in rejecting explicitly non-feasible solutions from populations [2]. However, this approach is worthwhile only if the ratio between non-feasible and feasible solutions in populations is very low. Preserving approaches [3, 4] ensure that evolutionary operators generate feasible solutions. Nevertheless, it may be hard to discover an encoding representation preserving implicitly feasibility. Another approach deals with repairing approaches whose aim is to add a local procedure bringing non-feasible solutions back into the feasible decision space. Repairing approach can dramatically increase the processing time. This is the reason why they are often with a predefined rate, i.e. a percentage of the non-feasible solutions are repaired at each generation. Then, a question remains: how these solutions should be selected? In the GENOCOP algorithm

[5], all solutions are repaired 15 % of the time. Here the question is: What kind of distribution should be uses? (e.g uniform, biased towards the last generation). Finally, the last approach is based on a encoding-decoding scheme to map individuals to feasible solution. Koziel et al. in [6] show the benefits of using such encoding-decoding approach. The only drawback is the difficulty to find the appropriate and most efficient mapping. Indirect constraint handling approaches drive algorithms to generate feasible solution discriminating non-feasible ones. The most obvious one is based on penalty factors. For instance, one way to implicitly reject non-feasible solutions is to apply a death penalty factor [7]. Death penalty factor suffers the same drawbacks as its equivalent which explicitly remove non-feasible solutions from populations. Penalty factors aims at transforming a constrained problem into an unconstrained one. Generally, constraints are removed while penalizing the fitness function when some are violated. Penalyzing functions have been first defined by Courant [8]. One can distinguishes exterior and interior penalizing functions. While the first one penalizes non-feasible solutions diverging from the feasible decision set, the second one prevent feasible solutions to become non-feasible by leaving the feasible decision set. A static penalized objective can be generally represented as follows: $f'(x) = f(x) + \mathcal{P}(d(x, F))$. $d(x, F)$ can be considered as a distance to the feasible region. For instance, $d(x, F)$ could be the number of violated constraints. The penalty function should have the following characteristics: $\mathcal{P}(0) = 0$ and if for all a and b such that $a \leq b$ one has $\mathcal{P}(a) \leq \mathcal{P}(b)$. For instance, $\mathcal{P}(x)$ could be equal to $k \cdot d(x, F)$. Nonetheless, one has to introduce new parameters and it is often hard to find the most appropriate ones. These optimal parameters may also change during the generations. This is the reason why dynamic penalty functions have been introduced to clearly take into account this aspect. Generally, the number of performed generations t is added to the penalty functions in this case [9, 10]. A multi-stage approach has been designed as well, to iteratively handle constraints. At each stage i, it tries to satisfy a new constraint i while using a death penalty for constraints $j < i$ which have been already satisfied for previous stages. This concept is refereed to as behavioral memory algorithm [11, 12]. To conclude with penalty factors, an adaptive algorithm [13] has been designed to define penalty factor based on population statistics. Due to space restrictions, this section only described the main strategies. Nevertheless, the interested authors can refer to a very detailed survey on constraints handling in [14].

3 Co-evolutionary approach based on constraints decomposition

For some problems, the combination of specific constraints can dramatically increase problem complexity. Finding feasible solutions can even be a hard task. As a result, it can be more interesting to first decompose the initial problem by separating the so called *coupling constraints*. In this manner, we generate k new sub-problems having less constrained domains, such that it is easier to generate solutions which are feasible for each sub-problem. The new approach presented hereafter aims at developing collaboration between sub-populations in order to satisfy feasibility first. This constraint decomposition mechanism breaks the links to constraints which are difficult to satisfy all at once. While a standard island model isolates sub-populations and uses migration, the approach proposed in this paper authorizes two parent solutions from different sub-populations to mate. This mechanism is less aggressive than migration and allows to bring new genetic material from one sub-population to another (see Figure 1). In addition, sub-populations are less sensible to destructive mutation driving to non-feasibility while preserving diversity.

4 Conclusion

In this paper, we first summarize the different handling constraints techniques existing in the literature as well as their main drawbacks. To cope with these issues, we proposed a new co-evolutionary approach which differs from the ones highlighted in the literature. We illustrated its workflow and how it is supposed to overcome the problems raised by the standard approaches to solve constrained optimization problems. We claimed that such approach is relevant when decision variables are strongly linked through constraint so that feasibility become a first goal to achieve. As future work, we will investigate the possibility to extend this approach to bi-level optimization problems where an optimization problem may constraint another one.

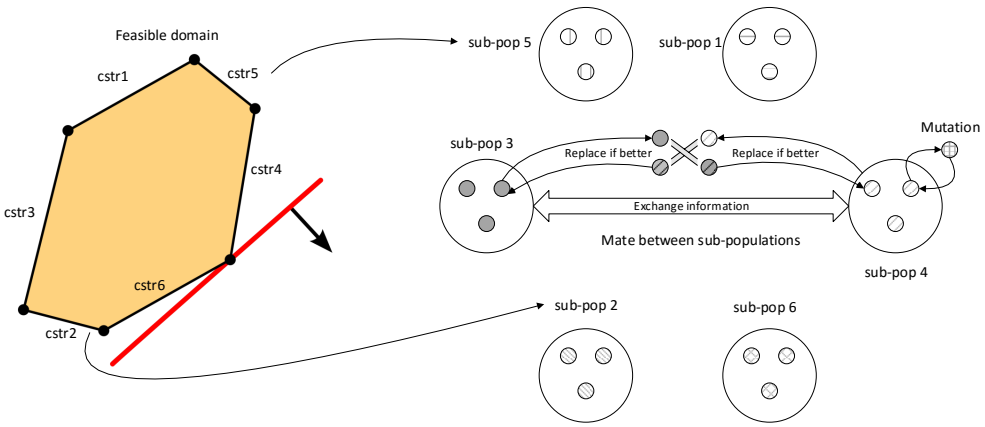


Fig. 1: Co-evolution mechanism

References

1. M. Guzek, P. Bouvry, and E.-G. Talbi. A survey of evolutionary computation for resource management of processing in cloud computing [review article]. *IEEE Computational Intelligence Magazine*, 10(2):53–67, May 2015.
2. D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
3. N/ Padhye, P. Mittal, and K. Deb. Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. *Computational Optimization and Applications*, 62(3):851–890, 2015.
4. M. Schoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, volume 1141 of *Lecture Notes in Computer Science*, pages 245–254. Springer Berlin Heidelberg, 1996.
5. Z. Michalewicz and G. Nazhiyath. Genocop iii: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *Evolutionary Computation, 1995., IEEE International Conference on*, volume 2, pages 647–651. IEEE.
6. S. Koziel and Z. Michalewicz. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary computation*, 7(1):19–44, 1999.
7. H. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993.
8. R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, 49(1):1–23, 01 1943.
9. J. Joines and C. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga's. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 579–584 vol.2, Jun 1994.
10. S. Kazarlis and V. Petridis. Varying fitness functions in genetic algorithms: Studying the rate of increase of the dynamic penalty terms. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, PPSN V, pages 211–220, London, UK, UK, 1998. Springer-Verlag.
11. Marc Schoenauer and Spyros Xanthakis. Constrained ga optimization. In *Proc. of the 5th Int'l Conf. on Genetic Algorithms*, pages 573–580. Morgan Kaufmann, 1993.
12. J. Sullivan and A. Pipe. An evolutionary optimisation approach to motor learning with first results of an application to robot manipulators. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 5, pages 4406–4411. IEEE, 1997.
13. D. Tate and A. Smith. Unequal-area facility layout by genetic search. *IIE transactions*, 27(4):465–472, 1995.
14. C. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245 – 1287, 2002.

Particle Swarm Optimization Algorithm for Improve The Final State Error of The Exact Boundary Controllability

A.Khernane[1] N.Khelil[2] S.Khernane[3]
[1]University of Batna and LMA Laboratory
UMK Biskra,Algeria
khernane_a@yahoo.fr
[2]LMA Laboratory, UMKBiskra, Algeria
n.khelil@univ-biskra.dz
[3] Student in Master at Department of Computer Science
University of Batna ,Algeria
khernane.sara@gmail.com

1 Abstract

The implementation of the Hilbert uniqueness method allows to approximate the control that returns the considered system to rest at time T with a final state error for exact boundary controllability of Euler-Bernoulli beam equation when the control is the Dirichlet type.
The aim of this study is to compare the calculation of this error when the selected points are equidistant with the error which points are chosen by Particle Swarm Optimization(PSO)
The results show the improvement of the final error in the second case compared to the first case.

2 Formulation of The Problem

Let T denote a given positive number and let $y^0(x)$ and $y^1(x)$ denote given functions defined on Ω=]0,1[. Let Σ={0, 1}x]0,T[, Q=]0,1[x]0,T[and $(y^0, y^1) \in L^2(\Omega)\text{x}H^{-2}(\Omega)$.

The Exact Dirichlet boundary controllability problem for the Euler-Bernoulli beam equation is: Find a control function v defined on Σ such that y satisfies:

$$\begin{cases} y_{tt} + y_{xxxx} = 0 & \text{in Q} \\ y(x,0) = y^0(x) \ , \frac{\partial y}{\partial t}(x,0) = y^1(x) & \text{in } \Omega \\ y(x,T) = 0 \ , \frac{\partial y}{\partial t}(x,T) = 0 & \text{in } \Omega \\ y(0,t) = 0 \ , y(1,t) = 0 & \text{t}\in[0,\text{T}] \\ \frac{\partial y}{\partial x}(0,t) = 0 \ \frac{\partial y}{\partial x}(1,t) = v(t) & \text{t}\in[0,\text{T}] \end{cases} \tag{1}$$

It is well known that state y and control function v such that (1) is satisfied exist provided T positive [4],[5].
The proposed method in [1] has explicitly determine the control $\upsilon*$ such that y satisfies(1)with a final state error

$$\|\xi\|^2 = \|y(x_i,T)\|^2_{L^2(\Omega)} + \|\frac{\partial y(x_i,T)}{\partial t}\|^2_{L^2(\Omega)} \tag{2}$$

The selected points are equidistant in the calculation of (2).
In this study, we prove that (2) can be optimized by particle swarm optimization(PSO).
The problem that we consider is to minimize (2)by taking the same example traited in [1].
For this, we try determine x_i by PSO so that the final state error is close to zero.

3 Overall Description Strategy of Particle Swarm Optimization

The particle swarm treatment[2],[3]supposes a population of individuals designed as real valued vectors particles, and some iterative sequences of their domain of adaptation must be established. It is assumed that these individuals have a social behavior, which implies that the ability of social conditions, for instance, the interaction with the neighborhood, is an important process in success-fully nding good solutions to agiven problem.
The strategy of the PSO algorithm is summarized as follows: We assume that each agent (particle) i can be represented in a N dimension space by its current position $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ and its corresponding velocity
$v_i = (v_{i1}, v_{i2}, ..., v_{iN})$. Also a memory of its personal (previous) best position is represented by $p_i = (p_{i1}, p_{i2}, ..., p_{iN})$, called (pbest), the subscript i range from 1 to s, where s indicates the size of the swarm. Commonly, each particle localizes its best value so far(pbest) and its position and consequently identies its best value in the group (swarm), called also(sbest) among the set of values (pbest). The velocity and position are updated as:

$$\upsilon_{ij}^{k+1} = \omega_{ij}\upsilon_{ij}^k + c_1 r_1^k \left[(pbest)_{ij}^k - x_{ij}^k\right] + c_2 r_2^k \left[(sbest)_{ij}^k - x_{ij}^k\right] \tag{3}$$

$$x_{ij}^{k+1} = \upsilon_{ij}^{k+1} + x_{ij}^k \tag{4}$$

where x_i^{k+1},v_i^{k+1} are the position and the velocity vector of particle i respectively at iteration k+1, c_1 and c_2 are acceleration coecients for each term exclusively situated in the range of 2- -4,w_j is the inertia weight with its value that ranges from 0.9 to 1.2, where as r_1, r_2 are uniform random numbers between zero and one. For more details, the double subscript in the relations (3) and (4)means that the rst subscript is for the particle i and the second one is for the dimension j.
The results are:

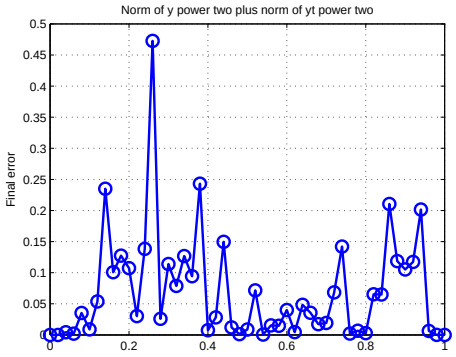


Figure 1: Equidistant points.

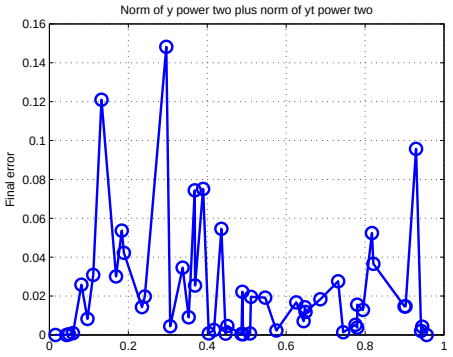


Figure 2: Points found by PSO.

The experimental results show that the final error is improvement in this study.

References

[1] A.Khernane.An approximation method for exact boundary controllability of Euler-Bernoulli System.Int.J.of mathematical,computational,Phy.Elec.and comuter Eng.Vol:9,N12,2015,pp 717-722.

[2] J.Kennedy and R.C.Eberhart, Swarm Intelligence.Morgan Kaufmann Publishers, San Fran- cisco, (2001).

[3] L.Djerou et al. Numerical Integration Method Based on Particle Swarm Optimization. Y. Tan et al. (Eds.) : Part I, LNCS 6728, Springer-Verlag Berlin Heidelberg, pp. 221-226, (2011)

[4] J.L. Lions. Contrlabilit exacte des systmes distribus, volume 1, Masson, Paris, 1988.

[5] E.Zuazua. Contrlabilit exacte dun modle de plaques vibrantes en un temps arbitrairement petit, C.R.Acad.Sci.Paris, Serie I, n7, 173-176, 1987.

New Generic Approach for Multi-objective Problems with Fuzzy Data

Oumayma Bahri[1,2], Nahla Ben Amor[1] and El-Ghazali Talbi[2]

[1] Laboratoire LARODEC, Université de Tunis, Le Bardo 2000, Tunisie
oumayma.b@gmail.fr nahla.benamor@gmx.fr
[2] Laboratoire CRISTAL, Université de Lille 1, Villeneuve d'Ascq 59650, France
el-ghazali.talbi@lifl.fr

1 Introduction

In practice, the problems encountered often involve multiple objectives to be optimized simultaneously and also include some inevitable uncertainty that may result from unreliable information sources. Such problems called multi-objective optimization problem (MOP) under uncertainty, become nowadays one of the major challenges in decision making area. In fact, the combination of both important aspects, namely multi-objectivity and uncertainty, leads often to dramatically increasing the complexity of these problems. Otherwise, an uncertain MOP is characterized by the necessity of generating a set of efficient solutions while considering uncertainties and their effects in the results.

Despite the wide applicability of this kind of problems, very little research works have been done to handle it as-is, without erasing any of their multi-objective or uncertain characteristics [6]. Unfortunately, almost all existing approaches have been often limited to transform the uncertain MOP into one or more mono-objective problems by using for example aggregation functions [7]. Some other approaches have been focused on treating the problem in its multi-objective context while ignoring the propagation of uncertain inputs to the objectives and obviously to the resulting solutions [8].

All these remarks lead us to propose a new generic approach for handling any uncertain multi-objective problem while considering uncertainty propagation through the optimization process. Specifically, we focused on the most critical and sensitive case where uncertainty is assumed to affect the objective functions. In addition, we suggested to use fuzzy sets in order to express the uncertain data in a suitable and natural way [5]. The next section outlines our main contributions to cope with the reported problem.

2 Contributions

As mentioned above, our aim is to handle MOPs with fuzzy data, in which fuzziness is expressed by triangular fuzzy numbers and thereby propagated to the set of objective functions. First, we have proposed a novel approach composed of three main stages:

- Definition of a new Pareto approach for ranking the generated fuzzy-valued objective functions since the standard Pareto dominance cannot be used in our fuzzy context [2].
- Fuzzy extension of two Pareto-based evolutionary algorithms in order to enable them working in a fuzzy space. The extended algorithms denoted E-SPEA2 and E-NSGAII integrate the proposed Pareto dominance in their fitness assignment strategy and use refined techniques of diversity preservation to the fuzzy context [3].
- Definition of new concepts of robustness to analyze fuzziness propagation in the optimization process [3].

The extended algorithms were implemented with the multi-objective module of ParadisEO-2.0 under Linux [11] and subsequently applied to solve a multi-objective variant of vehicle routing problem (VRP) with uncertain demands [9]. To validate them, some experimental tests were finally carried out across a set of fuzzy benchmark instances generated at random from the well-known crisp Solomon's benchmark [10]. In these tests, the fuzzy outcomes are approximated into exact

forms by calculating their expected values and then qualified using two classic multi-objective quality indicators namely, Hypervolume and Epsilon metrics.

At this level, we have obtained encouraging results, especially about the variation of extended algorithms over all the sampled fuzzy instances and their ability to deal with fuzziness. Nevertheless, approximating the generated solutions to exact values may be criticised, since it reduces the information provided by the fuzzy solutions and so affects their robustness. To this end, we have integrated the new robustness concepts into the search process of the two previously extended algorithms E-SPEA2 and E-NSGAII in order to enable them achieving robust optimal solutions. More precisely, we have incorporated in the fuzzy dominance relations, the robustness as an additional criterion for ranking solutions. Besides, we have followed the same previous experimental tests based on the VRP application and fuzzy sampled instances. Thereafter, we have used Monte-Carlo method for simulating exact "real" results. The simulations are generated randomly according to different distributions which are coherent with fuzzy values.

Finally, by observing the generated solutions with their respectively robustness degrees, we have deduced that they reached the desired robustness level compared to the simulated ones.

3 Conclusion

This paper proposes a novel approach for handling multi-objective problems with fuzzy data modelled via triangular fuzzy numbers. The proposed approach can be efficiently used for achieving optimal robust solutions for any MOP with fuzzy valued objectives.

As future work, we intend to extend performance indicators (i.e., Hypervolume indicator) to the robust fuzzy context. It would also be interesting to validate the proposed approach for different fuzzy multi-objective problems, in which fuzziness is expressed by other shapes like trapezoidal fuzzy numbers.

References

1. Bahri, O., Ben Amor, N. and Talbi, E.-G.: New Pareto approach for ranking triangular fuzzy numbers. IPMU'14, 443:264-273 (2014)
2. Bahri, O., Ben Amor, N. and Talbi, E.-G.: Optimization algorithms for multi-objective problems with fuzzy data. MCDM'14, 194-201 (2014)
3. Bahri, O., Ben Amor, N. and Talbi, E.-G.: B-Robustness Approach for Fuzzy Multi-objective Problems. IPMU'16, 1-13 (2016)
4. Talbi, E-G.: Metaheuristics: from design to implementation. Wiley (2009)
5. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems, 3-28 (1987)
6. Limbourg, P. and Daniel, E.S.: An Optimization Algorithm for Imprecise Multi-Objective Problem Functions. Evolutionary Computation, 1:459-466 (2005)
7. Paquete, L. and Stutzle, T.: Stochastic local search algorithms for multi-objective combinatorial optimization. Handbook of Approximation Algorithms and Metaheuristics (2007)
8. Teich, J.: Pareto-front exploration with uncertain objectives. Evolutionary Multi-Criterion Optimization (EMO'01), Lecture Notes in Computer Science (LNCS), 1993: 314-328 (2001)
9. Toth, P. and Vigo, D.: The Vehicle Routing Problem. Siam (2002)
10. Solomon, M.M.: Algorithms for the vehicle Routing and Scheduling Problem with Time Window Constraints. Operations Research, 35.2, 254-265 (1987)
11. Zitzler, E., Thiele, L., and al.: Performance assessment of multi-objective optimizers: An analysis and review. Evolutionary Computation, 7.2, 117-132 (2003)

Through the maximum a posteriori based metaheuristic

A. Ghoumari[1], A. Nakib[1], and P. Siarry[1]

Université Paris Est, Laboratoire LISSI EA 3956, 122 Rue P. Armangot,
94400 Vitry sur seine, France
nakib@u-pec.fr

1 Introduction

In large scale optimization problems,increasing the number of evaluations to explore the search space so as to reach the best solution does not suffice to find optimal solution. More specifically the problematic for evolutionary algorithms (EAs) is to find and respect the balance between intensification and exploration. For the exploration phase the measure of the diversity of population can be used, and for the intensification phase the role of operators is fundamental. Thus, we work on an EA which could both alternate improve the effect of those phases. EA based on a Bayesian approach uses a set of strategies aimed at improving the speed of convergence to the global optimum solution. During the exploration the BEA evaluates the diversity using a quality metric, here we used the Euclidian distance. It corresponds to the decision moments on figure 1. Then for the intensification, a strategy, chosen during the exploration, is applied during δ generations (δ a number of predefined generations). Thus the BEA is an algorithm relying on a prediction mechanism. The choice of the strategy is based on the diversity of the past populations. Indeed the chosen strategy for the future generations is the one which maximizes the diversity of these past populations and so improves the performance for the exploration phase.

A strategy is a couple of one crossover with one mutation. We have three crossovers and five mutations, so fifteen strategies possible. We use the BLX-α, the discrete and the linear crossovers. Moreover we have the Levy, Gaussian, Scramble mutations and the DE/RAND/1/BIN and the DE/RAND/2/BIN.

Algorithm 1: Bayes based Evolutionary Algorithm **BEA**

Generate Randomly a population
Initialize the probability of each strategy with a given metric
while stopping criteria are not satisfied **do**
  a. Choose the best strategy (couple of crossover and mutation) with maximum a posteriori rule after δ generations
  b. Select the chromosomes
  c. Apply crossover on selected parents
  d. Apply mutation on the selected individual
end while
return The best solution

For each generated population from the elected strategy, others strategies (i and j here) are applied without evaluation of the fitness of offsprings. Then, we compute the diversification of these populations, and the probability of each population among the δ present. Afterwards, based on the assumption that the probability of choosing each strategy is independent, then, the maximum a posteriori principle is used the predict the following strategy.

2 Results and discussions

In first, we illustrate the fact of using several strategies compared to one using one strategy to find the optimal solution using an evolutionnary algorithm. In fig. 1, the percentage of finding the optimal solution was studied. One can remark that when using several strategies the success rate

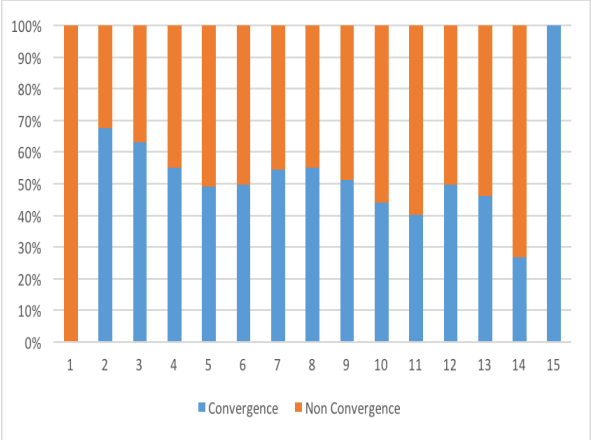


Fig. 1: Illustration of the advantage of using set of strategies.

increases. In order to compare the performance of BEA, a test on six shifted and high dimensional problems which were considered in CEC2008 special session and a competition on large scale global optimization[1] was done. In this paper, the dimension of the problems is set to D=100, D=500 and D=1000. All the test functions used in the following experiments are to be minimized. More details about the definitions of problems can be found in[1]. The experiments were done to compare 12 algorithms including the proposed approach BEA on defined problems. To have a fair comparison, we use the same comparison strategy by the suggestions of the organizers of CEC 2008 competition. The maximum number of evaluations MAX_{FEs} is set to $5000 \times D$ for all algorithms. For each test function, all algorithms are conducted over 25 runs. The average function error value of $F(x) - F(x^o)$ where $F(x^o)$ is the global optimum of $F(x)$ is recorded [1].

Table 1: Average Error of the 25 independent runs tested on CEC'2008 benchmark (D=100).

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6
DECC	2.72e-29	5.44e+1	1.42e+2	5.33e+1	2.75e-3	2.36e-1
DECC-ML	5.72e-28	2.79e-4	1.88e+2	0.00e+0	3.64e-3	3.38e-14
DECC-D	2.92e-29	5.24e+1	1.40e+2	5.44e+1	8.87e-4	1.22e-1
DECC-DML	4.73e-28	2.48e-4	1.92e+2	0.00e+0	7.88e-4	3.15e-14
MLCC	6.82e-14	2.52e+1	1.49e+2	4.38e-13	3.41e-14	1.11e-13
CMAES	3.19e-24	1.32e+1	4.33e+0	2.63e+2	8.88e-18	6.64e-1
EPUS-PSO	7.47e-1	1.86e+1	4.99e+03	4.71e+2	3.72e-1	2.06e+00
DEwSAcc	5.68E-14	8.25e+00	1.44e+02	4.38e+00	3.07e-14	1.13e-13
DMS-PSO	0.00e+0	3.65e+0	2.83e+02	1.83e+0	**0.00e+0**	**0.00e+0**
LSEDA-gl	2.27e-13	**2.21e-13**	2.81e+2	1.31e+2	2.84e-14	9.78e-14
JDEdynNP-F	9.32e-14	4.29e-1	1.12e+2	5.46e-14	2.84e-14	5.68e-14
BEA	**0.00e+0**	1.2288e+0	5,0982e+1	**0e+0**	**0e+0**	1.39e-14

References

1. K. Tang, X. Yao, P. N. Suganthan, C. Macnish, Y. Chen, C. Chen, Z. Yang. *Benchmark functions for the CEC'2008 special session and competition on high-dimensional real-parameter optimization.* Technical Report. Nature Inspired Computation and Applications Laboratory, USTC, China, 2007. http://nical.ustc.edu.cn/cec08ss.php.

Toward smart manufacturing using Genetic Algorithm for optimal scheduling

S. Kouki[1] and M. Jemni[2]

LaTICE Laboratory, National Graduate School of Engineer of Tunis
University of Tunis, Tunisia
1.samia.kouki@esstt.rnu.tn
2. mohamed.jemni@fst.rnu.tn

Keywords : smart manufacturing, scheduling problems, Genetic Algorithm (GA).

1 Introduction

In this paper, we present an ongoing research to investigate a new work about efficient scheduling in smart manufacturing. In fact, in the last few years [2, 3, 4] we developed several solutions for classical manufacturing scheduling problems. We targeted in particular the Permutation Flow Shop Problem (PFSP) to find optimal scheduling of n jobs in m machines with a set of constraints. We used mainly high performance computing environment such as grid computing and parallel algorithmics strategies. We hopefully, developed efficient techniques for the PFSP and we succeeded to resolve new data instances from Taillard's [5] benchmarks.

In this study, our objective is to move forward new topics related to smart manufacturing problems and their new challenges i.e. the huge number of customers. These new challenges address both big data problems and scheduling manufacturing constraints. In this context, genetic algorithms present an interesting alternative to provide high quality solutions for the PFSP in industrial manufacturing production.

2 Description of our Genetic Algorithm

Genetic Algorithm (GA) is an iterative search meta-heuristic used to generate useful and high quality solutions to hard optimization problems that cannot be solved effectively by exact methods [1]. It has been used to solve many different types of NP-hard problems. GA generates solutions to handle problems using techniques inspired from natural evolution and behavior of chromosomes within a population of individuals. It is founded on natural theories of evolution.

The efficiency of GAs depends on many parameters, such as the size and the composition of initial population, the selection strategy and mainly the combination of different operators like crossover and mutation [6]. Thus, finding good solutions to NP-hard problems requires a proper setting of genetic algorithm operators. In this work, our aim is to develop an efficient

sequential Genetic Algorithm for the PFSP. We propose to solve the PFSP using GA based on experimenting various combinations of its different parameters.

3 Main results

Crossover and mutation are the two most commonly used genetic search operators. Crossover produces offspring by recombining the information from two parents. Mutation is a process used in GAs to maintain genetic diversity by changing some randomly selected chromosomes.

Many selection strategies were presented in the literature, having different ways to compute the selection probability. The four major types of selection schemes that we experimented in this work are as follows: Roulette wheel Selection, Tournament Selection, Elitism and Uniform selection.

For the experimental evaluation we used a *Relative Performance*, denoted *RP*, that is calculated for all data instances using the following formula: $RP =$ ———

Where C_{max} is makespan criterion to optimize and *LB* is the lower bound of the instance.

Another Metric is used to compare our results to the literature, which is the *Average Relative Performance*, and denoted $PRmoy$. This parameter presents the average of the relative performances of a set of 10 instances. For example, for all the 10, 50x20 instances, the *RPmoy* is given by the following formula: ———

Where i is the rank of the instance in the dataset.

The first criterion addressed in this work is the selection method of individual from the initial population.

	Dataset								
Selection Method	**20X5**	**20X10**	**20X20**	**50X5**	**50X10**	**50X20**	**100X5**	**100X10**	**100X20**
Uniform	**0,4**	1,3	**0,7**	**0,5**	**1,5**	**2,6**	**0,2**	**1**	**2,2**
Roulette wheel	0,5	**1**	**0,7**	**0,5**	2,3	2,9	0,2	1,2	2,6
Tournament	0,6	1,6	2,3	0,6	2	3,4	0,4	3,3	2,9
Elitism	0,6	**1**	1,3	0,7	2,5	3,2	0,2	1,1	3

TABLE I: BEST TESTED SELECTION METHODS

Table I shows different values for each set of Taillard instances, while varying selection schemes. Best values for each set of instances are mentioned in bold.

We can deduce from TABLE I, that the Roulette wheel selection method provides good results only for few instances. The Elitism method is not very interesting but it is more efficient than a tournament one. These results confirm that the uniform selection scheme is the best method among all other ones. Therefore, in the rest of our experimental study, we will use the uniform selection method to adjust some other parameters for the GA.

For the generating method, we tested two variant of AGs using different methods to generate the initial population. Ran and $RanNEH$ techniques are used in order to choose the best way for population generation. Ran is random generating method however, $RanNEH$ is a combination between heuristics and random solutions. Thus, for a population composed of m individuals, $RanNEH$ consists on generating a single chromosome by NEH to provide good quality of the initial population.

Based on our experimental study, the mixed population of the initial generation is generally the best method to solve the PFSP. For selection methods, the experiments have proved the efficiency of the uniform method of selection compared to other tested methods such as Roulette wheel, Tournament and Elitism.

4 Conclusion

In this work, we investigated GA's paradigm in order to select appropriate operators and parameters of the GA to solve the PFSP. We obtained promising results and we already started working on other GA operators and parameters such as termination criteria and size of population.

References

[1] El-Ghazali Talbi, Metaheuristics - From Design to Implementation. Wiley 2009, ISBN 978-0-470-27858-1, pp. I-XXIX, 1-593

[2] S. Kouki, M. Jemni, and T. Ladhari, "Scalable Distributed Branch and Bound for the permutation Flow Shop Problem," in Proc. of 3PGCIC, pp. 503-508, 2013, Compiegne, France.

[3] S. Kouki, M. Jemni, and T. Ladhari, "A Load Balanced Distributed Algorithm to Solve the Permutation Flow Shop Problem Using the Grid", 15th IEEE International Conference on Computational Science and Engineering, December 5-7, 2012, Paphos, Cyprus.

[4] S. Kouki, M. Jemni, and T. Ladhari, "Deployment of Solving Permutation Flow Shop Scheduling Problem on the Grid," Conference on Grid and Distributed Computing (GDC) 2010, LNCS, Springer, December 13-15, 2010, Jeju Island, Korea.

[5] E. Taillard : Scheduling instances (2008), http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html,

[6] K. Deb, S. Agrawal, "Understanding Interactions among Genetic Algorithm Parameters," FOGA, Amsterdam, The Netherlands, pp. 265-286, 1998.

Multi-objective robust scheduling to maintain French nuclear power plants

Nicolas Dupin[1], El-Ghazali Talbi[1]

Univ. Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille
el-ghazali.talbi@univ-lille.fr, nicolas.dupin.2006@polytechnique.org

Abstract: Scheduling the maintenances of nuclear power plants is a complex industrial problem, formulated in 2-stage stochastic programming for the challenge EURO/ROADEF 2010 with uncertainty on demands, production costs and capabilities. The first stage optimization concerns the maintenance dates and refuelling decisions, whereas the second level concerns unit commitment problems to fulfil the power demands, ensuring feasibility and cost of the first stage decisions. Our extensions applies for the monthly reoptimisation of the maintenance planning with requirements of stability and robustness. A MIP formulation is first investigated to model the deterministic problem with stability requirements. The extension of the ROADEF Challenge considering uncertainty on outage durations is then investigated. An approach similar to 2 stage robust optimization gives inconsistent results, with often infeasibility cases to search robust feasible solutions. A robust approach inspired from multi-objective optimization allows to have robustified solutions for all the instances of the Challenge dataset. Robustness and stability issues emphasize the need of multi-objective optimisation to schedule the maintenance of nuclear power plants, furnishing Pareto front to support decision makers to arbitrate good compromise solutions.
Keywords: Robust Optimization, Mixed Integer Programming, Variable Neighbourhood Search, Pareto front, EURO/ROADEF 2010 Challenge, Maintenance scheduling.

1 Introduction

This paper addresses the large-scale power plant maintenance scheduling problem, which has been proposed for the ROADEF/EURO Challenge 2010 organized conjointly by ROADEF and EURO, respectively the French and the European Operational Research and decision support Societies. The problem was specified by the French utility company Électricité de France (EDF), to address the large-scale scheduling problem of nuclear power plant outages for maintenance and refuelling.

ROADEF/EURO Challenge 2010 The ROADEF/EURO Challenge 2010 was specified by the French utility company (EDF), to address the large-scale scheduling problem of nuclear power plant outages for maintenance and refuelling. This gave rise to a 2-stage stochastic formulation for the challenge, using discrete scenarios to model uncertainty. Uncertain data concern power demands, production capacities and costs. The first stage high-level problem concerns the outages weeks of nuclear power plants and the refuelling quantities, with scheduling constraints for outages. The second stage low-level problem computes for all stochastic scenario the production plan implied by the first stage decisions, to minimize the mean cost over all the scenarios.

State of the art ROADEF/EURO Challenge 2010 One major difficulty of the challenge is to handle the size of the problem. The best results were obtained in [1] with frontal meta-heuristics, for an aggressive local search. Exact methods reduced the problem to tackle the instances sizes. A first common simplification was to aggregate the production time steps to weeks. [2] is the only exact approach which did not aggregate the stochastic scenarios, solving a MIP relaxing only two types of constraints by Bender's decomposition. However, this approach was not efficient to tackle the real size instances. [3] fixes outages decisions on the average scenario, dualizing demand coupling constraints for a column generation procedure independent column generation sub-problems for all units, giving the best solutions among exact methods. Several approaches used MIP or LP inside a heuristic algorithm. Especially, [4] solves MIP models for the first level scheduling problem, without any production variables, and production or refuelling problems as LP when first level decisions are fixed. Heuristic methods iterating following the 2-stage structure were less efficient than [1], like approaches [5–9, 4]. We refer to [10] for a more general survey.

Robust optimization for scheduling problems Facing the difficulty of having uncertain input data of optimisation problems, robust optimisation as introduced in [11] aims to provide resistant solutions of uncertainty given in a defined *uncertainty set* without knowing probabilistic distributions. Minimizing the worst case cost for all occurrence in the uncertainty set, it leads to min-max problems, with case by case resolution. 2-stage robust optimisation was introduced in [12, 13] for sequential decision-making under uncertainty, where the decision-maker must make some strategical or tactical "here and now" decisions, before discovering the actual value of some uncertain data, having the opportunity to take further action, called recourse,in a more operational level with "wait and see" decisions, once uncertainty has outcome, while minimizing the worst case cost in the uncertainty set. We note that it exists a wide variety of robustified approaches with different definitions of robustness in the context of dynamic scheduling, we refer to [14] for a survey.

Paper outline This paper will provide dual bounds for the ROADEF Challenge, using MIP to compute bounds on restricted problems, proving that the different restrictions imply dual bounds for the whole problem. Section 2 gives an overview of the problem constraints, while section 3 presents a MIP model for the problem, relaxing only two set of constraints. To deal with smaller problems, section 4 proves that the aggregation of production time steps gives lower bounds. Section 5 proves that we can compute dual bounds with restrictions to single scenarios. Section 6 provides a parametric family of dual bounds, relaxing outages and aggregating constraints. Computational results are reported in section 7 and conclusions from this research are drawn in section 8.

Constraint	Description
CT1	for all s, t, the total production is the demand $\mathbf{Dem}_t^s$
CT2	Production of T1 plant $j \in \mathcal{J}$ is in $[\mathbf{Pmin}_{jt}^s, \mathbf{Pmax}_{jt}^s]$ for $s \in \mathcal{S}$ and $t \in \mathcal{T}$
CT3	Production of T2 plants are null during an outage
CT4-5	Production of T2 plant i is in $[0, \mathbf{Pmax}_{it}]$ for $t \in \mathcal{T}$,
CT6	Decreasing power profile at the end of nuclear cycles (relaxed)
CT12	Modulation constraints (relaxed)
CT7	bounds the refuelling of outage k of T2 plant i in $[\mathbf{Rmin}_{i,k}, \mathbf{Rmax}_{i,k}]$
CT8	Initial fuel stocks for T2 units have value $\mathbf{Xi_i}$
CT9	Dynamic evolution fuel stocks/production for T2 units.
CT10	Fuel losses at refuelling
CT11	The fuel level is in $[0, \mathbf{S}_{i,k}]$ for cycle k of T2 unit i.
	The fuel level must be lower than $\mathbf{A}_{i,k+1}$ to process outage $k+1$
CT13	Time windows constraints for the beginning dates of outages
CT14-18	Minimal spacing/ maximal overlapping constraints among outages
CT19	maximal overlapping constraints among outages with resource constraints
CT20	Maximal number of simultaneous outages
CT21	Maximal power off-line

Table 1. Definition of the constraints of the challenge ROADEF

2 Problem statement

We summarize here the problem description, we refer to [15] for the challenge specifications.

Set and index Two kinds of power plants are modelled. On one hand, Type-2 (shortly T2) power plants indexed with $i \in \mathcal{I}$, correspond to nuclear power plants. T2 power plants have to be shut down for refuelling and maintenance regularly. On the other hand, Type-1 (shortly T1) power plants are indexed with $j \in \mathcal{J}$, model other power plants with more flexibility in the production. *O*utages and *p*roduction campaigns are indexed with the cycles $k \in \mathcal{K}$ for all T2 plant. By convention, a cycle begins with the outage period for maintenance and refuelling, before the *p*roduction campaign.

The time horizon is discretized with two kind of homogeneous time steps. The outage decisions are discretized weekly and indexed with $w \in \mathcal{W} = [\![1; W]\!]$, whereas $t \in \mathcal{T}$ denotes production times steps from 8h to 24h. $\mathbf{D}_t$ denotes the duration of the production time steps. t_w denotes the first production period of the week $w \in W$, each production time step t is associated to its week w_t.

Objective function The objective function minimizes the expected cost of production while satisfying customer load for all time steps and all production scenarios. Production costs of T1 units j are $\mathbf{C}^s_{jt}$ proportional to the production levels for all scenarios s at time step t. Production cost of T2 units are calculated proportionally to the fuel consumption: proportional refuelling costs $\mathbf{C}^r_{i,k}$ are considered, and reduced with the proportional cost of the remaining fuel with a proportional factor $\mathbf{C}^f_{i,s}$ to avoid end-of-side effects.

Constraints description Table 1 defines the constraints with their nomenclature in the challenge specification [15]. CT1 to CT6 and CT12 are production constraints, whereas CT7 to CT11 are stock level constraints. The remaining constraints are specific to T2 plants outage scheduling.

3 MIP formulation with stability constraints ond objective

In this section, we provide a MIP formulation for the problem, relaxing only constraints CT6 and CT12 similarly as [2]. It leads to a MIP formulation where the only binary variables are the outages weeks decisions. A major modeling difference with [2] is in the binary variable definitions of $d_{i,k,w}$: we define $d_{i,k,w} = 1$ if and only if the outage beginning week for unit i's cycle k is before week w. Binary variables $x_{i,k,w}$ in [2] are equal to 1 if and only if outage beginning week for cycle (i,k) is exactly w. Such choice allows to have efficient branching following results of [16, 17]. Other continuous variables to have a linear formulation are refuelling quantities $r_{i,k}$ for each outage (i,k), T2 power productions $p_{i,k,t,s}$ at cycle k, fuel stocks at the beginning of campaign (i,k) (resp at the end) $x^{init}_{i,k,s}, x^{fin}_{i,k,s}$, T1 power productions $p_{j,t,s}$, and fuel stock $x^f_{i,s}$ at the end of the optimizing horizon. It gives rise to the MIP formulation:

$$v_0 = \min \sum_{i,k} \mathbf{C}^r_{i,k} r_{i,k} + \sum_{i,k,w} \mathbf{C}^{pen}_{i,k,w}(d_{i,k,w} - d_{i,k,w-1}) + \sum_{j,s,t} \pi_s \mathbf{C}^p_{j,s,t} \mathbf{D}^t \, p_{j,s,t} - \sum_{i,s} \pi_s \mathbf{C}^f_{i,s} x^f_{i,s} \quad (1)$$

$$\forall i,k,w, \quad d_{i,k,w-1} \leqslant d_{i,k,w} \quad (2)$$

$$\forall i,k, \quad d_{i,k,\mathbf{To}_{i,k}-1} \leqslant 0 \quad (3)$$

$$\forall i,k, \quad d_{i,k,\mathbf{Ta}_{i,k}} \geqslant 1 \quad (4)$$

$$\forall s,t, \quad \textstyle\sum_{i,k} p_{i,k,s,t} + \sum_j p_{j,s,t} = \mathbf{Dem}^{t,s} \quad (5)$$

$$\forall j,s,t, \quad \mathbf{Pmin}^s_{j,t} \leqslant p_{j,s,t} \leqslant \mathbf{Pmax}^s_{j,t} \quad (6)$$

$$\forall i,k,s,t, \quad p_{i,k,s,t} \leqslant \mathbf{Pmax}_{i,t}(d_{i,k,w_t-\mathbf{Da}_{i,k}} - d_{i,k+1,w_t}) \quad (7)$$

$$\forall i,k, \quad \mathbf{Rmin}_{i,k} \; d_{i,k,W} \leqslant r_{i,k} \leqslant \mathbf{Rmax}_{i,k} \; d_{i,k,W} \quad (8)$$

$$\forall i,s, \quad x^{init}_{i,0,s} = \mathbf{Xi}_i \quad (9)$$

$$\forall i,k,s, \quad x^{fin}_{i,k,s} = x^{init}_{i,k,s} - \textstyle\sum_t \mathbf{D}^t \, p_{i,k,s,t} \quad (10)$$

$$\forall i,k,s, \quad x^{init}_{i,k,s} - \mathbf{Bo}_{i,k} = r_{i,k} + \tfrac{\mathbf{Q}_{i,k-1}}{\mathbf{Q}_{i,k}}(x^{fin}_{i,k-1} - \mathbf{Bo}_{i,k-1}) \quad (11)$$

$$\forall i,k,s, \quad x^{init}_{i,k,s} \leqslant \mathbf{S}_{i,k} \quad (12)$$

$$\forall i,k,s, \quad x^{fin}_{i,k,s} \leqslant \mathbf{A}_{i,k+1} + (\mathbf{S}_{i,k} - \mathbf{A}_{i,k+1})(1 - d_{i,k+1,W}) \quad (13)$$

$$\forall i,k,s, \quad x^f_{i,s} \leqslant x^{fin}_{i,k,s} + \overline{S}_i(d_{i,k,W} - d_{i,k+1,W}) \quad (14)$$

$$\forall c,w, \quad \textstyle\sum_{(i,k)\in\mathbf{A}^c}(\alpha_{i,k,w} d_{i,k,w}) \leqslant \beta^c_w \quad (15)$$

$$d \in \{0,1\}^N, r,p,x \geqslant 0 \quad (16)$$

(2) is required with definition of variables d. (3) and (4) model CT13 time windows constraints: outage (i,k) is operated between weeks $\mathbf{To}_{i,k}$ and $\mathbf{Ta}_{i,k}$. (5) models CT1 demand constraints. (6) models CT2 bounds on T1 production. (7) models CT3, CT4 and CT5 bounds on T2 production. (8) models CT7 refuelling bounds, with a null refuelling when outage i,k is not operated, ie $d_{i,k,W} = 0$. (9) writes CT8 initial fuel stock. (10) writes CT9 fuel consumption constraints on stock variables of cycles k $x^{init}_{i,k,s}, x^{fin}_{i,k,s}$. (11) models CT10 fuel losses at refuelling. (12) writes CT11 bounds on fuel stock levels only on variables $x^{init}_{i,k,s}$ which are the maximal stocks level over cycles k. thanks to (10). (13) models CT11 min fuel stock before refuelling, these constraints are active

for a cycle k only if the cycle is finished at the end of the optimizing horizon, ie if $d_{i,k+1,W} = 1$, which enforces to have disjonctive constraints where case $d_{i,k+1,W} = 0$ implies a trivial constraints thanks to (12). (14) is a linearizing constraints to enforce $x^f_{i,s}$ to be the fuel stock at the end of the time horizon. $x^f_{i,s}$ is indeed the $x^{fin}_{i,k,s}$ such that $d_{i,k,W} = 1$ and $d_{i,k+1,W} = 0$, for the disjonctive constraints (14) that write a trivial constraints in the other cases thanks to (12), we define $\overline{S}_i = \max_k \mathbf{S}_{i,k}$. (15) is a common framework for scheduling constraints from CT14 to CT21, which was noticed independently in [18], [2] and [4].

$$\forall c_{14}, w, \quad \sum_{(i,k)\in \mathbf{A14}^{c_{14}}} (d_{i,k,w} - d_{i,k,w-(\mathbf{Da}_{i,k}+\mathbf{Se14}^{c_{14}})^+}) \leqslant 1 \tag{17}$$

$$\forall c_{15}, w \in W_{c_{15}}, \quad \sum_{(i,k)\in \mathbf{A15}^{c_{15}}} (d_{i,k,w} - d_{i,k,w-(\mathbf{Da}_{i,k}+\mathbf{Se15}^{c_{15}})^+}) \leqslant 1 \tag{18}$$

$$\forall c_{16}, w, \quad \sum_{(i,k)\in \mathbf{A16}^{c_{16}}} (d_{i,k,w} - d_{i,k,w-\mathbf{Se16}^{c_{16}}}) \leqslant 1 \tag{19}$$

$$\forall c_{17}, w, \quad \sum_{(i,k)\in \mathbf{A17}^{c_{17}}} (d_{i,k,w-\mathbf{Da}_{i,k}} - d_{i,k,w-\mathbf{Da}_{i,k}-\mathbf{Se17}^{c_{17}}}) \leqslant 1 \tag{20}$$

$$\forall c_{18}, w, \quad \sum_{(i,k)\in \mathbf{A18}^{c_{18}}} (d_{i,k,w} - d_{i,k,w-\mathbf{Se18}^{c_{18}}}) + (d_{i,k,w-\mathbf{Da}_{i,k}} - d_{i,k,w-\mathbf{Da}_{i,k}-\mathbf{Se18}^{c_{18}}}) \leqslant 1 \tag{21}$$

$$\forall c_{19}, w, \quad \textstyle\sum_{(i,k)\in \mathbf{A19}} (d_{i,k,w-\mathbf{L19}^{c_{19}}_{i,k}} - d_{i,k,w-\mathbf{L19}^{c_{19}}_{i,k}-\mathbf{Tu19}^{c_{19}}_{i,k}}) \leqslant \mathbf{Q19}^{c_{19}} \tag{22}$$

$$\forall c_{20}, w, \quad \sum_{(i,k)\in \mathbf{A20}^{c_{20}}_w} (d_{i,k,w} - d_{i,k,w-\mathbf{Da}_{i,k}}) \leqslant \mathbf{N20}^{c_{20}}_w \tag{23}$$

$$\forall c_{21}, w, \quad \sum_{i,k} \mathbf{Pmax}^w_i (d_{i,k,w} - d_{i,k,w-\mathbf{Da}_{i,k}}) \leqslant \mathbf{Imax}^{c_{21}}_w \tag{24}$$

4 Uncertainty model and 2-stage robust formulation

Operations for maintenance and refuelling are very complex, scheduling a lot of tasks with different manufacturers. Uncertainties can arise in the maintenance, which can have an impact in term of costs and feasibility of the final solutions. Operational requirements guided to a robust management of outage prolongations. $\mathbf{Da}_{i,k}$ is thus uncertain.

Uncertainty sets The originality of our robust problematic compared to [13, 11] is that $\mathbf{Da}_{i,k}$ are discrete, and constraints like (7) or (17) are non linear in $\mathbf{Da}_{i,k}$. This leads to consider a discrete enumeration of scenario, we denote Ω such uncertainty set. For all $\delta \in \Omega$, we denote by $\delta_{i,k}$ the outage prolongation expressed as an integer number of weeks, outage durations are now $\mathbf{DA}_{i,k}+\delta_{i,k}$. We denote $\overline{\mathbf{Da}_{i,k}} = \max_{\delta\in\Omega} \mathbf{Da}_{i,k}+\delta_{i,k}$ the maximal prolongation of outage (i,k). We assume that $\delta_{i,k} \in [\![0, \overline{\delta_{i,k}}]\!] = \mathbb{Z} \cap [0, \overline{\delta_{i,k}}]$. A first natural uncertainty set is $\Omega_{worst} = \prod_{i,k} [\![0, \overline{\delta_{i,k}}]\!]$. In this case, all the worst prolongation can be considered, which is very pessimistic. One way to reduce the conservatism is to restrict the number of prolongations using a cardinality constraint, like in [13]: $\Omega^N_{card} = \left\{ \delta \in \Omega_{worst} \middle| \delta_{i,k} \leqslant \overline{\delta_{i,k}}\epsilon_{i,k} \text{ and } \sum_{i,k} \epsilon_{i,k} \leqslant N,\ \epsilon_{i,k} \in \{0,1\} \forall i,k \right\}$.

Generic form of deterministic MIP Whatever the deterministic formulation considered following variants of sections 4 and 5, we can write the deterministic problem with following form:

$$\min_{x\in\{0,1\}^n\times\mathbb{R}^m_+, y\geqslant 0} \quad cx + q\,y \tag{25}$$

$$A\,x \quad \leqslant a \tag{26}$$

$$T_\delta\,x + W\,y \leqslant h \tag{27}$$

$$B\,y \quad \leqslant b \tag{28}$$

x denotes variables $d_{i,k,w}$ et $r_{i,k}$, real decision variables, whereas y denotes second level variables of productions and fuel stocks, only continuous variables. (26) express coupling among first level variables, with constraints (2), (3), (4), (8) and some scheduling constraints, this will be discussed below case by case. (27) denotes constraints duplicated for all scenario δ, that couples first level to second level variables, typically constraints (7), (6) (5), (9), (10), (13), (12) and eventually TODO.

2 level robust scheme The problem structure is similar to 2-stage robust problems: first stage variables x are "here and now" decisions, whereas production decisions y "wait and see" are implemented once the uncertainty is revealed. This situation is similar to game theory problems, analogously with a fictive adversary choosing the best strategy in Ω to penalize outages prolongations after the date decisions are fixed, with the possibility to adjust the production after the choice of the adversary. It leads to the min-max-min scheme:

$$\begin{array}{lll} \min\limits_{x,y\geqslant 0} cx + \mathcal{Q}(x) & \text{with } \mathcal{Q}(x) = \max_{\delta\in\Omega} \min_y qy & \\ s.c: \ Ax \leqslant a & & s.c: T_\delta\, x + W\, y \leqslant h \\ \quad x \in \{0,1\}^n \times \mathbb{R}_+^m & & \quad B\, y \leqslant b \end{array} \tag{29}$$

The min-max-min choice of x is conditioned for all occurrence $\delta \in \Omega$ by the feasibility of second level solutions y. Otherwise, the fictive adversary would choose in his maximization a $\delta \in \Omega$ leading to infeasibility in y, such that $\mathcal{Q}(x) = +\infty$. With such paradigm, the choice of x is based on the best objective value after recourse on the worst case happening in the uncertainty set.

MIP Linearization To linearize the min-max-min problem, we duplicate variables y and constraints for all scenario δ. It gives rise to variables y_δ and duplicated constraints $T_\delta\, x + W\, y_\delta \leqslant d$. We can also introduce a continuous variable C^{rob}, that will be $C = \max_{\delta\in\delta} \min q(\delta)y$, thanks to constraint (30.4). It leads to following linearization :

$$\begin{array}{lll} \min\limits_{x\in\{0,1\}^n\times\mathbb{R}_+^m, y_\delta\geqslant 0} & cx + C^{rob} & (30.1) \\ & A\, x \leqslant a & (30.2) \\ \forall\delta & T_\delta\, x + W\, y_\delta \leqslant h & (30.3) \\ \forall\delta & B\, y_\delta \leqslant b & (30.4) \\ \forall\delta & q\, y_\delta \leqslant C^{rob} & (30.5) \end{array} \tag{30}$$

Such matrix structure is similar to stochastic optimization with recourse, with coupling variables x, C^{rob} and independent sub-problems for all scenario δ. Specialized Benders reformulation is thus naturally investigated.

Benders cuts Once first level variables C^{rob} et x are fixed, the problem is restricted to feasibility questions: for all scenarios δ, do it exist a production plan y_δ, fulfilling the constraints with a cost lower than C^{rob}? We have as many feasibility questions as scenarios, independent sub-problems. The feasibility question for constraints (27), (28) et (30.5) an d a scenario δ, can be reformulated as following optimization problem :

$$\begin{array}{lll} \eta^* = \min\limits_{\eta, y_\delta\geqslant 0} \eta \ (= -\max -\eta) & & \\ W\, y_\delta \leqslant \eta\mathbb{1} + h - T_\delta\, x & (\alpha) \\ B\, y_\delta \leqslant \eta\mathbb{1} + b & (\beta) \\ q\, y_\delta \leqslant \eta + C^{rob} & (\gamma) \end{array} \tag{31}$$

$\mathbb{1}$ is a vector filled with 1. Feasibility of (27), (28) et (30) is equivalent to $\eta^* = 0$. (31) is feasible, so using strong duality, we have:

$$-\eta^* = \min_{\alpha,\beta,\gamma\geqslant 0} \alpha.(h - T_\delta x) + \beta.b + \gamma.C^{rob} \tag{32}$$

$$\alpha.W + \beta.b + \gamma.q \geqslant 0 \qquad (y_\delta) \tag{33}$$

$$-\alpha.\mathbb{1} - \beta.\mathbb{1} - \gamma.\mathbb{1} \geqslant -1 \qquad (\eta) \tag{34}$$

We recognize, extreme directions of cone defined by constraints (33), . The constraint matrix does not depend on first stage decisions and on δ. This allowes to reformulate (30) on x variables with master constraints (26), and all extreme direction (α, β, γ) of cone (33):

$$\min_{x\in\{0,1\}^n\times\mathbb{R}_+^m, C^{rob}\geqslant 0} \quad cx + C^{rob} \tag{35}$$

$$A\, x \leqslant a \tag{36}$$

$$\forall(\alpha,\beta,\gamma)\ \alpha.(h - T_\delta x) + \beta.b + \gamma.C^{rob} \geqslant 0 \tag{37}$$

Furthermore, we provided a constructive separation algorithm to generate cuts, computing (31) to generate (α, β, γ). This Benders decomposition allows to generate one cut per scenario.

Specific case of CT14-CT15 scheduling constraints Scheduling constraints CT14-CT21 are interesting to study separately, because of different structures regarding outage durations $\mathbf{Da}_{i,k,\delta}$. CT16 constraints do not depend outage durations $\mathbf{Da}_{i,k,\delta}$, constraints (19) can thus be written in the master problem. The case of CT14 and CT15 is also specific, these constraints can be reformulated with RHS uncertainty (we refer to [18, 19]). The robustness us equivalent to Soyster approach, the reformulation of [20] proves that robust CT14 and CT15 constraints are equivalent to the deterministic constraints with $\mathbf{Da}_{i,k} = \overline{\mathbf{Da}}_{i,k}$, that can also be incorporated in the master problem:

$$\forall w, \sum_{(i,k)\in\mathbf{A14}}(d_{i,k,w} - d_{i,k,w-(\overline{\mathbf{Da}}_{i,k}+\mathbf{Se14})^+}) \leqslant 1 \tag{38}$$

$$\forall w \in [\mathbf{d}_{15}, \mathbf{f}_{15}], \sum_{(i,k)\in\mathbf{A15}}(d_{i,k,w} - d_{i,k,w-(\overline{\mathbf{Da}}_{i,k}+\mathbf{Se15})^+}) \leqslant 1 \tag{39}$$

5 Parametric robustified formulation

For the previous formulation, feasible set of robust solution induce feasibility of solutions for all single outage prolongation. In the case where no such robust solution exists, or have a prohibitive cost, a lighter definition of robustness is required to have consistent robustified solutions.

MIP robustified formulation The further robustified formulation consider the impact of prolongation only for the feasibility of constraints CT14 and CT15, allowing violations of robust constraints with a cost penalization. For all constraints $c \in CT14$ and $c \in CT15$, continuous variables $z^{(14)}_{c,w}, z^{(15)}_{c,w} \geqslant 0$ are introduced to penalize robust violations, paying cost $\mathbf{Cpen}^{rob}$ for violations. This gives rise to a MIP formulation similar to the deterministic formulation, minimizing $\sum_w \mathbf{Cpen}^{rob}(z^{(14)}_{c,w} + z^{(15)}_{c,w}) + f^{det}_{obj}$ with $f^{det}_{obj} = \sum_{i,k} \mathbf{C}^{rld}_{i,k} r_{i,k} + \sum_{i,k,w} \mathbf{C}^{pen}_{i,k,w}(d_{i,k,w} - d_{i,k,w-1}) + \sum_{j,t} \mathbf{C}^{prd}_{j,t} \mathbf{D}^t\, p_{j,t} - \sum_i \mathbf{C}^{val}_i x^{fin}_i$, adding to the previous deterministic formulation the constraints:

$$\forall w, c \in CT14 \sum_{(i,k)\in\mathbf{A14}^c}(d_{i,k,w} - d_{i,k,w-(\overline{\mathbf{Da}}_{i,k}+\mathbf{Se14})^+}) \leqslant 1 + z^{(14)}_{c,w} \tag{40}$$

$$\forall c \in CT15, w \in [\mathbf{d}^c_{15}, \mathbf{f}^c_{15}], \sum_{(i,k)\in\mathbf{A15}^c}(d_{i,k,w} - d_{i,k,w-(\overline{\mathbf{Da}}_{i,k}+\mathbf{Se15})^+}) \leqslant 1 + z^{(15)}_{c,w} \tag{41}$$

Constructive matheuristic resolution To face the large size of instances, heuristic decompositions are useful to compute quickly primal solutions. The advantages of MIP facility to model and solve the problem on reduced instances. A first approach is to built solutions iteratively in a relax-and-fix procedure, to compute successively the solutions for cycles k once the cycles $k' < k$ are fixed by previous optimisation, relaxing continuously the cycles such that $k' > k$. An other approach is to proceed unit by unit, similarly to a greedy heuristic.

Algorithm 1: POPMUSIC VND with MIP neighbourhoods

Input: an initial solutions, a set and order of neighbourhoods to explore
Initialisation: `currentSol` = initSolution, $\mathcal{N}$ =initial neighbourhood.
while the stopping criterion is not met
    define the MIP with incumbent `currentSol` and the neighbourhood $\mathcal{N}$)
    define `currentSol` as warmstart
    `currentSol` = solveMIP(MIP,timeLimit($\mathcal{N}$))
    $\mathcal{N}$ = `nextNeighborhood`($\mathcal{N}$)
end while
return CurrentSolution

VNS resolution Once a feasible solution is built with previous constructive matheuristics, a VND iterates in a local search computing iterations with B&B resolution with MIP neighbourhoods. The current solution is the primal solution given by the last B&B resolution and it is also defined as warmstart for the next B&B resolution to improve the efficiency of B&B primal heuristics, enabling RINS or Local Branching heuristics from the beginning. This ensures that the solution given by the MIP resolution is at least as good as the current solution, this algorithm is thus a

steepest descent algorithm. The stopping criterion could be a maximal time limit or a maximal number of iterations, or being in a local extremum for all neighbourhoods. The key point is the neighbourhoods definition and description. Neighbourhoods are defined with three characteristics:

- The **restriction of search space**: Variable fixations or other extra constraints that the current solution satisfies to limit the MIP combinatoric to have an easier B&B resolution.
- a **B&B stopping criterion**: it must defined so that the B&B resolution is efficient in a short resolution time.
- a **specific parametrization of the MIP resolution**: for an efficient B&B resolution in the defined time limit.

The multi-index structure allows to define partitioning neighbourhoods:

- $\mathcal{N}_I^{units}$ unit selection: only units $U \in \mathcal{U}$ are reoptimised. Successive neighbourhoods defining a partition of $\mathcal{I}$ are computed iteratively.
- $\mathcal{N}_{[\underline{w},\overline{w}]}^{TW}$: all outages are reoptimised in the time window $[\underline{w},\overline{w}]$.
- $\mathcal{N}_{k,k'}^{cycles}$: variables relative cycles k'' with $k \leqslant k'' \leqslant k'$ are reoptimised.

Algorithm 2: Pareto front computation with VND and MIP neighbourhoods

Initialisation: compute `bestObjective` with VND resolution $\mathbf{Cpen}^{rob} = 0$.
compute `currentObjective,robustViolations` with VND resolution with a high value of $\mathbf{Cpen}^{rob}$.
Let `robustViolations` the number of violations of robust constraints of the solution of `currentObjective`.
`Pareto front <-` (`currentObjective,robustViolations`)
while `currentObjective>bestObjective`
    define `currentObjective,robustViolations` as wamstart for VND
    `currentObjective` = solve VND with at most `1+robustViolations` robust violations allowed
    `robustViolations = 1+robustViolations`
    `Pareto front <-` (`currentObjective,robustViolations`)
end while
return `Pareto front`

Multi-objective optimization Cost optimisation and robustness can be concurrent objectives. Calculating Pareto fronts of best compromise cost/robustness solutions are interesting. Optimizing with variations $\mathbf{Cpen}^{rob}$ of provides points of the Pareto front. To compute more efficiently the Pareto front, VND can be derived, constructing solutions from the more to the less robust. First Pareto point is built with a high value of $\mathbf{Cpen}^{rob}$ by VND. This point is also a feasible solution with 1 more robust violation allowed, defining a warmstart for the computation of the next Point of the Pareto front:

6 Computational results

Tests were computed with a laptop running Linux Ubuntu 12.04 with an Intel Core2 Duo processor, 2.80GHz. Our implementation used the modeling language OPL to solve MIP with Cplex 12.5 and OPL script to iterate MIP computations. We used the dataset from the EURO/ROADEF 2010 challenge. Instances from dataset B and X are representative from the real-life instances, B8 and B9 were much more combinatorial, due to the lack of time windows constraints for cycles $k \geqslant 3$, these instances are representative from the real life instances.

6.1 Computational results on deterministic formulations

We analyse now the characteristics of the resolution of the deterministic problem of section 3. Without stability cost ($\mathbf{C}_{i,k,w}^{pen} = 0$), the problem is similar to the ROADEF Challenge, the frontal resolution allows to analyse hypothesis of [2, 4, 3]. First results justify the commonly used simplification to aggregate production time steps to weekly time steps, and to aggregate scenarios. Using

Cplex 12.3, the frontal resolution was completely inefficient instances B8 et B9 even with time steps and scenarios aggregation and relaxation of difficult constraints CT6 and CT12, 1h resolution time is not enough to compute the LP relaxation. The size of these instances is a strong limiting factor. For the other instances, the frontal MIP resolution is efficient, with low gaps between the best primal and dual bounds in 1h resolution time.

Using Cplex 12.5, LP relaxations can be computed for all instances in 1h. However, the solutions of the LP relaxation are not useful to compute primal solutions for B8 and B9. In these cases, several production cycle can overlap and the LP relaxation gives very few integer variables, specially for last cycles. Hence, this is a big handicap for MIP primal heuristics, relying on the LP relaxation such as Feasibility Pump, RINS or Local Branching. The MIP frontal resolution is always inefficient on B8 and B9.

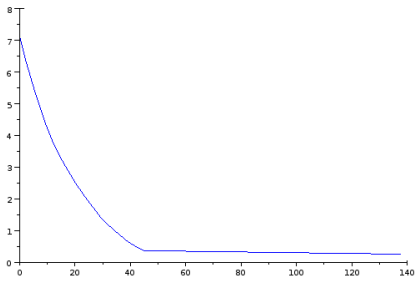


Fig. 1. Instance B7-3-120

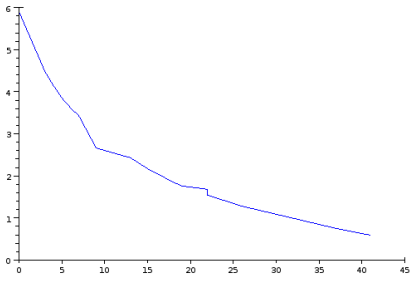


Fig. 2. Instance X12-3-120

Fig. 3. Pareto fronts for the trade off financial cost/solution stability

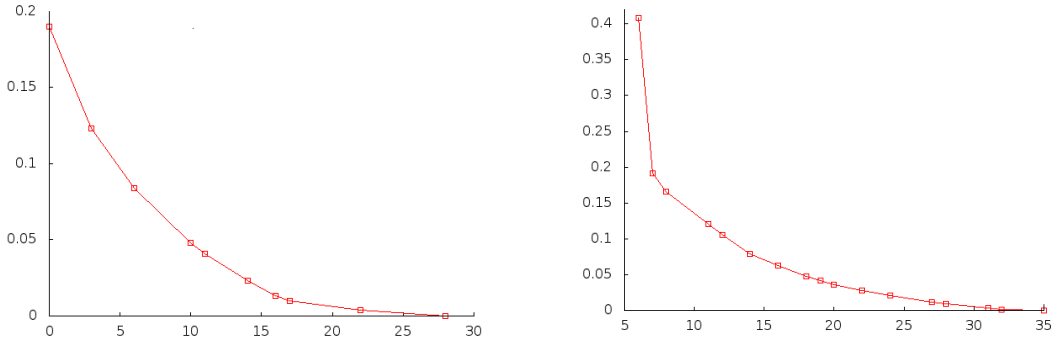
Adding stability costs change fundamentally the MIP convergence characteristics, guiding the solution search around the baseline solution improves very significantly the resolution time. An explication is that a lot of solution with similar cost exist with $\mathbf{C}^{pen}_{i,k,w} = 0$ which was already notices for the Challenge. It induces "pseudo-symetries" in the Branch&Bound tree search which is known to be a bottleneck for the Branch&Bound method. An application of these properties is that it offers efficient reparation of partial infeasible solution built by a constructive heuristic, with a high penalization of the distance to the partial solution, similarly to Feasibility Pump. The computation of Pareto fronts starting with a poor first solution shows the impact of a good baseline solution, we refer to Figure 3.

Constructive Matheuristics of section 5 applies to furnish feasible solutions for all instances. Relax and fix approach gave very good solutions, but required sometimes long calculus time to converge to solution in the MIP iterations. The greedy POPMUSIC strategy, using the previous repairing strategy using a partial solution combining optimal solutions for single unit optimisation allows to build quickly primal solutions for all instances, but with a lower quality of primal solutions than relax and fix strategies. The VND local search applied starting with the greedy POPMUSIC solutions is very efficient, improving quickly primal solutions and requiring few VND iterations to compute a local minimum for all neighbourhoods. The resulting matheuristic is more efficient in the ration improvement of solution/computation time than the relax and fix heuristics. Furthermore, comparing the local extrema given by the VND to the optimal solutions of small instances (instances were truncated to obtain optimally proven solutions), we had no example where the optimal solutions were better than the VND solutions more than the tolerance gap of Cplex parametrized to $0,01\%$. The tolerance gap of Cplex of $0,01\%$ induces sometimes that the VND solutions were better than the "optimal" frontal solutions, within the $0,01\%$ tolerance gap.

6.2 Computational results on robust formulations

First results with robust formulation of section 4 give few feasible solutions, even with the tiny uncertainty set considering a scenario for all unit i, where all the prolongations are null except the cycle $k = 1$ of unit i which have a prolongation of 1 week. One reason is that the scheduling

constraints CT14 and CT15 have a big impact and there is few margins with such constraints to provide a feasible planning. Definition of robustness must also be relaxed for operational considerations. Furthermore, Benders decomposition of section 4 is challenging to manage rounding errors: dual variables computed in subproblems introduce rounding errors in the constraints (37), each iteration of Benders decomposition adding rounding errors.



On instance B7-3-120 On instance X12-3-120

Fig. 4. Pareto fronts for the trade off financial over-cost/robustness

Parametric approach of section 5 is conceptually simpler, but it provides solutions for all instances following the operational requirement. Resolution characteristics are similar to the deterministic problem, the VND approach provides easily solutions for all instances like in [19]. We already mentionned that the deterministic formulation without stability cost induces lots of solutions close to the optimal cost. Robustified formulation allows to trade-off among the best quality solutions to provide operationally interesting solutions. Furthermore, this breaks pseudo-symetries which is helpful for a Branch&Bound resolution.

The parametric resolution allowed to compute Pareto front of good compromise robustness/cost solutions for all sizes of instances. Several configurations appeared, which is illustrated Figure 4. On instance `B7_3_120`, a robust solution exists for all uncertainty scenario with a low overcost to the optimal solution (0,2%). On instance `X12_3_120`, no feasible solution exists to be resistant for all the uncertainty scenario. The most robust solution violating the least robust constraints has an over-cost of $0,4\%$, over-costs decreasing quickly. In all cases, Pareto curves are interesting for decision support, for a further choice of decision makers to trade-offf risk cover and financial over-costs. Over-costs due to over-costs are generally low, which justifies to introduce robustness for the industrial application.

7 Conclusions and perspectives

Conclusions Multi-objective extensions of the 2010 EURO/ROADEF 2010 Challenge were investigated in this work. Our extensions are useful in the problematic of monthly reoptimisations, where stability and robustness of the planning are crucial for the operational application. This study is operationally useful as the 2010 ROADEF challenge furnish lots of solution with similar costs, do decide on the solution to implement among the best ones with operational trade-off. Penalizing or bounding the distance to the initial solutions allows to address stability questions. This extension allows to have a MIP formulation omitting constraints CT6 and CT12 to formulate the problem with less continuous variables. Considering stability objectives or constraints accelerates the MIP frontal resolution, and allows to trade off interesting operational solutions.

Introducing uncertainty and robustness to outage duration can lead to 2 stage robust min-max-min formulation. Anyway, the resolution with Benders reformulation is limited in the resolution capacities, with difficulties induced by numerical stability. Furthermore, this formulation is too conservative to give consistently solutions on the ROADEF dataset. A lighter robust version, penalizing robust infeasibilities is consistent, for a 2-objective resolution computing Pareto fronts with a VNS algorithm relying on the deterministic MIP formulation.

Perspectives Financial stakes to arbitrate cost/stability/robustness are promising in the challenge dataset for the operational application, these results should be consolidated on real data. Multi-objective optimisation is an appropriate way to compute the maintenances of nuclear power plants. Other objectives are interesting for a multi or many objective approaches for a sustainable development optimization: minimizing nuclear wastes by consuming at most the fuel during outages, minimizing carbon emissions ...

References

1. Gardi, F., Nouioua, K.: Local search for mixed-integer nonlinear optimization: a methodology and an application. Lecture Notes in Computer Science **6622** (2011) 167–178
2. Lusby, R., Muller, L., Petersen, B.: A solution approach based on benders decomposition for the preventive maintenance scheduling problem of a stochastic large-scale energy system. Journal of Scheduling **16** (2013) 605–628
3. Rozenknopf, A., Calvo, R.W., Alfandari, L., Chemla, D., Létocart, L.: Solving the electricity production planning problem by a column generation based heuristic. Journal of Scheduling (2012)
4. Jost, V., Savourey, D.: A 0-1 integer linear programming approach to schedule outages of nuclear power plants. Journal of Scheduling **16** (2013) 551–566
5. Anghinolfi, D., Gambardella, L., Montemanni, R., et al.: A matheuristic algorithm for a large-scale energy management problem. Lecture Notes in Computer Science **7116** (2012) 173–181
6. Brandt, F.: Solving a large-scale energy management problem with varied constraints. Master's thesis, Karlsruhe Institute of Technology (2010)
7. Brandt, F., Bauer, R., Völker, M., et al.: A constraint programming-based approach to a large-scale energy management problem with varied constraints. Journal of Scheduling **16** (2013) 629–648
8. Gavranović, H., Buljubasić, M.: A hybrid approach combining local search and constraint programming for a large scale energy management problem. RAIRO Operations Research **47** (2013) 481–500
9. Godskesen, S., Jensen, T., Kjeldsen, N., Larsen, R.: Solving a real-life, large-scale energy management problem. Journal of Scheduling **16** (2013) 567–583
10. Froger, A., Gendreau, M., Mendoza, J., Pinson, E., Rousseau, L.: Maintenance scheduling in the electricity industry: A literature review. European Journal of Operational Research (2015)
11. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. Mathematical Programming **98** (2003) 49–71
12. Thiele, A., Terry, T., Epelman, M.: Robust linear optimization with recourse. Technical report (2009)
13. Atamtürk, A., Zang, M.: Two-stage network flow and design under demand uncertainty. Operations Research **55** (2007) 662–673
14. Herroelen, W., Leus, R.: Robust and reactive project scheduling: a review and classification of procedures. int. j. prod. res. **42** (2004) 1599–1620
15. Simovic, T., et al.: Challenge ROADEF/EURO 2010: a large-scale energy management problem with varied constraints. EDF R&D Technical Report (2010)
16. Dupin, N.: Tighter mip formulations for the unit commitment problem with min-stop ramping constraints. (To appear in EURO Journal of Computationial Optimization)
17. Yıldız, S., Vielma, J.P.: Incremental and encoding formulations for mixed integer programming. Operations Research Letters **41** (2013) 654–658
18. Joncour, C.: Problèmes de placement 2D et application à l'ordonnancement: modélisation par la théorie des graphes et approches de programmation mathématique. PhD thesis, Bordeaux (2010, pp 147-166)
19. Dupin, N.: Modélisation et résolution de grands problèmes stochastiques combinatoires: application à la gestion de production d'électricité. PhD thesis, Lille 1 (2015)
20. Soyster, A.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research **21** (1973) 1154–1157

A new hybrid method to solve the multi-objective optimization problem for a composite hat-stiffened panel

Ahmad El Samrout[1] *, Oussama Braydi[1,2], Rafic Younes[2] and Francois Trouchu[3], and Pascal Lafon[1]

[1] University of Technology of Troyes, Institut Charles Delaunay, France
[2] Faculty of engineering, Lebanese University, Hadath, Lebanon
[3] Montreal Polytechnic, Montreal, Canada

Abstract. In this paper we present a new hybrid meta heuristic by combining Multi-objective bat algorithm (MOBA) and variable neighborhood search (VNS). The hybrid meta heuristic is coupled with response surface methodology (or meta modeling) to solve the mechanical multi-objective optimization problem of hat stiffened composite panel. The optimization criteria were the weight and the rigidity of the panel. Experimental results show that our suggested approach is quite effective, as it provides solutions that are competitive with the results obtained by using MOBA alone.

Keywords: hat stiffened panel, multi-objective optimization, meta modeling, multi-objective bat algorithm,variable neighborhood search

1 Introduction

The design of industrial structures in aeronautics is a constantly evolving field, due to the perpetual need to gain weight and space in this domain. A lighter aircraft means fewer loads to compensate, and thus less fuel consumption, which decreases the greenhouse effect. Therefore, a lighter aircraft is beneficial at both economic and environmental levels.
Composite material has been used in aerospace construction because of their high stiffness to weight ratio and their resistance to fatigue and corrosion. In particular, stiffened composite panels are widely used in aircraft's fuselage, as well as in wings and tail sections. However, such materials induce an additional weight to the overall structure. Therefore, a rigidity-weight optimization of stiffened panels becomes a necessity in aerospace industry.

In [1] a new method for manufacturing self-stiffened composite panels using flexible injection was presented. Some mechanical properties of the new panels, such as stiffness, were then evaluated using a three points bending test. The

* ahmad.el_samrout@utt.fr

author studied the influence of the stiffener on the mechanical properties of the composite.

The aim of our study is to improve the hat stiffened panels produced in [1] using meta heuristics and response surface methodology. Two objective functions are considered; the weight minimization and the rigidity maximization.
The kind of problem is well known in panel optimization literature. While some researchers try to optimize the geometrical shape of the panel [2], [3], [4], others rely on the stacking sequence of the composite panel as their optimization variable [5], [6], [7], [8].
The use of meta modeling and multi objective meta heuristics is also well known in these optimization problems, for instance [9] presented an optimization procedure for a geometric design of a composite material stiffened panel with conventional stacking sequence using static analysis and hygrothermal effects. The procedure is based on a global approach strategy, composed by two steps: first, the response of the panel is obtained by a neural network system using the results of finite element analyses and, in a second step, a multi-objective optimization problem is solved using a genetic algorithm. In [10] a process to compare three genetic algorithms (GAs) for the solution of multiobjective optimization problem of a T-shape composite stringer under compression loads has been presented.

This paper is organized as follows. In section 2 the geometrical dimensions, physical proprieties, boundary conditions and finite element model of the panel are illustrated. In section 3 the formulation of the optimization problem along with the meta model and the hybrid meta heuristic are introduced. It is also devoted for the presentation and the interpretation of the results. Finally, conclusion is given in section 4.

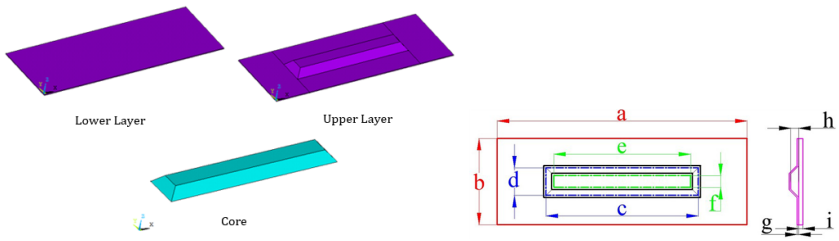
2 Model presentation and validation

The author in [1] developed a new procedure to produce composite plates of size $400 \times 140 \times 3mm$ reinforced with a centrally located Omega feature, then he tested these panels using three points bending tests that he called omega test and inverse omega test.

2.1 Panel composition and geometry

The sandwich hat-stiffened composite panel consists of three components: an upper and a lower composite layers, separated by a foam core (see Figure 1.a)

The geometric parameters of this panel are presented in Figure 1.b. Eight variables can represent the panel in a sufficient manner, but since we are trying to study the influence of foam core, it is safe to assume that $a = 406mm$ and $b = 140mm$ are constant. Also for the sake of simplicity we will assume that i is equal to g. Table 1 shows the initial values of the panel's variables.



(a) (b)

Fig. 1: a) Panel composition , b) Panel geometry.

Table 1: variable initial values

variables	c	d	e	f	g	h
values [mm]	260	46	240	20	1.5	14

2.2 Material Proprieties

The core is made from Foam $K20(E = 1.5GPa; \nu = 0.3)$. The upper and lower layers and the contour are made from glass fiber-epoxy. The ply lay-up is $[90, 0, 90]$ with a total number of 3 plies, each has a thickness of $0.47mm$.

2.3 Meshing

The meshing (see Figure 2) is divided into two steps, the first step is to generate the mesh of the core with tetrahedral elements. The second step is to generate the skins of mesh (lower and upper) with hexahedral elements in each element with 8 nodes and consists of 3 sections 'Shell' (90/0/90) with respect to the main axis X. Orientation of the elements was taken into account.

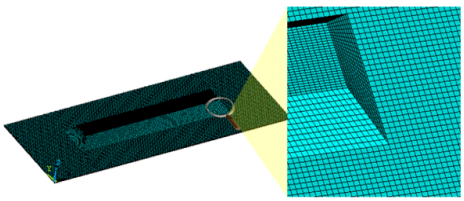
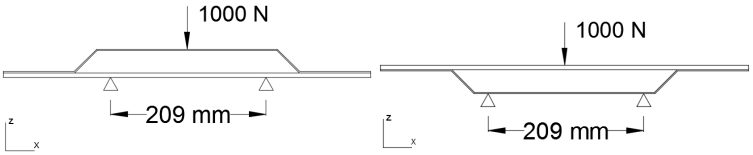


Fig. 2: Meshing

2.4 Loading and Boundary conditions

Our model will treat only the elastic behavior of the panel. According to [1], the composite panel will act as an elastic panel while the bending force is under $2000N$. In our finite element model a static force of $1000N$ will be applied to mid plan of the panel (upper layer in the case of omega test, and lower layer in the case of inverse omega). While two supports distant by $209mm$ hold the panel on the opposite side (see Figure 3).



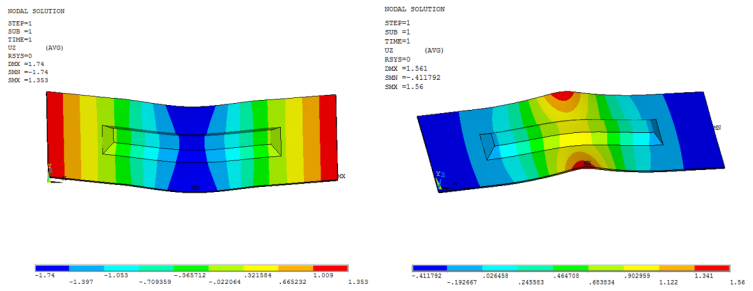
(a) (b)

Fig. 3: Loading a) Omega loading , b) inverse omega.

The displacement along axis X, Y, and Z will be blocked for the supports. To represent the force, the displacement along axis X and Y will be blocked along the central support and the loading is distributed across all nodes in the Z direction ($1000/50 = 20N$ per node for omega and $1000/72 = 13.88N$ per node for inverse omega).

2.5 Result and verification

The results of the finite element simulation are shown in Figure 4.



(a) (b)

Fig. 4: Finite element results tests a) omega , b) inverse omega.

The comparison of stiffness values for the composite panel, in the elastic range, between the experimental results and the finite elements results, for both omega and inverse omega is given in Table 2. This small range of error justifies our model.

Table 2: Comparison between theoretical and experimental results

	experimental results [N mm^{-1}]	finite elements results [N mm^{-1}]	Error %
OMEGA	580.75	574.71	1.05
INV OMEGA	584.43	640.61	8.77

3 Multi objective optimization

The two objectives of the optimization problem are to maximize the rigidity R and to minimize the weight W. The design variables are the dimensions of the panel $X = \{c, d, e, f, g, h\}$. The values of the lower bound X_l and upper bound X_u of X are shown in Table 3.
Therefore, the multi-objective optimization problem can be formulated as follows:

$$Minimize\ F_{obj}(W(X), \frac{1}{R(X)}), \tag{1}$$

$$Subject\ to: \quad X_l < X < X_u \tag{2}$$

Table 3: Upper and lower bounds for each variable

Variables	c	d	e	f	g	h
Lower bound X_l[mm]	234	44	200	18	1.24	11.4
Upper bound X_u[mm]	266	52	249	26	1.56	14.6

3.1 Meta Modeling

Finite element analysis usually costs a huge computational time. A common solution to this problem is to use Meta modeling. A Meta model or surrogate

model is a model of the model, i.e. a simplified model of an actual model. In our case, the Meta model will try represent the mechanical behavior of the panel with response to the bending experiences. It should be able to predict the deformation of the plate, and hence it's rigidity simply by knowing the geometrical dimensions of the plate. Various types of Meta models include polynomial equations, neural network, Kriging, etc. These types often share the same steps; first a design of experiments is established where a certain number of experiments is done, each time with a different set of inputs. Then a systematic method to determine the relationship between inputs affecting the process and the output of that process is applied.
In our example the cubic face centered design was adopted as design of experiments, then the finite element experience was repeated 80 times. The Meta model used in our example to find the correlation between the deformation of the panel and its geometrical parameters is Kriging which is a method of interpolation for which the interpolated values are modeled by a Gaussian process governed by prior covariances. It is widely used in the domain of spatial analysis and computer experiments.
By calculating the error percentage, i.e. the difference between the response of the finite element model and the response of the meta model, one can determine the degree of accuracy of our meta model in both omega and inverse omega tests. In the case of omega, the error is smaller than $4 \times 10^{-2}\%$, while in the case of inverse omega the error is around $3 \times 10^{-6}\%$.

3.2 Hybrid algorithm

In this section, we attempt to optimize the meta model obtained in 3.1 w.r.t the problem formulated in Equation 1 using a new hybrid based on MOBA and VNS.

Multi-objective bat algorithm (MOBA) was first introduced in [11]. It is a meta heuristic that imitates the bat's echolocation system. It can be summarized as follows: Each virtual bat flies randomly with a velocity v_i at position (solution) x_i with a varying frequency or wavelength $\varphi \in [\varphi_{min}, \varphi_{max}]$ and loudness A_i and where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution and x_* is the current global best location (solution) which is located after comparing all the solutions among all the n bats at each iteration t. As a bat searches and finds its prey, it updated its position and velocity and changes frequency, loudness and pulse emission rate r. Search is intensified by a local random walk. Selection of the best continues until a stop criteria is met.

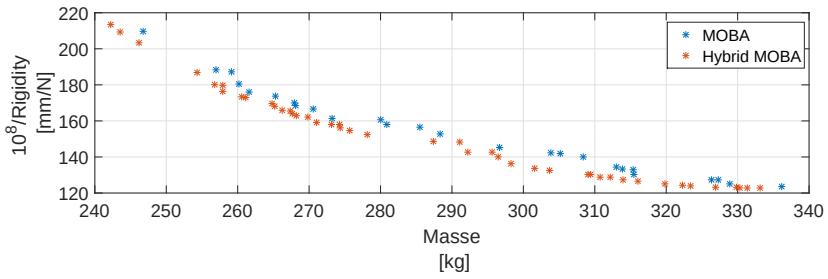
Variable neighborhood search (VNS),was initially proposed by [12]. It is a meta heuristic that explores distant neighborhoods of the current incumbent solution, and moves from there to a new one if and only if an improvement was made. The local search method is applied repeatedly to get from solutions in the neighborhood to local optima.

Hybrid algorithm The choice MOBA and VNS as components for the hybrid algorithm is justified by the fact that these two meta heuristics are complementary to each other; MOBA is a population-based method that is capable of exploring the search space , while VNS is a trajectory-based method that is known for intensifying the search.
In our hybrid algorithm, the solutions obtained by MOBA are taken as initial solutions for VNS. From these "good" solutions, VNS divides the search space into sub-structures and then guides the search aggressively towards better solutions. The algorithm is clarified in Algorithm 1. We implemented the algorithms in MATLAB 2015a language and ran it on a PC with 2.2 GHz and 8 GB RAM memory, with a bat population size of 20 bat and an initial loudness parameter of 0.25 and a pulse rate of 0.5. The frequency range was [0,2].

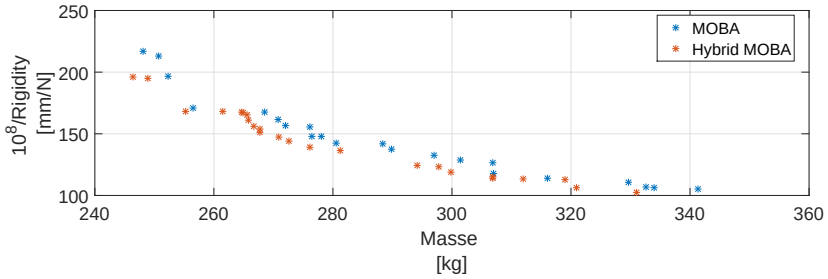
Algorithm 1 hybrid MOBA

procedure MOBA
  Objective functions $f_1(x), ..., f_k(x)$
  Initialize the bat population x_i $(i = 1, 2, \ldots, n)$ *and* v_i
  for $j = 1 \rightarrow N$ *(points on Pareto fronts)* **do**
    Generate K *weights* $w_k \geq 0$ *so that* $\sum_{k=1}^{K} w_k = 1$
    Form a single objective $f = \sum_{k=1}^{K} w_k f_k$
    while $t <$ *Max number of iterations* **do**
      $\varphi_i = \varphi_{min} + (\varphi_{max} - \varphi_{min})\beta$
      $v_i^{t+1} = v_i^t + (x_i^t - x_*)\varphi_i$
      $x_i^{t+1} = x_i^t + v_i^t$
      if *rand* $> r_i$ **then**
        Random walk around a selected best solution
      Generate a new solution by flying randomly
      if $(rand < A_i)$ & $(f(x_i) < f(x_*))$ **then**
        Accept the new solutions
        increase r_i *and reduce* A_i
      Rank the bats and find the current best x_*
    Record x_* *as a non-dominated solution*
procedure VNS
  Divide the set of non dominated solutions obtained in MOBA into k_{max} *structures*
  for *every solution* x *obtained in MOBA* **do**
    $k = 1$
    while $(k \leq k_{max})$ & (*Max number of iterations is not reached*) **do**
      Shaking: *generate a point* $x^{'}$ *at random from the* k^{th} *neighborhood of* x
      Local search: *apply a local search method with* $x^{'}$ *as initial solution;*
      denote with $x^{''}$ *the so obtained local optimum*
      if ($x^{''}$ *is better than* x) **then**
        $x \leftarrow x^{''}$
        $k \leftarrow 1$
      else $k \leftarrow k + 1$

3.3 Results



(a)



(b)

Fig. 5: Pareto frontier a) Omega , b) inverse omega.

Figure 5 shows the Pareto frontier (PF) for both omega and inverse omega cases, obtained using MOBA and hybrid MOBA (MOBA + VNS). All the points on PF are equally "good", and each point represent a set of dimension that guaranties an optimum panel.
In order to evaluate the solutions obtained using MOBA + VNS, and compare them with those obtained using only MOBA, we will use two metrics proposed by [16]:

1. MID (mean ideal distance): The closeness between Pareto solution and ideal point (0, 0). The lower value of MID, the better of solution quality we have.
2. SNS: The spread of non-dominance solution. The higher value of SNS, the better of solution quality we have.

Table 4 gives a comparison between MOBA and hybrid MOBA in omega and inverse omega cases using MID and SNS metrics. It shows clearly that our hybrid model is superior to the original algorithm except for the spread (SNS) in the case of inverse omega test.

Table 4: Comparison between MOBA and hybrid MOBA

	hybrid MOBA omega	MOBA omega	hybrid MOBA inverse omega	MOBA inverse omega
MID	326.8727	330.5729	318.1531	326.7924
SNS	13.6755	13.2973	11.3325	13.7764

4 Conclusion

In this paper we presented a new hybrid meta heuristic based on the combination of MOBA and VNS, because of their complementary strengths. The meta modeling technique is also used to make the optimization more suitable.The hybrid algorithm is tested on a multi-objective optimization problem of hat stiffened composite panel. The optimization criteria were the weight and the rigidity of the panel. Experimental results show that our suggested approach is quite effective, as it provides solutions that are competitive with the results obtained by using MOBA alone.

References

1. Rifay, M.: Fabrication de panneaux auto-raidis par injection flexible. Masters thesis, cole Polytechnique de Montral (2010)
2. Kaufmann, M., Zenkert, D., Mattei, C.: Cost optimization of composite aircraft structures including variable laminate qualities. Composite Science Technology 68(13), 2748–2754 (2008)
3. Rikards, R., Abramovich, H., Kalnins, K. ,Auzins, J.: Surrogate modeling in design optimization of stiffened composite shells. Composite Structures 73 (2), 244–51 (2006)
4. Badran, S.F. , Nassef, AO , Metwalli, S.M.: Y-stiffened panel multi-objective optimization using genetic algorithm. Thin Wall Structures 47 (11), 1331–42 (2009)
5. Irisarri, F.X., Bassir, D.H., Carrere, N., Maire, J.F.: Multiobjective stacking sequence optimization for laminated composite structures. Composites Science and Technology 69 (7-8),983–90 (2009)
6. Irisarri, F.X., Laurin, F., Leroy, F.H., Maire, J.F: Computational strategy for multiobjective optimization of composite stiffened panels. Composite Structures 93 (3), 1158–67 (2011)
7. Todoroki, A., Ishikawa, T.: Design of experiments for stacking sequence optimizations with genetic algorithm using response surface approximation. Composite Structures 64 (3-4), 349–357 (2004)
8. Todoroki, A., Sekishiro, M.: Stacking sequence optimization to maximize the buckling load of blade-stiffened panels with strength constraints using the iterative fractal branch and bound method. Composites Part B: Engineering 39(5), 842–50 (2008)
9. Marín, L., Trias, D., Badalló, P., Rus, G., Mayugo, J.A.: Optimization of composite stiffened panels under mechanical and hygrothermal loads using neural networks and genetic algorithms. Composite structures 94 (11), 3321–3326 (2012)

10. Badalló, P., Trias, D., Marín, L., Mayugo, J.A.: A comparative study of genetic algorithms for the multi-objective optimization of composite stringers under compression loads. Composites Part B: Engineering 47, 130–136 (2013)
11. Yang, X.: Bat Algorithm for the multi-objective optimization. Int. J. Bio-Inspired Comput. 3, 267–274 (2011)
12. Mladenović, N., Hansen, P.:Variable neighbourhood search. Computers and Operations Research 24 (11): 1097-1100 (1997)
13. Badran, S.F., Ashraf, O.N., Sayed M.M.: Y-stiffened panel multi-objective optimization using genetic algorithm. Thin-Walled Structures 47 (11), 1331–1342 (2009)
14. Todoroki, A., Sekishiro, M.: Modified Efficient Global Optimization for a Hat-Stiffened Composite Panel with Buckling Constraint . Tokyo Institute of Technology, Tokyo 152-8552, Japan- DOI: 10.2514/1.34548.
15. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. Transactions on Evolutionary Computation 6(2), 182–197 (2002)
16. Behnamian,J., Fatemi Ghomi,S.M.T., Zandieh,M.: A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. Expert Systems with Applications 36 (8), 11057–11069,(2009)

A Simple and Efficient Variable Neighborhood Structure For The Satisfiability Problem

Noureddine Bouhmala

Department of Maritime Technology and Innovation, Ravein 215, 3184 Borre, Norway.
noureddine.bouhmala@hbv.no

Abstract. The Satisfiability Problem (SAT) is viewed as one of the fundamental optimization problems. This NP-complete problem refers to the task of finding a variable assignment that satisfies all the clauses in a Boolean Formula. Most local search algorithms used to solve SAT problems rely on the 1-flip neighborhood structure. This paper introduces a simple and efficient variable neighborhood strategy. The effectiveness of this neighborhood structure is tested using GSAT-RW and an evolutionary algorithm on various benchmark instances.
Keywords: satisfiability problem, memetic algorithm, variable neighborhood search.

1 Introduction

The SAT problem which is known to be NP-complete [3] is defined as follows. Given a set of n Boolean variables and a conjunctive normal form (CNF) of a set of m disjunctive clauses of literals, where each literal is a variable or its negation which takes one of the two values ***True*** or ***False*** and a positive constant k, the task is to determine whether there exist an assignment of truth values of the variables that satisfies the maximum number k of clauses. SAT still deserves much research attention from a wider community of researchers due to its theoretical and practical importance. It is a widely used modeling framework for simulating complex systems that turn out to be of combinatorial nature. Different state-of-the-art local search algorithms for solving SAT have been developed [1]. Several of these algorithms are enhanced versions of earliest GSAT [13] and WalkSAT [14] algorithms. Examples include GSAT/Tabu [10], WalkSAT/Tabu [11], R-Novelty+ [9] heuristics, variable and clause weighting algorithms [17] , dynamic parameter tuning algorithms [8], adaptive memory-based local search hybrid approaches [7] , larger neighborhood search algorithms [20], Learning Automata [4],, Iterated Robust Tabu Search (IRoTS) [15], and finally algorithms based on a new diversification scheme to prevent cycling [2]. VNS has been applied to a wide variety of problems from combinatorial optimizations. Examples include feature selection in data mining [12], scheduling problem [5], and vehicle routing problem [19]. However, the author is not aware of any paper introducing VNS for solving SAT except the work conducted in [6], where the focus is on a variant called the weighted MAX-SAT in which each clause is assigned a positive weight and the objective of this problem is to maximize the sum of weights of satisfied clauses by any assignment.

2 Meta-Heuristics

2.1 GSAT-RW

The choice behind GSAT [13] algorithm has been motivated by the fact that it represents the basic architecture for most stochastic local search algorithms for MAX-SAT. The introduction of an element of randomness (i.e, noise) into local search methods is common practice for improving effectiveness through diversification. In this spirit, the GSAT-RW algorithm, starts with a randomly chosen assignment. Thereafter, two possible strategies are used for selecting the variable to be flipped at each iteration of the algorithm. The first strategy is taking a *walk-step*, which amounts to randomly selecting a currently unsatisfied clause and then flipping one of its variables, also in a random manner. Thus, at each walk-step, at least one unsatisfied clause becomes satisfied. The other strategy uses a greedy search to choose a random variable from the set *PossFlips*, which contains the variables that when flipped (individually) achieve the largest decrease (or the least increase) in the total number of unsatisfied clause.

2.2 Memetic Algorithms (MAs)

Memetic Algorithms are stochastic global search heuristics in which Evolutionary Algorithms-based approaches are combined with search heuristics techniques. This hybridization is used in order to accelerate the discovery of high quality solutions, for which evolution alone would take too long to discover. The memetic algorithm used in this paper is implemented as follows:

- **Representation:**
  The chromosomes (individuals) which are assignments of values to the variables are encoded as strings of bits, the length of which is the number of variables (or clusters if MA is operating on a coarse level). The values ***True*** and ***False*** are represented by 1 and 0 respectively. In this representation , an individual X corresponds to a truth assignment and the search space is the set $S = \{0, 1\}^n$.
- **Fitness function:**
  The notion of fitness is fundamental to the application of memetic algorithms. It is a numerical value that expresses the performance of an individual (solution) so that different individuals can be compared. The fitness of a chromosome (individual) is equal to the number of clauses that are unsatisfied by the truth assignment represented by the chromosome.
- **Initial population:**
  A initial solution is generated using a population consisting of 50 individuals. According to our computational experience, larger populations do not bring effective improvements on the quality of the results. At the coarsest level, MA will randomly generate an initial population of 50 individuals in which each gene's allele is assigned the value 0 or 1.
- **Crossover:** The individuals are visited in random order. An unmatched individual i_k is matched randomly with an unmatched individual i_l. Thereafter, the two-point crossover operator is applied using a crossover probability to each matched pair of individuals. The two-point crossover selects two randomly points within a chromosome and then interchanges the two parent chromosomes between these points to generate two new offspring. Recombination can be defined as a process in which a set of configurations (solutions referred as parents) undergoes a transformation to create a set of configurations (referred as offspring). The creation of these descendants involves the location and combinations of features extracted from the parents. The reason behind choosing the two point crossover are the results presented in [18] where the difference between the different crossovers are not significant when the problem to be solved is hard. The work conducted in [16] shows that the two-point crossover is more effective when the problem at hand is difficult to solve.
- **Mutation:** Let $C = c_1, c_2, ...c_m$ be a chromosome represented by a binary chain where each of whose gene c_i is either 0 or 1. In our mutation operator, each gene c_i is mutated through flipping this gene's allele from 0 to 1 or from 1 to 0 if the probability test is passed.
- **Selection:** The selection operator acts on individuals in the current population. Based on each individual quality (fitness), it determines the next population. In the roulette method, the selection is stochastic and biased toward the best individuals. The first step is to calculate the cumulative fitness of the whole population through the sum of the fitness of all individuals. After that, the probability of selection is calculated for each individual as being $P_{Selection_i} = f_i / \sum_1^N fi$, where f_i is the fitness of individual i.
- **Local Search:** A fast and simple local search is used for one iteration during which it seeks for the variable-value assignment with the largest decrease or the smallest increase in the number of unsatisfied clauses. Random tie breaking strategy is used between variables with identical score.
- **Convergence Criteria:** As soon as the population tends to loose its diversity, premature convergence occurs and all individuals in the population tend to be identical with almost the same fitness value. The proposed memetic algorithm is assumed to reach convergence when no further improvement of the best solution (the fittest chromosome) has not been made during two consecutive generations.
- **Changing Neighborhood:** Having improved the assignment at the neighborhood N_{m+1}, the assignment must be projected onto its parent neighborhood N_m. The projection process is trivial; if a cluster $C_i \in N_{m+1}$ is assigned the value of true then the matched pair of clusters that it represents, C_j and $C_k \in N_m$ are also assigned the value true. The same process is used for VNS-GSAT-RW.

3 The Variable Neighborhood Structure

Variable neighborhood search (VNS for short) [6] aims at finding a tactical interplay between diversification and intensification to overcome local optimality using a combination of a local search with systematic changes of neighborhood. Diversification refers to the ability to explore many different regions of the search space, whereas intensification refers to the ability to obtain high quality solutions within those regions. The proposed variable neighborhood structure works as follows:

- **Phase 1**: Let P denotes the set of literals of the problem to be solved. The first phase of the algorithm consists in constructing a set of neighborhood satisfying the following property: $N_1(x) \subset N_2(x) \subset .....N_{k_{max}}(x)$. The starting (default) neighborhood with $k = 0$ consists of a move based on the flip of a single variable. A flip means assigning the opposite state to a variable (i.e change True $\rightarrow$ False or False $\rightarrow$ True). The first neighborhood N_1 is constructed from P_0 by merging literals. The merging procedure is computed using a randomized algorithm. The literals are visited in a random order. If a literal l_i has not been matched yet, then a randomly unmatched literal l_j is selected, and a new literal l_k (a cluster) consisting of the two literals l_i and l_j is created. The set N_1 consists of the move based on flipping predefined clusters each having 2^1 variables. The new formed clusters are used to define a new and larger neighborhood N_2 and recursively iterate the process until the desired number of neighborhood (k_{max}) is reached. Thereafter a random solution is generated from the largest neighborhood ($N_{k_{max}}$) . The random solution consists in assigning a random state (True or False) to each cluster and all the literals within that cluster will get the same state.
- **Phase 2:** The second phase which is the most crucial aims at selecting the different neighborhoods according to some strategy for the effectiveness of the search process. The strategy adopted in this work is to let VNS start the search process from the largest neighborhood $N_{k_{max}}$ and continues to move towards smaller neighborhood structures. The motivation behind this strategy is that the order in which the neighborhood structures have been selected offers a better mechanism for performing diversification and intensification. The largest neighborhood N_{max} allows WS to view any cluster of literals as a single entity leading the search to become guided in far away regions of the solution space and restricted to only those configurations in the solution space in which the literals grouped within a cluster are assigned the same value. As the switch from one neighborhood to another implies a decrease in the size of the neighborhood, the search is intensified around solutions from previous neighborhoods in order to reach better ones. Once the search has reached the convergence criterion with respect to a neighborhood N_i, the assignment reached on that neighborhood must be projected on its parent neighborhood N_{i-1}. The projection algorithm is simple; if a cluster $c_i \in \mathrm{N}_m$ is assigned the value of true then the merged pair of clusters that it represents, c_j, $c_k \in \mathrm{N}_{m-1}$ are also assigned the true value.

4 Experimental Results

- GSAT-RW
  To illustrate the potential gains of the proposed variable neighborhood structure when combined with GSAT-RW, a selected benchmark suite covering different domains (Random-3-SAT, SAT-encoded graph coloring problems, SAT-encoded logistics problems, SAT-encoded block world planning problems, SAT-encoded quasi-group Problems) These instances are by no means intended to be exhaustive but rather give an indication of typical performance behavior. All these benchmark instances are known to be hard and difficult to solve and are available from the SATLIB website (http://www.informatik.tu-darmstadt.de/AI/SATLIB). All the instances are satisfiable instances and have been used widely in the literature in order to give an overall picture of the performance of different algorithms. Due to the randomization of GSAT-RW each problem instance is run 100 times with a cutoff parameter (maximum allowed solving time is set at 300 seconds). GSAT is assumed to reach convergence with respect to a given neighborhood if the best assignment has been improved for 50 consecutive iterations. The cardinality of the neighborhood (k_{max}) is set such that the

number of the formed clusters is 10% of the size of the problem instance (i.e, a problem with 100 variables will lead to k_{max} equals to 4 (N_0 , N_1 , N_2 , N_3)).

- MA
  A set of large problem instances taken from real industrial bounded model checking hardware designs is used to test the performance of VNS-MA.. This set is taken from (http://www.informatik.tu-darmstadt.de/AI/SATLIB). All the benchmark instances used in this experiment are satisfiable instances. The tests were carried out on a DELL machine with 800 MHz CPU and 2 GB of memory. The code was written in C and compiled with the GNU C compiler version 4.6. The parameters used in the experiment are listed below:
  * Crossover probability = 1
  * Mutation probability = 0.1.
  * Population size = 50 .
  * MA is assumed to have reached convergence and switch to a smaller neighborhood if the fitness of the fittest chromosome remains unchanged during five consecutive generations.
  * The cardinality of the neighborhood is the same as the one chosen for GSAT-RW.

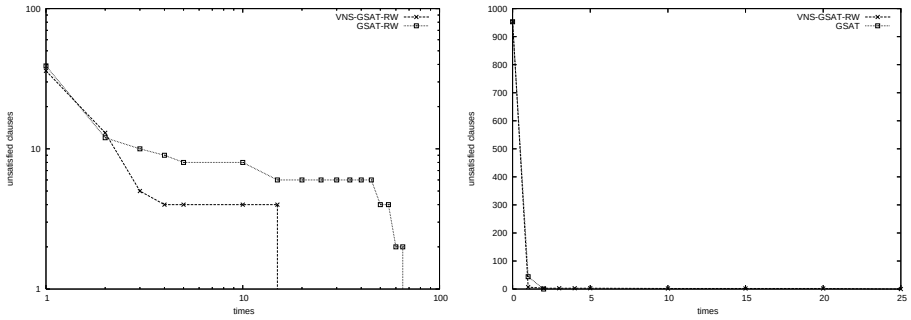


Fig. 1. Evolution of the best solution on a 300 variable problem with 1117 clauses (logarithmic scale) (flat100.cnf). .Evolution of the best solution on a 116 variable problem with 953 clauses (block-medium). Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses.

Overall, at least for the instances tested here, we observe that the search pattern happens in two phases. In the first phase that corresponds to the early part of the search, both GSAT-RW and VNS-GSAT-RW behave as a hill-climbing method. This phase can be described as a fast-working one, with a large number of the clauses being satisfied. The best assignment

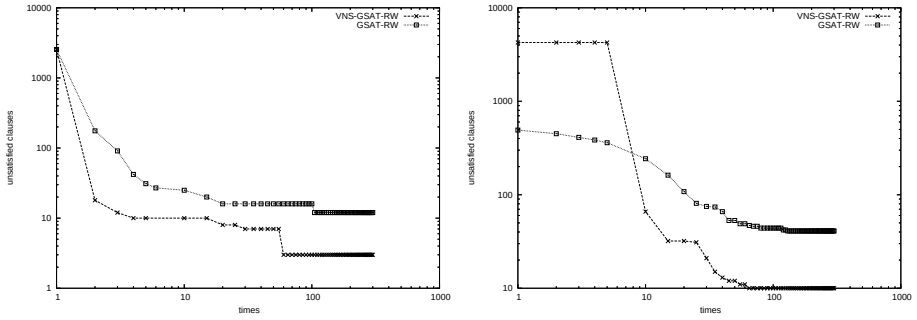


Fig. 2. Left: Evolution of the best solution on a 600 variable problem with 2550 clauses (logarithmic scale)(f600.cnf). Right: Evolution of the best solution on a 1000 variable problem with 4250 clauses (f1000.cnf)(logarithmic scale). Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses

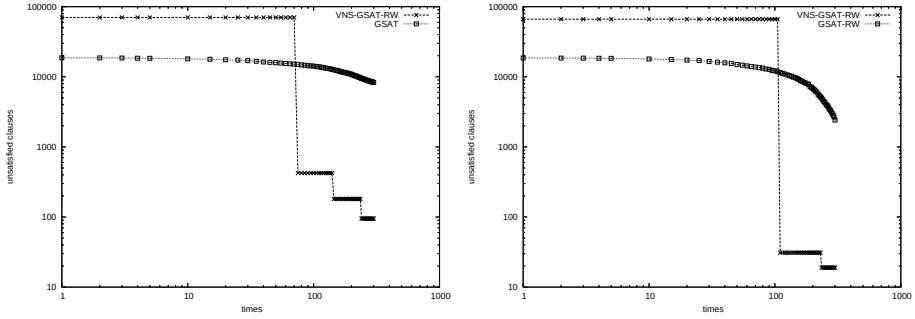


Fig. 3. Left: Evolution of the best solution on a 2000 variable problem with 8500 clauses (logarithmic scale)(f2000.cnf). Right: Evolution of the best solution on a 2125 problem with 66272 clauses (g125-17.cnf)(logarithmic scale). Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses.

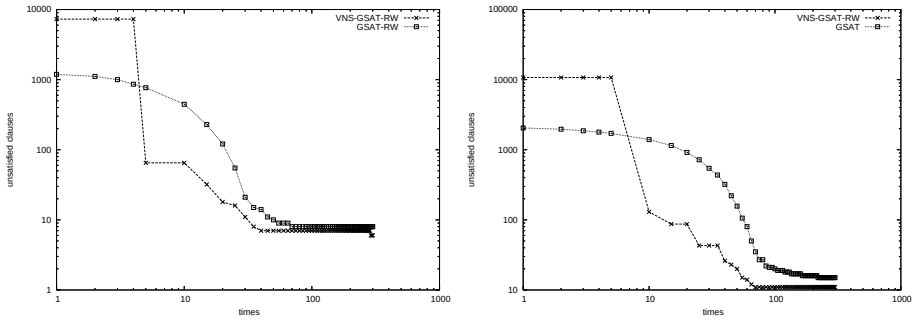


Fig. 4. Left: Evolution of the best solution on a 843 variable problem with 7301 clauses (logisticsb.cnf) (logarithmic scale). Right: Evolution of the best solution on a 4713 variable problem with 21991 clauses (logarithmic scale)(logisticsc.cnf). Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses.

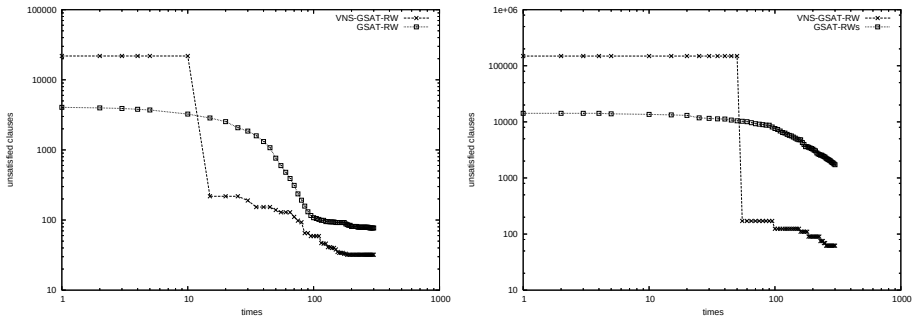


Fig. 5. SAT-encoded quasigroup: Left: Evolution of the best solution on a 129 variable problem with 21844 clauses (logarithmic scale)(qg6-9.cnf). Right: Evolution of the best solution on a 512 variable problem with 148957 clauses (logarithmic scale)(qg1-8.cnf). Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses.

climbs rapidly at first, and then flattens off as we mount the plateau, marking the start of the second phase. The plateau spans a region in the search space where flips typically leave the best assignment unchanged. The long plateau becomes even more pronounced as the number of flips increases, and occurs more specifically in trying to satisfy the last few remaining clauses. The transition to the plateau corresponds to a change to the region where a small number of flips gradually improves the score of the current solution ending with an improvement of the best assignment. The plateau is rather short with VNS-GSAT-RW compared with that of GSAT. The projected solution from one larger neighborhood to another finer one offers an elegant mechanism to reduce the length of the plateau as it consists of more degrees of freedom that can be used for further improving the best solution. Comparing GSAT-RW with VNS-GSAT-RW for small sized problems (up to 1500 clauses), both algorithms seem to be reaching the optimal quality solutions. It is not immediately clear which of the two algorithms converges more rapidly. This is probably highly dependent on the choice of the instances in the test suite. For example, the run time required by VNS-GSAT-RW for solving instance flat100-239 is more than 12 times higher than the mean run-time of GSAT-RW (25 sec vs 2 sec). The situation is reversed when solving the instance block-medium (20 sec vs 70 sec). The difference in convergence behavior of both algorithms start to become more distinctive as the size of the problem increases. All the plots show a clear dominance of VNS-GSAT-RW over GSAT-RW throughout the whole run. VNS-GSAT-RW shows a better asymptotic convergence (to around $0.008\% - 0.1\%$) in excess of the optimal solution as compared with GSAT-RW which only reach around $(0.01\% - 11\%)$. Figures 6-10 compares MA against VNS-MA and shows how the best assignment (fittest chromosome) progresses during the search. The performance of MA is unsatisfactory and is getting even far more dramatic for larger problems as the percentage excess over the solution is higher compared to that of VNS-MA. The curves show no cross-over implying that VNS-MA dominates MA. The asymptotic performance offered by VNS-VMA is impressive. In some cases, the difference in performance reaches 30% during the first seconds, and maintains it during the whole search process. However, on other cases, the difference in performance continues to increase as the search progresses. The differences in quality is within 66% for problems less than 150.000 clauses. For larger problems, the difference can get as high as 77%. This is perhaps not a surprise since there is a much greater potential for large problems, and hence the variable neighborhood structure is more likely to be of assistance. The larger the problem, the more neighborhoods is needed, and consequently the more refinement at different neighborhoods takes place.

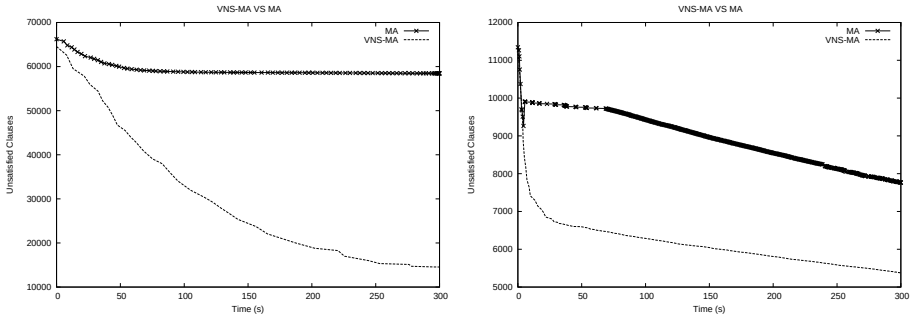


Fig. 6. bmc-galileo9.cnf: $|V| = 63624$,$|C| = 326999$. Along the horizontal axis we give the time (in seconds) , and along the vertical axis the number of unsatisfied clauses. bmc-ibm-1.cnf: $|V| = 9685$, $|C| = 55870$

5 Conclusions

In this work, two hybrid approaches combining VNS with MA and GSAT-RW have been described. VNS follows a simple principle that is based on systematic changes of neighborhood within the search. The set of neighborhood proposed in this paper can easily be incorporated

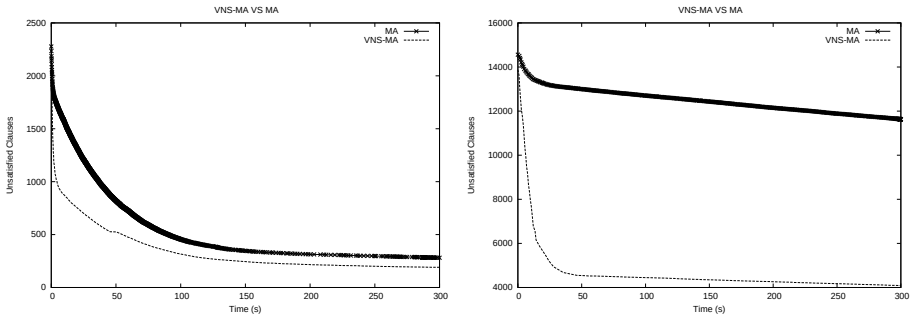


Fig. 7. bmc-ibm-2.cnf: $|V| = 3628$, $|C| = 14468$. Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses. bmc-ibm-3.cnf: $|V| = 14930$, $|C| = 72106$.

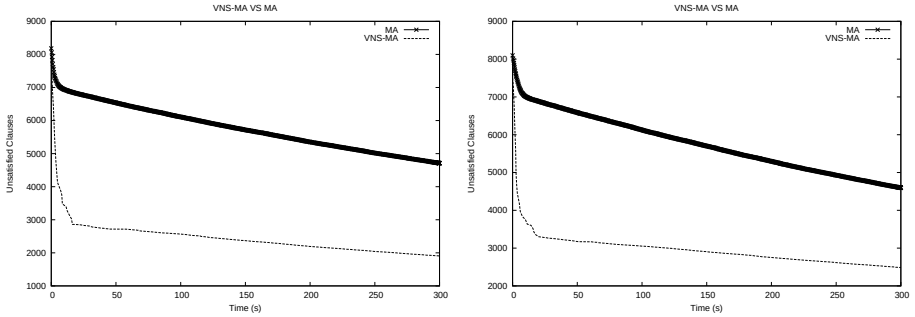


Fig. 8. bmc-ibm-5: $|V| = 9396$, $|C| = 41207$. Along the horizontal axis we give the time in seconds , and along the vertical axis the number of unsatisfied clauses. bmc-ibm-7.cnf: $|V| = 8710$, $|C| = 39774$.

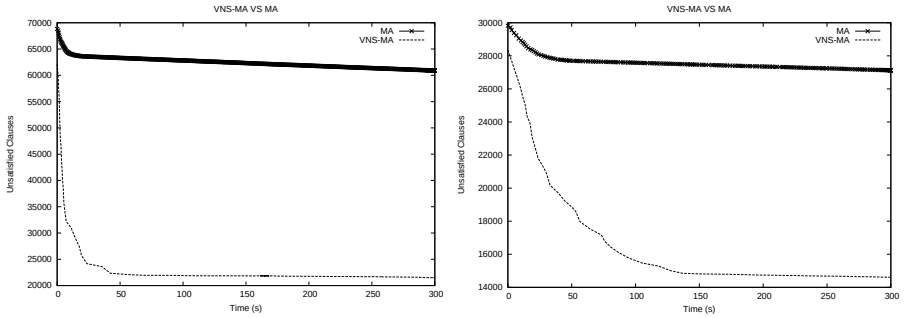


Fig. 9. bmc-ibm-10.cnf: $|V| = 61088$, $|C| = 334861$. Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses.

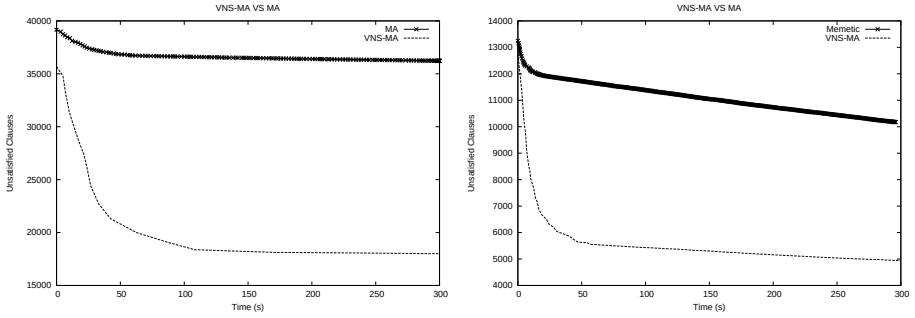


Fig. 10. bmc-ibm-12.cnf: $|V| = 39598$, $|C| = 19477$. Along the horizontal axis we give the time in seconds, and along the vertical axis the number of unsatisfied clauses. bmc-ibm-13.cnf: $|V| = 13215$, $|C| = 6572$

into any meta- heuristic when dealing with various combinatorial optimization problems. Starting the search from the largest neighborhood and moving systematically towards the smallest neighborhood is a better strategy for performing diversification and intensification. The key success behind the efficiency of VNS-MA and VNS-GSAT-RW relies on the neighborhood structure used. VNS-MA draws its strength from coupling the optimization process across different neighborhoods. By allowing MA and GSAT-RW to view a cluster of variables as a single entity, the search becomes guided and restricted to only those configurations in the solution space in which the variables grouped within a cluster are assigned the same value. As the size of the clusters varies from one neighborhood to another, the size of the neighborhood becomes adaptive and allows the possibility of exploring different regions in the search space while intensifying the search by exploiting the solutions from previous neighborhoods in order to reach better solutions.

References

1. Boughaci,D., and Drias,H.: Efficient and experimental meta-heuristics for MAX- SAT problems. In Lecture Notes in Computer Sciences, WEA 2005,3503/2005:501-512, (2005).
2. Cai,S., and Su,K.: Configuration checking with aspiration in local search for SAT. In: Proc. of AAAI-12. pages , 434-440, (2012).
3. Cook,S.A.: The complexity of theorem-proving procedures. Proceedings of the Third ACM Symposium on Theory of Computing, pages 151-158, (1997)
4. Granmo,O.C. and Bouhmala, N.: Solving the Satisfiability Problem Using Finite Learn- ing Automata. International Journal of Computer Science and Applications, 4(3): pages 15–29, (2007)
5. Guo,P., and Chenh, W.: A general variable neighborhood search For single-machine total tardiness shedulingf problem with step-deteriorating jobs. Jounal of Indusrial and management Optimization, Volume 10, Number 4, (2014).
6. Hansen,P., Jaumard,B., Mladenovic,N., and Parreira, A.D.: Variable neighborhood search for maximum weighted satisfiability problem. Technical Report G-2000-62, Les Cahiers du GERAD, Group for Research in Decision Analysis,(2000).
7. Jin-Kao, H., Lardeux,H., and Saubion, F.: Evolutionary computing for the satisfiability problem. In Applications of Evolutionary Computing, volume 2611 of LNCS, pages 258-267, University of Essex, England, (2003).
8. Li,C.M., Wei,W., Zhang,H.: Combining adaptive noise and look-ahead in local search for SAT, Lecture Notes in Computer Science 4501 , pages 121-131, (2007).
9. Li, C.M., and Huang, Q.W.: Diversification and determinism in local search for satisfiability. Proceedings of the Eighth International Conference on Theory and Applications of Satisfiability Testing (SAT-05), volume 3569 of Lecture Notes in Computer Science, pages 158-172, (2005).
10. Mazure, B., *Saïs*, L., and *Grégoire*, E.: Tabu search for SAT. Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97), pages 281-285, (1997).
11. McAllester, D., Selman,B., and Kautz, H.: Evidence for invariants in local search. proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97), pages 321-326, (1997).
12. Mucherino, A., and Liberti, L.: A VNS-Based Heuristic for Feature Selection in Data Mining. Hybrid Metaheuristics, studies in computational intelligence , volume 434, pages 353-368,(2013).
13. Selman, B., Levesque,H., and Mitchell, D.: A New Method for Solving Hard Satisfiability Problems. Proceedings of AAA92, pages 440-446, MIT Press, (1992).
14. Selman,B., Kautz,H.A., and Cohen,B.: Noise Strategies for Improving Local Search. Proceedings of AAAI'94, pages 337-343. MIT Press, (1994).
15. Smyth,K. Hoos,H. and *Stü*,T.: Iterated Robust Tabu Search for MAX-SAT. Advances in Artificial Intelligence, Lecture Notes in Computer Science, Volume 2671, pages 129-144, (2003).
16. Spears, W.: Adapting Crossover in Evolutionary Algorithms. Proc of the Fourth Annual Conference on Evolutionary Programming, MIT Press, pages 367-384, (1995).
17. Thornton,J., Pham, D.N., Bain, S., and Ferreira, V.: Additive versus multiplicative clause weighting for SAT. Proceedings of the AAA-I 2004, pages 191-196, (2004).
18. Vrajitoru, D.: Genetic programming operators applied to genetic algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 686-693, Orlando (FL). Morgan Kaufmann Publishers, (1992).
19. Xu,Y., Wang,L., and Yang,Y.: Dynamic Vehicle Routing Using an Improved Variable Neighborhood Search Algorithm. Journal of Applied Mathematics Volume 2013, Article ID 672078, Hindawi Publishing Corporation, (2013).
20. Yagiura,M., and Ibaraki,T.: Efficient 2 and 3-Flip Neighborhood Search Algorithms for the MAX SAT: Experimental Evaluation. Journal of Heuristics, 7: pages 423-442, (2001).

A multi-objective approach for demand side management in smart grids

Zineb Garroussi[1,2] (✉), Rachid Ellaia[1,2], and El-Ghazali Talbi[2,3]

[1]Laboratory of Study and Research in Applied Mathematics, Mohammadia School of Engineering.
BP. 765, Ibn Sina Avenue, Mohammed V University, Rabat, Morocco
[2]Inria Lille - Nord Europe, DOLPHIN Project-team, 59650 Villeneuve dAscq, France.
[3]Université Lille 1, LIFL, UMR CNRS 8022, 59655 Villeneuve dAscq cedex, France
zinebgarroussi@emi.ac.ma

Abstract. This paper deals with the scheduling of the household appliances over a one-day time horizon subdivided to 96 time slots, each time slot is equal to 15 minutes. Given the load profile and the time windows of the time shiftable appliances, the outdoor temperature, the maximum, the minimum, the dead-band limits and the desired indoor temperature, the parameters of the electric heater and the time-varying electricity price, our aim is to minimize the total cost of electricity usage and the total discomfort of all houses while fulfilling the time shiftable and thermal appliances' constraints. The discomfort is divided into two parts: timing discomfort and thermal discomfort, the former part is modeled by lowering the delay time in the use of time shiftable appliances, the latter part a penalty is attributed to deviations from the desired thermal state. In order to avoid the creation of new power peaks caused by the load shifting to the least price periods, the standard deviation from the ideal load curve (the average load of all houses) is also minimized simultaneously with the previous objectives. We propose a multi-objective evolutionary algorithm to tackle the optimization model in a reasonable computing time. The simulation results from different case studies are presented and show the effectiveness of the proposed algorithm in reducing the total cost, ensuring a comfort level for all houses, as well as preventing the creation of new peaks.

Keywords Time-varying pricing, household, appliance, scheduling, evolutionary algorithm.

1 Introduction

In the wholesale electricity market, the cost of the electricity supply changes substantially depending on the season and time of day. However, in the retail market, consumers usually paid their electricity consumption based on static prices, which means it costs the same amount regardless of when it is used [7]. For this reason, the electricity demand remains relatively unresponsive to the wholesale prices, requiring the massive reinforcement of extra production capacities and distribution networks. The emerging of smart grid technology is expected to help the implementation of electricity time-varying pricing where demand is sensitive to the prices. Through an automated energy management system (EMS) it will be effectively and accurately able to automate the consumers' electricity use in response to the grid and weather conditions, the desired thermal comfort level, and local energy generation. Indeed the primary objective of an automated energy management system is helping to optimally schedule the electricity used during on-peak periods through some demand response techniques, including peak shaving, flexible loads shifting, and valley filling [3]. These last years, the deployment of demand response benefits in homes has become more challenging with respect to the diversity of household electric appliances, the multiobjective conflicting nature of the residential consumer's targets, and the incoordination of energy demand between homes. Hence, based on the two way flow of information and power, and the electricity time-varying pricing, several scheduling optimization algorithm is proposed in [1,4,6,9]. [4] tackles the anticipation layer of a home automation system. The household energy allocation is controlled by taking into account predicted events. The problem is formulated as a constraint satisfaction problem. Two objectives are considered: the energy cost and the user comfort. The thermal comfort criterion was defined by the threshold and treated as a constraint. Due to the NP-Hard complexity of the problem a tabu search (TS) is applied, minimizing a penalty function of constraint violation

in a first phase and the energy cost in the second phase once a feasible solution was found. However, in spite of the relatively high computation time to schedule only two electricity consumption tasks and two heating systems, the results show also that the TS algorithm settings are problem dependent, and different strategies are proposed to deal with all situations. In [6] the load scheduling using a multiobjective genetic approach were performed by the authors and promising results were obtained. Two objectives are considered the electricity bill and the end-user's dissatisfaction. In [1] a non linear formulation of a simple electrical load management in smart homes where appliances have non linear time varying power consumption, and solved by a customized evolutionary algorithm combined with a local search technique. In [9] a cooperative particle swarm optimization is used to solve appliances scheduling problem for a set of smart homes. The authors classify the appliances into time shiftable devices and power shiftable devices. The simulation results indicate that the coordination between homes leads to reduce the electricity cost and avoid peak rebounds. In this paper, we propose a multiobjective evolutionary algorithm (EA) to schedule the controllable appliances in multi-home context. The objective is to reduce the electricity cost, discomfort while avoiding peak rebounds. The discomfort is divided into two parts; timing discomfort and thermal discomfort, the former part is modeled by lowering the delay time in the use of time shiftable appliances due to the load shifting, the latter part we attribute a penalty to deviations from the desired thermal state. The algorithm schedules the household appliances, ensures a comfort level, as well as flattens the total aggregated load curve of all houses. The remainder of this paper is structured as follows. In section II, the mathematical model is formulated. Section III introduces the multiobjective EA approach used in this study. In section IV, simulations and results are given. Section V concludes the paper.

2 Mathematical Model

This section presents the modeling of the household appliances scheduling problem and lists the household appliances' constraints. In this model we consider two kinds of household appliances, time shiftable appliances (TSA) and thermal appliances (TA).

2.1 Objectives

The objectives of the household appliances scheduling problem is the minimization of the total cost F_{cost} in (1), the total discomfort $F_{discomfort}$ in (2), and the standard deviation of the electricity consumption F_{std} in (3) of all houses over the entire scheduling period. Mathematically, the scheduling problem considered as a multiobjective optimization problem and is formulated as follows:

$$F_{cost} = \min \sum_{h=1}^{H} \sum_{t=1}^{T} \sum_{a \in \mathcal{A} \cup \mathcal{C}} P_{h,t}^{a} \times c_t \tag{1}$$

$$F_{discomfort} = \min \frac{1}{H} \sum_{h=1}^{H} \sum_{a \in \mathcal{A}} \frac{U_{TSA}^{h,a}}{a} + \frac{1}{H} \sum_{h=1}^{H} \sum_{c \in \mathcal{C}} \frac{U_{TA}^{h,c}}{c} \tag{2}$$

$$F_{std} = \min \sqrt{\frac{\sum_{t=1}^{T} (\sum_{a \in \mathcal{A} \cup \mathcal{C}} P_{h,t}^{a} - Ideal)^2}{T}} \tag{3}$$

Where $Ideal$ is the average load for all household appliances considered and is calculated as follows:

$$Ideal = \frac{\sum_{h=1}^{H} \sum_{t=1}^{T} \sum_{a \in \mathcal{A} \cup \mathcal{C}} P_{h,t}^{a}}{T} \tag{4}$$

Here, t is the time slot indice, h is the house indice, T is the number of time slots representing the scheduling horizon, H is the number of houses, c_t is the electricity price of the grid at time slot t, a is the appliance indice, $\mathcal{A}$ set of time shiftable appliances, $\mathcal{C}$ set of thermal, $P_{h,t}^{a}$ is the power consumed by the appliance a at time slot t in the $h-th$ house, $U_{TSA}^{h,a}$ is the delay time of the $a-th$ TSA of the $h-th$ house, $U_{TA}^{h,c}$ is the discomfort level of $c-th$ TA of the $h-th$ house.

2.2 Time shiftable appliances (TSA)

Let the $ST^{h,a}_{current}$, $ST^{h,a}_{min}$, $PS^{h,a}$, $ET^{h,a}_{max}$, and $D^{h,a}_{work}$ the current starting time, the minimum starting time, the preferred starting time, the maximum ending time, and the processing time duration of the $a-th$ TSA of the $h-th$ house respectively ($ST^{h,a}_{min} < PS^{h,a} < ET^{h,a}_{max} - D^{h,a}_{work}$). The processing time $D^{h,a}_{work}$ is the number of time slots for the operation of the TSA appliance. ($ET^{h,a}_{max} - ST^{h,a}_{min} >= D^{h,a}_{work}$). The delay time of TSA is shown in equation. 5. Figure 1 illustrates the relationships between the TSA comfort level parameters. The later the time shiftable appliance operates, the larger delay time becomes. The greater $ET^{h,a}_{max} - ST^{h,a}_{min}$ is, the more possible solutions there are.

$$U^{h,a}_{TSA} = \begin{cases} 0 & \text{if } ST^{h,a}_{min} \leq ST^{h,a}_{current} \leq PS^{h,a} + D^{h,a}_{work} \\ \frac{ST^{h,a}_{current} - PS^{h,a}}{ET^{h,a}_{max} - D^{h,a}_{work} - PS^{h,a}} \times 100, & \text{if } PS^{h,a} + D^{h,a}_{work} < ST^{a}_{h,current} \leq ET^{h,a}_{max} - D^{h,a}_{work} \end{cases} \tag{5}$$

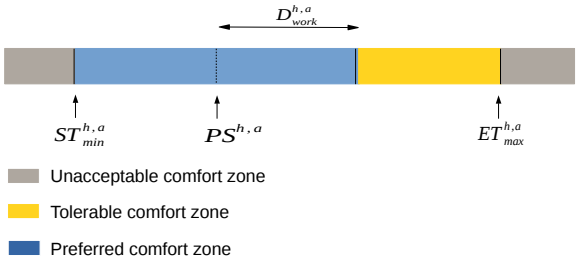


Fig. 1: Illustration of time shiftable appliances comfort level parameters

2.3 Thermal appliances (TA)

According to the temperature model given in [2], the indoor temperature at every time slot t can be expressed by the following equation. 6:

$$T^{in}_{h,t+1} = \epsilon T^{in}_{h,t} + (1-\epsilon)(T^{out}_{h,t} \pm \frac{COP \cdot p^{hvac}_{h,t}}{A}) \tag{6}$$

Where ϵ is the inertia factor, A is the thermal conductivity (kW/C) and COP is the coefficient of performance. $T^{out}_{h,t}$, $T^{in}_{h,t}$ is the outdoor temperature at time slot t of the household h. The discomfort level of the HVAC is expressed as follows :

$$U^{h,HVAC}_{TA} = \frac{100}{T} \sum_{t}^{T} \frac{d^{in}_{h,t}}{\Delta T^{in}_{h,max}} \tag{7}$$

Where $\Delta T^{in}_{h,max} = max(T^{des}_{h,t} - T^{in}_{h,min}, T^{in}_{h,max} - T^{des}_{h,t})$ and $d^{in}_{h,t}$ is calculated as follows :

$$d^{in}_{h,t} = \begin{cases} T^{des}_{h,t} - T^{in}_{h,t} & \text{if } T^{in}_{h,min} \leq T^{in}_{h,t} < T^{des}_{h,t} - \Delta T^{in}_{L} \\ 0 & \text{if } T^{des}_{h,t} - \Delta T^{in}_{h,L} \leq T^{in}_{h,t} < T^{des}_{h,t} + \Delta T^{in}_{h,U} \\ T^{in}_{h,t} - T^{in}_{h,t} & \text{if } T^{des}_{h,t} + \Delta T^{in}_{U,h} < T^{in}_{h,t} \leq T^{in}_{h,max} \\ \Delta T^{in}_{h,max}, & \text{otherwise} \end{cases} \tag{8}$$

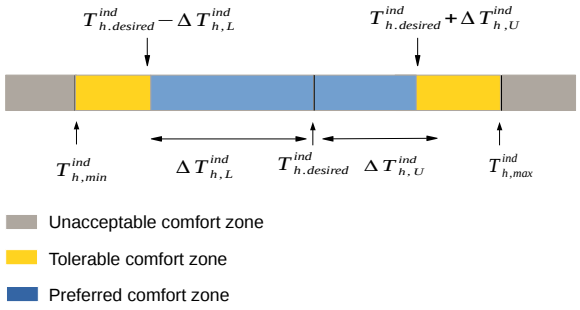


Fig. 2: Illustration of HVAC comfort level

Equation 7 ensures that the indoor temperature of house h needs to be felt in in the acceptable comfort range, where $T^{in}_{h,min}$, $T^{in}_{h,max}$, $T^{des}_{h,t}$, $\Delta T^{in}_{h,L}$ and $\Delta T^{in}_{U,h}$ are respectively the minimal, maximal, the desired temperature, the lower and upper dead band limits the user can set without a significant effect on his/her thermal comfort. Figure 2 illustrates the relationships between the HVAC comfort level parameters.

3 Multiobjective Evolutionary Algorithm

In this section we describe the proposed algorithm to deal with the household appliance scheduling problem using the pareto approach. The main major steps of the multiobjective EA algorithm are stated as follows:

3.1 Individual representation and initialization

In the decision space, the representation of solutions determines the efficiency and effectiveness of the algorithm, the representation vector X is a vector with a dimension of $H \times |a| + T$, it's used as the chromosome representation of the individual as shown as follows in equation. 9:

$$X = [ST^{current}, P^{HVAC}] \tag{9}$$

where

$$ST^{current} = [ST^{current}_{h}, ST^{current}_{h+1}, ..., ST^{current}_{H}]$$

$$ST^{current}_{h} = [ST^{current}_{h,a}, ST^{current}_{h,a+1}, ..., ST^{current}_{h,|a|}]$$

$$P^{HVAC} = [P^{HVAC}_{h}, P^{HVAC}_{h+1}, ..., P^{HVAC}_{H}]$$

$$P^{HVAC}_{h} = [P^{HVAC}_{h,1}, P^{HVAC}_{h,2}, ..., P^{HVAC}_{h,T}]$$

$ST^{current}_{h,a}$ is an integer variable representing the starting time of TSA a of the h household, and is initialized with a random integer value between $ST^{min}_{h,a}$, and $ET^{max}_{h,a} - D^{work}_{h,a}$, and $P^{HVAC}_{h,t}$ is a real variable representing the power required by the HVAC of household h at time slot t, and is initialized with a random real value between 0.0 and $P^{HVAC}_{h,max}$ where $P^{HVAC}_{h,max}$ is the maximum power of the HVAC appliance.

3.2 Selection, crossover and mutation

Here, a deterministic tournament is used between two randomly chosen individuals. A random swap is implemented as a crossover operator. It picks a random position in the starting time and HVAC power decision vectors respectively, and exchanges their values. This crossover operator guarantees the producing feasible offsprings. The mutation provides diversity in the population and is carried in the algorithm by changing a certain fraction of starting time and power of HVAC respectively. The new generated starting time and power of HVAC will be generated randomly between $[ST_{h,a}^{min} - ET_{h,a}^{max} - D_{h,a}^{work}]$ and $[0.0, P_{h,max}^{HVAC}]$ respectively.

4 Simulations and Results

4.1 Data

The 24 hour day-time is divided into $T = 96$ equal time slots, each time slot $t \in \{1, ..., T\}$ is 15 minutes. We assume to consider the time-varying electricity prices as shown in Fig. 3. The outdoor temperature curve is given in Fig. 4. It is supposed that the number of households is $H = 5$. The power load profiles of EDW, ECW, and ECD are respectively $[1.2, 1.2, 0.2, 1.1, 0.68, 0.8, 0.6]$, $[0.52, 0.65, 0.52]$, $[2.95, 2.91, 2.90, 0.19]$, and are taken from the paper [5]. The parameter settings of TSA are listed in Table. 1. The HVAC settings data are taken from [2] ($P_{h,max}^{HVAC} = 3.5kW, COP = 2.5, \epsilon = 0.98, A = 0.45kW/^{\circ}C$), $T_{h,min}^{in}$, T_{h}^{des}, $T_{h,max}^{in}$, $\Delta T_{h,L}^{in}$ and $\Delta T_{U,h}^{in}$ are $15^{\circ}C, 20^{\circ}C, 24^{\circ}C$, 2, and 3 respectively. The parameter settings of the EA algorithm are as follows. The population size is 100, the number of iterations is 100. The probability of the crossover is 0.25. The probability of the mutation is 0.35. All the simulations are carried out with ParadisEO 2.0.1 framework [8], and executed on Intel Core i3 380M 2.53 GHz personal computer with 4.0 gigabyte of RAM. Figures are displayed using Matlab.

Table 1: Parameter settings of TSA

Time slot parameter / Time shiftalbe appliance a	$ST_{min}^{h,a}$	$ET_{max}^{h,a}$	$D_{work}^{h,a}$	$PS^{h,a}$
ECW	20	96	3	24
ECD	20	96	4	24
EDW	20	96	7	24

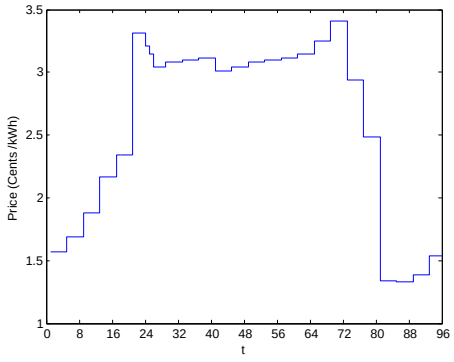


Fig. 3: Electricity prices

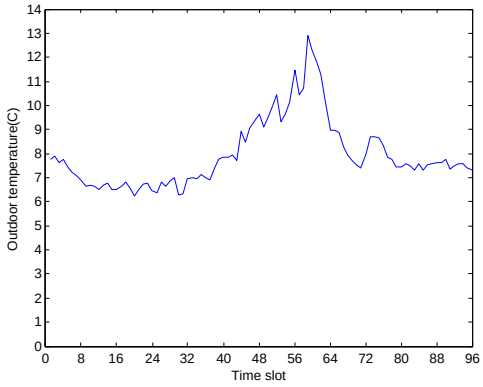
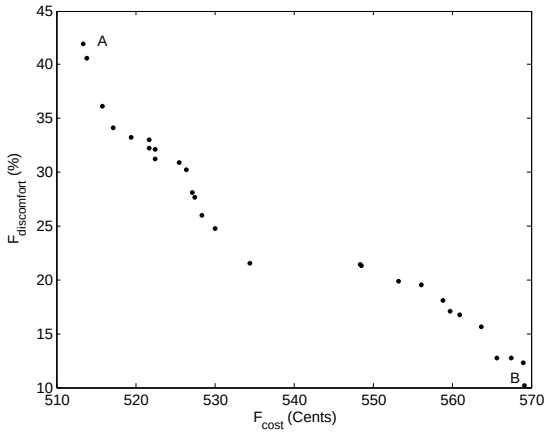


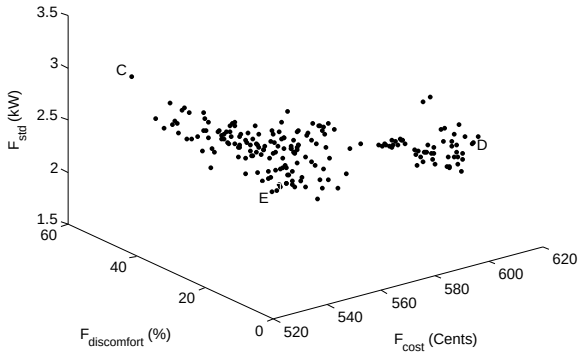
Fig. 4: Outdoor temperature

4.2 Results

Figure. 5 displays the pareto optimal front in the two (F_{cost} and $F_{discomfort}$) and (F_{cost}, $F_{discomfort}$ and F_{std}) objective spaces, which are noted later by Case I and Case II respectively. The extreme solutions (A) and (B) for case I, (C), (D), and (E) for case II are displayed in figures. 6, 7, 8, 9, and 10 respectively. The solution includes the power transmitted from the grid to the TSA, the total power demand, and the mean indoor temperature of all houses at each time slot t over a period of one day. Table (2) lists the simulation results for Case I and II with a reasonable running time equal to 6.33 and 9.12 respectively. As we can see, the solution (A) is the most favorable economically,however, the maximum power peak was at its highest level in time slot compared with other solutions and the the mean indoor temperature is below the dead band lower limit for most of the time slots. Solutions (B) and (D) are the most favorable for the point of timing and thermal comfort. The mean indoor temperature is between the dead band lower limit and the desired temperature for most of the time slots in order to guarantee the preferred thermal comfort. The scheduling of TSA current starting times is near to the preferred starting times which is equal to 24 for all TSA appliances. Solution (E) is the most favorable from the point of view of the grid with the lowest maximum peak and minimal standard deviation with respect to other solutions.

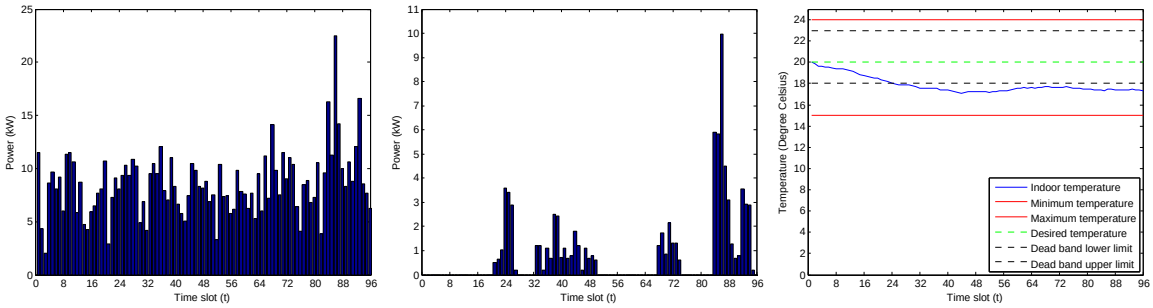


(a) Case I



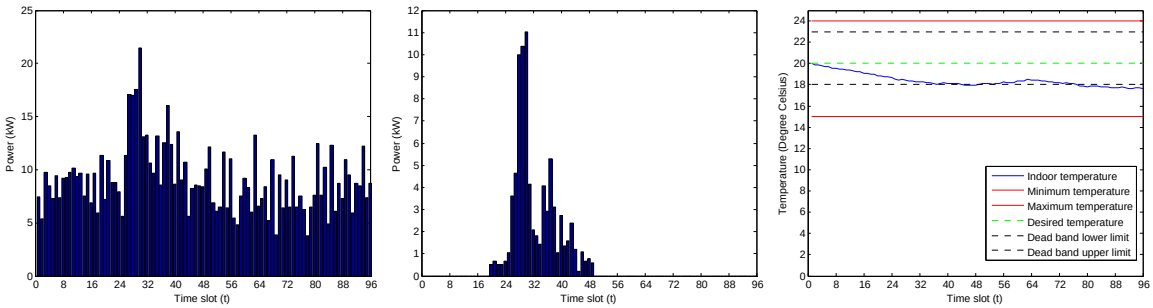
(b) Case II

Fig. 5: Pareto optimal front



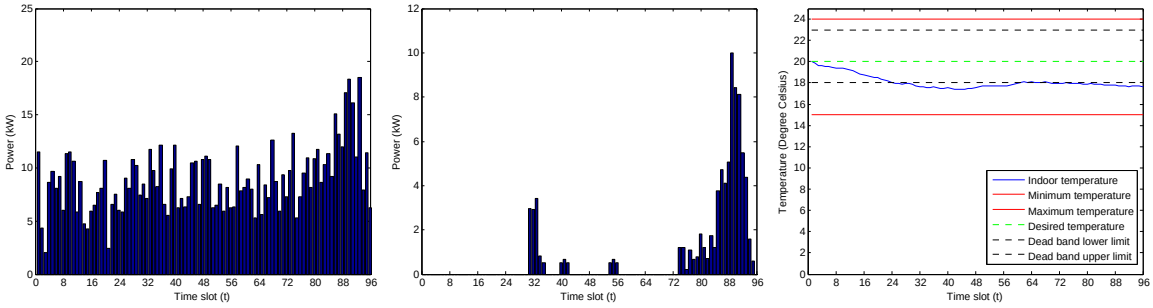
(a) Total power (b) TSA power (c) Indoor temperature

Fig. 6: Solution A



(a) Total power (b) TSA power (c) Indoor temperature

Fig. 7: Solution B



(a) Total power (b) TSA power (c) Indoor temperature

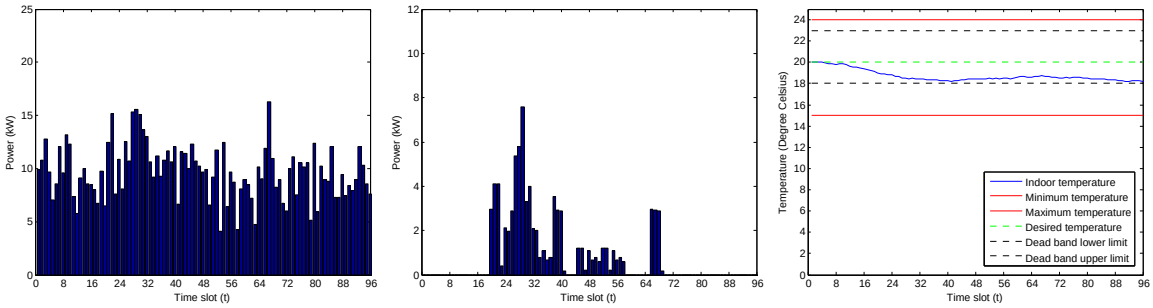
Fig. 8: Solution C

Table 2: Simulation results

Solutions	Cost (Cents)	Discomfort (%)	STD (kW)	Mean Power (kW)	Maximum peak (kW)	Runtime (s)
Case I (Two objectives)						
Solution A	513.412	41.9427	3.00948	8.5504	22.4679	6.33
Solution B	569.159	10.1546	3.07063	9.1471	21.4435	
Case II (Three objectives)						
Solution C	528.72	48.15560	3.036960	8.9053	18.5217	9.12
Solution D	603.545	7.321010	2.506480	9.7048	16.3013	
Solution E	561.982	31.7631	1.95812	9.0863	15.0263	

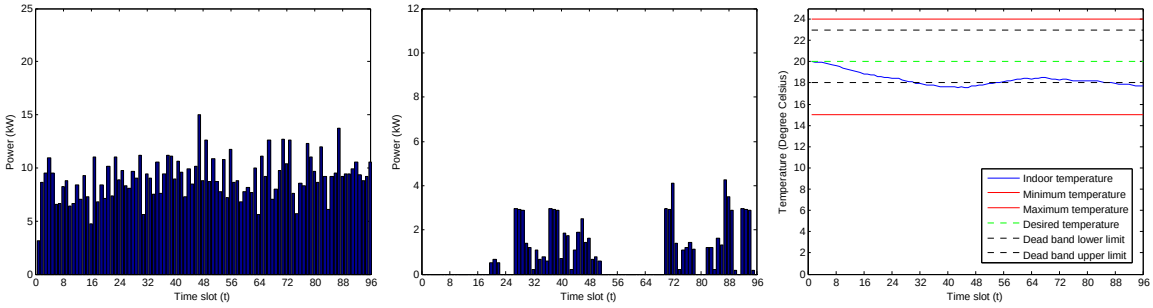
5 Conclusions

This paper presents a problem of a households appliance scheduling under the electricity time-varying pricing. We developed an evolutionary algorithm to schedule both the starting time of



(a) Total power (b) TSA power (c) Indoor temperature

Fig. 9: Solution D



(a) Total power (b) TSA power (c) Indoor temperature

Fig. 10: Solution E

time shiftable appliances and the power of thermal appliances. Future studies should investigate a stochastic programming model of the household appliance scheduling. We focus on the electricity price uncertainty which can expect to lead to significant changes of the solutions compared with the deterministic household appliance scheduling model given in this work.

References

1. Florian Allerding, Marc Premm, Pradyumn Kumar Shukla, and Hartmut Schmeck. Electrical Load Management in Smart Homes Using Evolutionary Algorithms. In Jin-Kao Hao and Martin Middendorf, editors, *Evolutionary Computation in Combinatorial Optimization*, number 7245 in Lecture Notes in Computer Science, pages 99–110. Springer Berlin Heidelberg, April 2012. DOI: 10.1007/978-3-642-29124-1_9.
2. S. Althaher, P. Mancarella, and J. Mutale. Automated demand response from home energy management system under dynamic pricing and power and comfort constraints. *IEEE Transactions on Smart Grid*, 6(4):1874–1883, July 2015.
3. B. Priya Esther and K. Sathish Kumar. A survey on residential demand side management architecture, approaches, optimization models and methods. *Renewable and Sustainable Energy Reviews*, 59:342 – 351, 2016.
4. Long Duy Ha, S. Ploix, E. Zamai, and M. Jacomino. Tabu search for the optimization of household energy consumption. In *Information Reuse and Integration, 2006 IEEE International Conference on*, pages 86–92, Sept 2006.
5. M. Pipattanasomporn, M. Kuzlu, S. Rahman, and Y. Teklu. Load profiles of selected major household appliances and their demand response opportunities. *Smart Grid, IEEE Transactions on*, 5(2):742–750, March 2014.
6. Ana Soares, Carlos Henggeler Antunes, Carlos Oliveira, and ÃĄlvaro Gomes. A multi-objective genetic approach to domestic load scheduling in an energy management system. *Energy*, 77:144–152, December 2014.
7. Kathleen Spees and Lester B Lave. Demand response and electricity market efficiency. *The Electricity Journal*, 20(3):69–85, 2007.
8. E.G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.
9. Jiawei Zhu, Fabrice Lauri, Abderrafiaa Koukam, and Vincent Hilaire. *Artificial Intelligence Applications and Innovations: 11th IFIP WG 12.5 International Conference, AIAI 2015, Bayonne, France, September 14-17, 2015, Proceedings*, chapter Scheduling Optimization of Smart Homes Based on Demand Response, pages 223–236. Springer International Publishing, Cham, 2015.

Parallel metaheuristics

Solving the Uncapacitated Single Allocation p-Hub Median Problem on GPU

A. Benaini[1], A. Berrajaa[1,2], J. Boukachour[1], M. Oudani[1,3]
1. Normandie UNIV, UNIHAVRE, LMAH, 76600 Le Havre, France.
2. University Mohammed I, LaRI, Oujda, Morocco.
3. University Sidi Mohammed ben abdellah, MSC Lab, Fez, Morocco.
abdelhamid.benaini@univ-lehavre.fr, berrajaa.achraf@gmail.com,
jaouad.boukachour@univ-lehavre.fr, mustapha.oudani@usmba.ac.ma

Abstract: A parallel genetic algorithm (GA) implemented on GPU clusters is proposed to solve the Uncapacitated Single Allocation p-Hub Median problem. The GA uses binary and integer encoding and genetic operators adapted to this problem. Our GA is improved by generated initial solution with hubs located at middle nodes. The obtained experimental results are compared with the best known solutions on all benchmarks on instances up to 1000 nodes. Furthermore, we solve our own randomly generated instances up to 6000 nodes. Our approach outperforms most well-known heuristics in terms of solution quality and time execution and it allows hitherto unsolved problems to be solved.

Keywords: Parrallel genetic algorithms, GPU, CUDA, p-hub median problem.

1 Introduction

Hubs are sort of facilities that serve to transfer, transhipment and sort in a many-to-many complex distribution networks. They find their applications in airline passengers and fret networks, telecommunications and postal delivery networks. In the air traffic, hubs are the central airports for the long haul by cargo planes for goods and major carriers of passengers. In the telecommunication networks, hubs may be concentrators, routers, multiplexers [26]. In the postal distribution networks, hubs are the major sorting centre and cross docking messaging. The development of this type of network is due to economy of scale achieved by consolidating the traffic through the hub-hub arcs [1].

A rich scientific literature about hub location problems has been developed since 1980 and articles number has increased recently. Different variants of hub location problems have been defined and classified according to allocation way: the single allocation where each spoke (non-hub node) is assigned to exactly one hub and the multiple allocation that enables the spokes to be allocated to several hubs. The p-hub median problem when the number p of hubs to be located is given otherwise the problem is hub location. According to hubs capacities, the problem is said to be Uncapacitated (resp. capacitated) if hubs have infinite (resp. finite) capacities. There are several other kinds of hub problems like the p-hub centre problem where the objective is to minimize the maximum travel time between two demand centres [8], the hub arc problem which aims to overtake the shortcoming of the p-hub median problem by introducing the bridges arcs between hubs without discount factor [9], the dynamic hub location problem where either cost, demands or resources may vary in the planning horizon [12]. Other constraints can be taken into account, such as, hubs congestion, non-linear costs, stochastic elements, or vehicles routing constraints [14]. Reviews, synthesis and classification on models and methods used in literature on different variants of the hub location problem can be found in [4], [10], [19], [23], [30].

This paper deals with the Uncapacitated Single Allocation p-Hub Median Problem (USApHMP) for which we propose a parallel GA approach on GPU. To our knowledge, this is the first parallel GPU implementation for solving this problem. The remainder of this paper is organized as follows, related works are provided in Section 2. In Section 3, we present the mathematical formulation of the problem. The parallel GA approach is described in Section 4 followed by the GPU implementation in Section 5. Computational results are reported in the Section 6 and finally Conclusion and perspectives are given in Section 7.

This work was partially supported by the Region Normandy under CLASSE project.

2 Related Works

O'kelly et al. [31] presented the first mathematical formulation for the USApHMP as a quadratic integer program. They developed two heuristics and reported numerical results for CAB data (Civilian Aeronautics Board) with 25, 20, 15 and 10 nodes. Campbell et al. [8],[9] presented different formulations for the p-hub median problem, the uncapacitated hub location problem, the p-hub center problem and the hub covering problem. Possible extensions with flow thresholds are also studied. They introduced the p-hub median problem and proposed two heuristics to handle instances with 10–40 nodes and up to 8 hubs. Skorin-Kapov et al. [35], developed different mixed 0–1 linear formulations for the multiple and the single p-hub median problems and reported results on CAB data set. Sohn and Park [36], studied the special case of single allocation two-hubs location problem. In this particular case, the quadratic program is transformed to a linear program and to a minimum cut problem. Abdinnour-Helm [1] proposed a hybrid GA and tabu search heuristic and reported the results on the CAB data set. Ernst and Krishnamoorthy [16], presented a solving approach to the multiple allocation p-hub median problem and described how the approach can be adapted to the single allocation case. Results are reported on AP data for multiple allocation case up to 200 nodes. Bryan [7] studied four hub-and-spoke networks. The first is concerned by capacitated network, the second focus in minimum threshold model, the third determines the numbers of open hubs and the last introduce flow-dependent cost function.

Horner and O'Kelly [21] proposed a model implemented in a GIS environment to prove that hub networks may emerge naturally on traffic networks to take advantages of economies of scale. Labbé et al. [26] studied the polyhedral properties of the single assignment hub location problem and proposed a Branch-and-Cut algorithm for solving this variant of hub location. Chen [11] proposed a hybrid heuristic to solve the USAHLP based on a combination of an upper bound method search, simulated annealing and tabu list heuristic. Tests were performed on CAB data and AP data up to 200 nodes. Silva and Cunha [34] proposed three variants of tabu search heuristics and a two-stage integrated tabu search to solve the problem. The authors used the multi-start principle to generate different initial solutions which are improved by tabu search. They solved larger instances with 300 and 400 nodes. Ilic et al. [22] proposed a general variable neighborhood search for the USApMLP. They reported the results on AP and PlanetLab instances and Urand instances up to 1000 nodes. de Camargo and Miranda [14], introduced the single allocation hub location problem under congestion. A generalized Benders decomposition algorithm is proposed to solve AP instances.

Maric et al. [27] proposed a memetic algorithm based on two local search heuristics. They tested their algorithm on the well-known benchmarks and created larger scale instances with 52–900 nodes. They gave the optimal solutions of AP data up to 200 nodes. Bailey et al. [5] proposed a Discrete Particle Swarm Optimization (DPSO) to solve the USAHLP. They obtained the optimal solutions on all CAB data set and on AP data up to 200 nodes. Damgacioglu et al. [13] introduced a planar version of the uncapacitated hub single allocation hub location problem. This version has the particularity that a hub can be located anywhere in the plan. They reported the results on benchmarks AP data instances. Ting and Wang [38] proposed a threshold accepting TA algorithm to solve the USAHLP and reported results on the AP and CAB benchmarks. Meier and Clausen [28] made use of the data set structures to propose new linearization of the quadratic formulation of the problem. Indeed, the Euclidean distance in instances enabled to get linearization of three classical and two new formulations of the single allocation problem. They obtained optimal solutions on the AP data up to 200 nodes. Rostami et al. [33] introduced a new version of the USApHMP where the discount factor between hubs representing scale economy in hub-hub arcs is replaced by a decision variable. They proposed a Branch-and-bound algorithm and Lagrangian relaxation to compute lower bounds. Recently, Abyazi-Sani and Ghanbari [2] proposed a Tabu Search heuristic for solving the USAHLP and reported the results both on CAB data and AP data set up to 400 nodes. Kratica [25] proposed a GA for solving the uncapacited multiple allocation hub problem. Binary encoding and adapted genetic operation to this problem are used (only allocation hubs are given as the solution). He shows, under experimental results on ORLIB instances with up to 200 nodes that GA approach quickly reaches all optimal solution that are known. Topcuoglu et al. [39] present a GA approach to solve the uncapacited hub location problem. We use their encoding and GA operators in our parallel GA. However, we generate initial solutions differently from the middle nodes (rather than randomly initial solution as in [39]) with aiming to reach more quickly the best solutions.

Parallel GA implementations have been the subject of many works. There is extensive emerging research in this field and several studies suggest different strategies to implement GAs on different parallel machines [3], [18], [20], [24], [32], [36], [37]. There are three major types of Parallel GAs: (1) the master-slave model, (2) island model and (3) fine-grained model. In the master-slave model, the master node holds the population and performs most of the GA operations. The fitness evaluation, the crossover, the correction and mutation operations on groups of individuals are made by each slave. In a coarse-grained model, the

operations. In fine-grained models, each node has a single individual, and which it executes GA communicate with several neighboring nodes. In this case, the population is the collection of all the individuals in each node. There are conflicting reports over whether multiple independent runs of GAs with small populations can reach solutions of higher quality or can find acceptable solutions faster than a single

In this work, we propose GPU implementation of GA for solving the USApHMP. Several GPU implementations of parallel GA are proposed in the literature. Among them, [6], [32] presented the mapping of the parallel island-based GA on GPU. Our approach is similar to these implementations, nevertheless, the migration step is replaced by a selection of the best solutions in each iteration, and the generation of the initial solution is quite different (from the middle nodes).

3 Problem formulation

The USApHMP can be stated as follows: given N nodes 1…N, we try to locate p hubs and to find an optimal allocation of spokes to hubs (one hub for each spoke) that minimizes the sum of the total flow cost.

Let Z_{ik} be the binary decision variable equal to 1 if the node i is assigned to the hub k, 0 otherwise, Y^{i}_{kl} the flow between the hubs k and l originated from the node i, C_{ik} the unit cost for the flow in the arc (i,k), O_i and D_i are the originated and destined flow to the node i respectively.

The USApHMP is formulated as a MIP (Mixed Integer Program) by Ernst and Krishnamoorthy [15] as follows:

$$minimize \sum_{i}\sum_{k} C_{ik} Z_{ik} (\chi O_i + \delta D_i) + \sum_{i}\sum_{k}\sum_{l} \alpha C_{kl} Y^{i}_{kl} \quad (1)$$

Subject to:

$$\sum_{k} Z_{ik} = 1, \forall\, i \in N \quad (2)$$

$$Z_{ik} \leq Z_{kk}, \forall\, i,k \in N \quad (3)$$

$$\sum_{l} Y^{i}_{kl} - \sum_{l} Y^{i}_{lk} = O_i Z_{ik} - \sum_{j} W_{ij} Z_{jk}, \forall\, i,k \in N \quad (4)$$

$$\sum_{k} Z_{kk} = p \quad (5)$$

$$Z_{ik} \in \{0,1\}, \quad 1 \leq i,k \leq N$$

$$Y^{i}_{kl} \geq 0, \quad 1 \leq i,l,k \leq N$$

The objective function (1) minimizes the total cost of flow transportation between all origin-destination nodes. Constraint (2) imposes to each spoke to be assigned to exactly one hub (additionally each hub is allocated to itself). Constraint (3) requires that spokes will be assigned to hubs if the last one were open. Constraint (4) is the flow conservation constraint and constraint (5) imposes to locate exactly p hubs.

The USApHMP is known to be NP-hard with exception of special cases that are solved in polynomial time. When the set of hubs is fixed then the problem can be solved in $O(n^3)$ time using the shortest-path algorithm [17].

4 Genetic algorithm description

Genetic algorithms are well-known search approaches that are applied in the wide field of optimization. So, we propose a parallel GA to solve the USApHMP on GPU. Our implementation quickly

reaches the optimal or best solutions for all benchmarks. In the next subsection we detail the encoding chromosomes and how generate the initial solution of this problem.

4.1 Encoding and initial solution

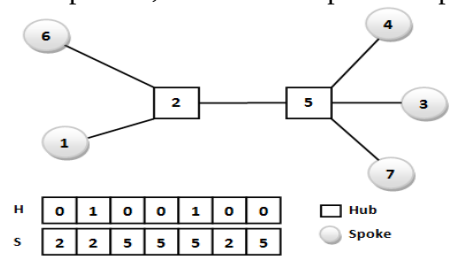
Each solution of the problem is represented by two N-arrays H and S (this encoding was used in [39]) where:

- H represents hub locations i.e H[i]= 1 if node i is a hub, H[i]=0 otherwise.
- S represents the allocation of spokes (non-hub nodes) to hubs i.e S[i]= k where k is the assigned hub for the node i. Additionally, each hub is allocated to itself.

In [39], the initial solution is generated pseudo randomly. Here, we proceed differently in order to quickly achieve best hubs locations. So to build an initial solution with p hubs, we first compute the p middle nodes i.e. the p hubs i with smallest distances d_i to the other nodes with $d_i = \sum_j C_{ij}$. Thus, the p initial hubs are chosen among the p middle nodes. Then each node is allocated to its nearest hub.

Numerical example:

The Fig .1 shows an example of a solution with 7 nodes, 2 hubs (nodes 2 and 5). The nodes 1, 2 and 6 are allocated to hub 2 and the other nodes are allocated to node 5. The encoding of this solution is given in Fig. 1:



The initial population is generated by duplication of the initial solution by randomly permuting one hub with one spoke.

Fig. 1: simple network encoding

4.2 Genetic operators

Random single point-crossover operator is used and infeasible offspring are corrected by a specific operator to ensure validity of solutions, in terms of number of hubs by assigning the corresponding spokes to their neighbor hubs. The permutation of two hubs is used as a mutation operator. These operators are noted crossover(), correction() and mutation() respectively.

4.3 Solution evaluation

The following Eval() definitions (fitness) are used in the standard benchmarks to evaluate the solutions quality. The fitness version for CAB data is given by:

$$(\sum_i \sum_k C_{ik} Z_{ik} (\chi O_i + \delta D_i) + \sum_i \sum_k \sum_l \alpha C_{kl} Y^i_{kl}) * 1/ \sum_i \sum_j W_{ij}$$

The fitness version for all data instances except PlanetLab and CAB data is given by:

$$10^{-3}(\sum_i \sum_k C_{ik} Z_{ik} (\chi O_i + \delta D_i) + \sum_i \sum_k \sum_l \alpha C_{kl} Y^i_{kl})$$

Note that the reason to multiply by 10^{-3} is to obtain the unit cost for flow transportation. We discover this when we tried to reproducing optimal solutions and we confirm it by contacting M.R Silva [34].

5 GPU implementation

The Graphics Processing Units are now available in most of personal computers. They are used to accelerate the execution of variety of problems. The smallest unit in GPU that can be executed is called thread. Threads (all executing the same code and can be synchronized) are grouped into blocks of equally sized and blocks are grouped in grid (blocks are independent and cannot be synchronized).

The memory hierarchy of the GPU consists of three levels: 1) the global memory that is accessible by all threads. 2) the shared memory accessible by all threads of a block and 3) the local memory (register) accessible by a thread. Shared memory has a low latency (2 cycles) and is of limited size. Global memory has a high latency (400 cycles) and is of large size (4 GB for the Quadro). An entire block is assigned to a single SM (Stream Multiprocessor). Each SM is composed of 32 streaming processors that share a limited size shared memory. Several blocks can run on the same SM. Each block is divided into Warps (32 threads

by Warp) that are executed in parallel. The programmer must control the block sizes, the number of Warps and the different memories access.

A typical CUDA program is a C program where the functions are distinguished based on whether they are meant for execution on the CPU or on the GPU. The functions executed on the GPU are called kernels and are executed by several threads. We implemented the GA on GPU (Nvidia Quadro with 4 GB and 384 cores running under CUDA 7.5 environment) and we compare it to sequential implementations of best known results existing articles in the literature in terms of time computations and on solutions quality. We showed the effectiveness of our implementation on several instances of the USApHMP.

Fig. 2 gives the schema of the parallel GA implementation on GPU. The following parameters are used: The number of node N, the population size n, the number of generations R, the number of iterations in the inner-loop N1, the number of iterations in the outer-loop N2.

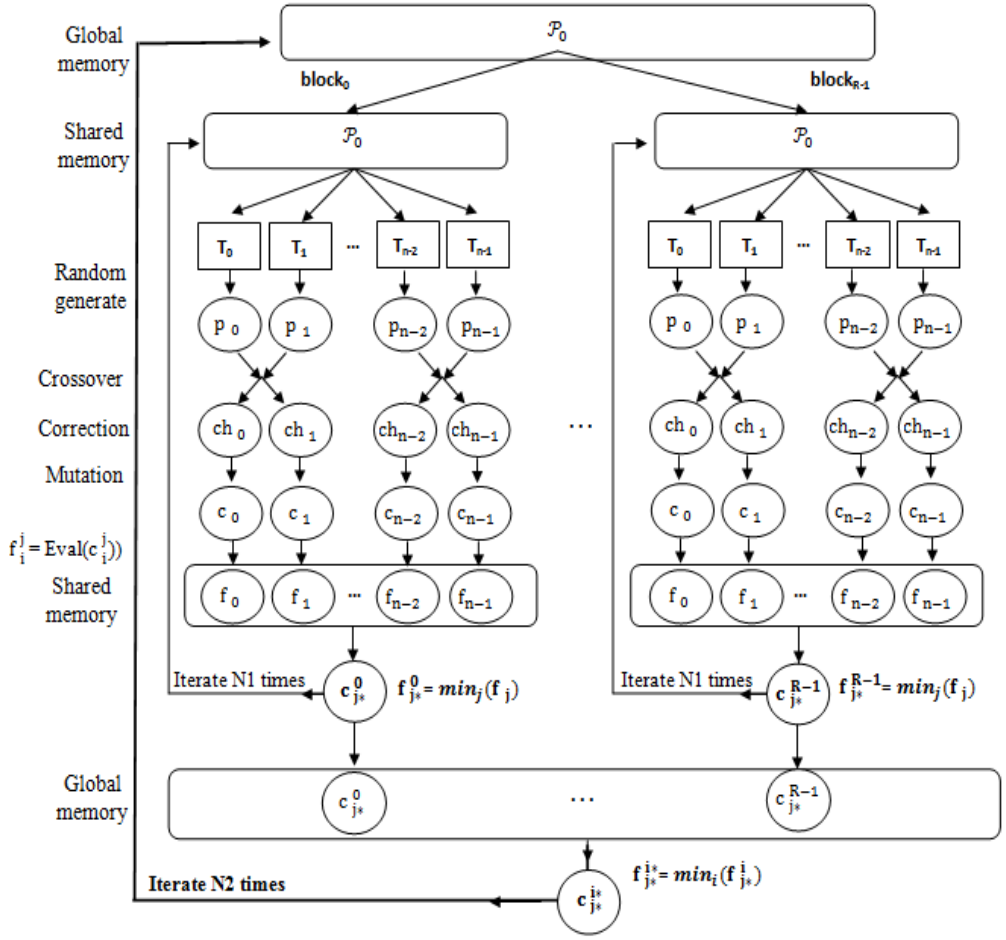


Fig. 2 The schema of the parallel GPU implementation of the GA.

We partition the GPU on R blocks each one is a gird n x 1 of threads. The master thread of each block is the thread 0 and the global master thread is the thread 0 of the block 0. The block i, $0 \leq i < R$ stores in its shared memory the data required to execute one GA starting from an ancestor individual $\mathcal{P}_0$(initial feasible solution) generated as indicated in section 3 by the CPU and copied in the global memory of the GPU.

More precisely, let $T^i_0, \ldots, T^{R-1}_{n-1}$ the threads of the block i. Starting from $\mathcal{P}^i_0$, each thread T^i_j generate a new solution (individual) p^i_j by applying a random permutation to $\mathcal{P}^i_0$ (initially, $\mathcal{P}^i_0 = \mathcal{P}_0$ and is updated after each iteration of the inner-loop). $\mathcal{P}^i_0 = p^i_0, \ldots, p^i_{n-1}$ is the initial population of the GA executed by the block i. Note that the population size n is the same for all the blocks.

Now, we explain how the block i executes the GA. Each thread T^i_{2j} of the block i generate two children namely ch_1 and ch_2 by crossowing the parents p^i_{2j}, p^i_{2j+1}then T^i_{2j} applies the mutation to $ch_1$to get a new individual say c^i_{2j} and T^i_{2j+1} applies the mutation to ch_2 to get a new individual say c^i_{2j+1}. Next, each thread

T_j^i executes the correction operator to ensure the validity of the solution, verifies that all nodes are assigned to the nearest hubs and finally it computes $f_j^i = Eval(c_j^i)$.

Note that all c_j^i and f_j^i are stored in the shared memory of the block i. So, the master thread of block i selects the individual c_{j*}^i with $min_j\{f_j^i\}$ and updates the ancestor s_0^i as c_{j*}^i for the next iteration. This inner-loop of GA terminates after N1 iterations (the same for all the blocks).

The c_{j*}^i, $0 \leq i < R$, are copied in the global memory and the individual c_{j*}^{i*} with the $min_i\{f_{j*}^i$ is selected as the final solution or as the new value of the ancestor s_0 for the next iteration of the outer-loop. The process is repeated N2 times.

The pseudo CUDA code executed by the CPU is the following:

1. Generate($\mathcal{P}_0$); //the ancestor individual
2. Copy $\mathcal{P}_0$in the global memory of the GPU.
3. Define the blocks and the grid :
   dim3 dimBlock(n,1);
   dim3 dimGrid(R,1);
4. Launch the kernel GA($\mathcal{P}_0$) : GA<<<dimGrid,dimBlock>>>($\mathcal{P}_0$);
5. Read the solution from the global memory.

6 Computational results

6.1 Benchmarks used

We used four types of data: CAB, AP, PlanetLab and Urand :

- CAB (Civilian Aeronautic Board) data set is set of instances introduced in [31] based on airline passenger flow between 25 US cities. It contains distances (which satisfy triangle inequality) and symmetric flow matrix between the cities. The size instances are of 10, 15, 20 and 25 nodes. The distribution and collection factors δ and χ are equal to 1.

- AP (Australian Post) data set are real-world data set representing mail flows in Australia. The distribution and collection factor δ and χ equal 3 and 2 respectively while the discount factor α takes 0.75 for all instances. The mail flows are not symmetric and there are possible flows between each node and itself.

- Urand data set are random instances up to 400 nodes generated by Meyer et al. [29]. The instances with 1000 nodes were generated by Ilic et al. [22]. Nodes coordinates were randomly generated from 0 to 100000 and the flow matrix was randomly generated.

- The PlanetLab instances are node-to-node delay information for performing Internet measurements [22]. In these networks, $\chi = \alpha = \delta = 1$ and the distance matrix doesn't respect triangle inequality.

6.2 Best known solutions vs. our results

We report the results for the three data set introduced above. We compare our results with those of Ilic et al. [22] in terms of computing time. Note that in our GPU implementation, the number of blocks is the same for all problems. So time compute of all problems is the same. We use shared memory to reduce the time computation. However, the time transfer between the CPU and GPU varies according to the number of nodes. Throughout the rest the given times are the time of the complete program (calculation of the initial solution, data transfers between the CPU and GPU, calculation of the solution).

For the CAB data set, we obtained the optimal solutions in all instances (up to 25 nodes) in a short computing time. Since solving these instances is not anymore a challenge (all instances are solved to optimality by previous work), we report only our computing times for solving these data instances. We studied the scale economy generated in hub-hub arcs and its relationship with initial and final costs which represent the collection and distribution cost. Typically, a hub-and-spoke transportation chain is composed by tree segments: the first and the third called pre and post haul respectively are the initial and final arcs while the second is the long haul segment (hub-hub arcs). In CAB data, we can express the cost from an origin i to a destination j through the two hubs k and l as: $\boldsymbol{C_{ij}^{kl} = \chi C_{ik} + \alpha C_{kl} + \delta C_{jl}}$ where $\alpha \leq 1$ represents the scale economy generated by consolidating flows between hubs while χ and δ represent the distribution and the collection costs and are often greater than 1. A question is how distribution and collection cost

influence the scale economy thresholds? In fact, as illustrated in Fig. 3, the average inter-hub distance changes as we vary the distribution and collection factors. We can see clearly that the long-haul relevance threshold is lower when the distribution and collection costs are lower.

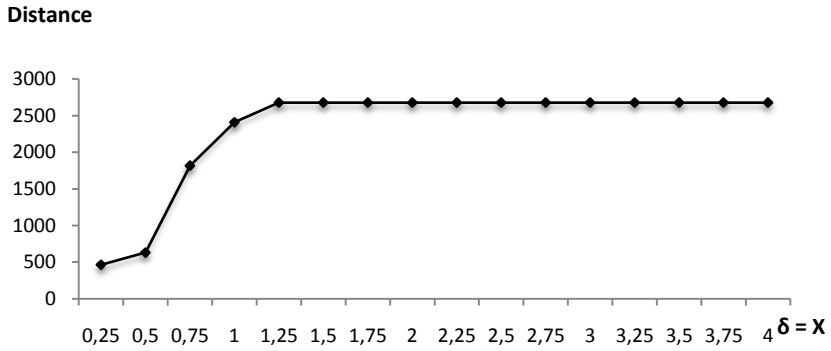


Fig. 3 The average inter-hubs distance

The following notations are used in Tables 1-5:

- N: nodes number in the instance.
- p: hubs number.
- Best Sol: the Best solution if it is known otherwise "-"is written.
- GPU Sol: the best solution obtained by GPU, with mark "opt" when solution in GPU is the optimum for the current instance.
- T_{OPT} : the best time (in seconds) for the best solution.
- T_{GPU} : the time (in seconds) for our GA parallel.

These tables give a comparison of our results to the best known results for USApHMP in the benchmarks previously introduced. As shown in Table 1 we obtained optimal solutions for all the AP data instances in time ≤ 7.42 s. Note that, the results using AP data instances for the p-hub median variant with 300 and 400 nodes are not reported before in the literature and we think that finding exact solutions using standard solvers (CPLEX, Gurobi...) is a serious challenge. So, we can think that our results are since now the best solutions for 300 and 400 nodes instances. We report our results for PlanetLab instances in Table 2. It is clear that our approach outperforms those of literature [22] either in cost and computing time. The state of the art solutions given in [22] reports the results for 12 instances. Each instance is characterized by nodes number n and by p hubs to be located with p ≈$\sqrt{n}$.

Table 1: Results on AP data

N	p	Best Sol	GPU Sol	T_{GPU}	N	p	Best Sol	GPU Sol	T_{GPU}
10	2	167493.06	opt	0.007	100	5	136929.444	opt	1.310
	3	136008.13	opt	0.012		10	106469.566	opt	1.310
	4	112396.07	opt	0.014		15	90533.523	opt	1.49
	5	91105.37	opt	0.019		20	80270.962	opt	1.63
20	2	172816.69	opt	0.020	200	5	140062.647	opt	3.602
	3	151533.08	opt	0.031		10	110147.657	opt	3.722
	4	135624.88	opt	0.039		15	94459.201	opt	3.783
	5	123130.09	opt	0.043		20	84955.328	opt	3.841
25	2	175541.98	opt	0.033	300	5	-	**174914.73**	5.631
	3	155256.32	opt	0.045		10	-	**134773.55**	5.711
	4	139197.17	opt	0.050		15	-	**114969.85**	5.896
	5	123574.29	opt	0.061		20	-	**103746.44**	5.876
40	2	177471.67	opt	0.063	400	5	-	**176357.92**	6.741
	3	158830.54	opt	0.110		10	-	**136378.19**	6.846
	4	143968.88	opt	0.167		15	-	**117347.10**	7.102
	5	134264.97	opt	0.213		20	-	**104668.27**	7.423
50	2	178484.29	opt	0.092					
	3	158569.93	opt	0.163					
	4	143378.05	opt	0.250					
	5	132366.953	opt	0.271					

Table 2: Results on PlanetLab

Instance	N	p	Best Sol	GPU Sol	T_{OPT}	T_{GPU}
01-2005	127	12	2927946	**2904434**	148.954	**0.47**
02-2005	321	19	18579238	**18329984**	462.790	**6.95**
03-2005	324	18	20569390	**20284132**	543.844	**7.54**
04-2005	70	9	739954	**730810**	0.682	**0.28**
05-2005	374	20	25696352	**25583240**	622.612	**8.32**
06-2005	365	20	22214156	**22191592**	581.776	**7.94**
07-2005	380	20	30984986	**30782956**	546.688	**8.47**
08-2005	402	21	30878576	**30636170**	637.686	**8.74**
09-2005	419	21	32959078	**32649752**	684.900	**9.34**
10-2005	414	21	32836162	**32687796**	731.930	**9.12**
11-2005	407	21	27787880	**27644374**	588.344	**9.22**
12-2005	414	21	28462348	**28213748**	680.382	**9.18**

Table 3: Results on Urand instances

N	p	Best Sol	GPU Sol	T_{GPU}	N	p	Best Sol	GPU Sol	T_{GPU}
100	2	36930.31	opt	0.0375	300	2	328702.42	opt	0.2215
	3	34532.88	opt	0.0265		3	308765.08	opt	0.9175
	4	32608.28	opt	0.0245		4	293636.81	opt	1.6100
	5	31107.70	opt	0.1135		5	282116.88	opt	0.5060
	10	27058.40	opt	0.4695		10	251393.30	opt	12.8275
	15	25408.56	opt	2.7925		15	236781.77	opt	27.6640
	20	24377.65	opt	7.3640		20	228005.19	opt	153.2925
200	2	148235.45	opt	0.0175	400	2	579982.35	opt	0.1735
	3	139223.25	opt	0.0575		3	543717.32	opt	1.7115
	4	132676.89	opt	0.3920		4	519217.48	opt	2.2275
	5	127220.02	opt	0.6895		5	501421.52	opt	1.4730
	10	112539.21	opt	4.5300		10	446361.10	opt	16.8700
	15	105690.52	opt	37.3460		15	422284.78	opt	111.4295
	20	102022.32	opt	68.6685		20	407110.51	opt	228.8615

Table 4: Results on Urand large instances

N	p	Best Sol	GPU Sol	T_{OPT}	T_{GPU}
1000	2	198071412.53	**8184986.50**	1.7245	9.321
	3	169450816.35	**7024184.00**	8.1550	9.785
	4	150733606.87	**6184749.01**	2.2240	10.431
	5	142450250.26	**5860994.06**	58.6070	**10.89**
	10	114220373.07	**4752317.00**	187.8385	**13.7**
	15	-	**4228256.88**	-	15.23
	20	198071412.53	**3928617.48**	403.4280	17.923

Table 5: Results on larger Urand instances (generated by us)

N	p	GPU Sol	T_{GPU}	N	p	GPU Sol	T_{GPU}
1500	20	**454787506**	196	**4000**	20	**3234999192**	3076
	30	**407155164**	286		30	**2983891783**	3276
	40	**380114045**	423		40	**2769550514**	3365
	50	**363586538**	574		50	**2644606684**	3648
2000	20	**805749722**	477	**5000**	20	**5085803132**	4662
	30	**733375448**	580		30	**4656787498**	4720
	40	**686515363**	714		40	**4353561395**	4996
	50	**655938000**	965		50	**4143849388**	5112
3000	20	**1804950952**	1157	**6000**	20	**7398401957**	5614
	30	**1642145354**	1544		30	**6675723961**	5748
	40	**1538548764**	1869		40	**6293053841**	5964
	50	**1468780124**	2086		50	**5999780197**	6212

The Table 3 reports computational results for the Urand instances. We can see that our parallel GA obtained the best solutions for instances up to 400 nodes and outperforms those of Ilic et al., [22] for instances with 1000 nodes as illustrated in Table 4. Concerning the computing times, our approach is faster and gets the solutions in a time lapse less than 18s for all instances while the best-known time reaches 7 minutes. A remarkable thing is that the time execution gap of our algorithm with Ilic et al., [22] algorithm is important for large values of p. We report in Table 5 results for larger instances generated by us using the same generation procedure as for the Urand instances as stated in [29]. These new challenging instances consist of large networks up to 6000 nodes that have not been solved before.

7 Conclusion and perspectives

We developed a parallel GA for the Uncapacitated Single Allocation p-Hub Median problem and we implement it on GPU. We showed the effectiveness of our implementation on the well-known benchmarks for this problem. Indeed, our approach improved the best known solutions in cost and computing times for well-known benchmarks instances with up to 1000 nodes. Also it allowed solving large instances problem unsolved before. Further, we work on the design and implementation of an exact parallel tree-based algorithm to solve the studied hub problems as these algorithm structures seems to be suitable for the GPU architectures. Another issue is to tackle other versions of the hub problem especially capacitated case, multiple allocation variants and other more specific problems (with congestion, with vehicles routing constraints, etc.). Other metaheuristics in particular those based on one solution may be studied from the parallelism viewpoint.

Acknowledgments

We thank Dr, M. O'Kelly, M.R. Silva, C.B. Cunha, Ilic A and R. Abyazi-Sani for instances data set provided.

References

[1] Abdinnour-Helm, S. (1998). A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operational Research*, *106*(2), 489–499.
[2] Abyazi-Sani, R., & Ghanbari, R. (2016). An efficient tabu search for solving the uncapacitated single allocation hub location problem. *Computers & Industrial Engineering*, *93*, 99–109.
[3] Alba E, Troya JM (1999). A Survey of Parallel Distributed Genetic Algorithm. Journal Complexity 4(4): 31–52.
[4] Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European journal of operational research*, *190*(1), 1–21.
[5] Bailey, A., Ornbuki-Berrnan, B., & Asobiela, S. (2013, April). Discrete pso for the uncapacitated single allocation hub location problem. In *Computational Intelligence in Production and Logistics Systems (CIPLS), 2013 IEEE Workshop on* (pp. 92–98). IEEE.
[6] Benaini, A., Berrajaa, A. and Sbihi, A. (2016). An adapted GA for solving the steel continuous casting problem on GPU, submitted to HPCS 2016 Austria.
[7] Bryan, D. (1998). Extensions to the hub location problem: Formulations and numerical examples. *Geographical Analysis*, *30*(4), 315–330.
[8] Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, *72*(2), 387–405.
[9] Campbell, J. F., Ernst, A. T., & Krishnamoorthy, M. (2005). Hub arc location problems: part I—introduction and results. *Management Science*, *51*(10), 1540–1555.
[10] Campbell, J. F., & O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, *46*(2), 153-169.
[11] Chen, J. F. (2007). A hybrid heuristic for the uncapacitated single allocation hub location problem. *Omega*, *35*(2), 211–220.
[12] Contreras, I., Cordeau, J. F., & Laporte, G. (2011). The dynamic uncapacitated hub location problem. *Transportation Science*, *45*(1), 18–32.
[13] Damgacioglu, H., Dinler, D., Ozdemirel, N. E., & Iyigun, C. (2015). A genetic algorithm for the uncapacitated single allocation planar hub location problem.*Computers & Operations Research*, *62*, 224–236.

[14] de Camargo, R. S., & Miranda, G. (2012). Single allocation hub location problem under congestion: Network owner and user perspectives. *Expert Systems with Applications*, *39*(3), 3385–3391.
[15] Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, *4*(3), 139–154.
[16] Ernst, A. T., & Krishnamoorthy, M. (1998). Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, *104*(1), 100–112.
[17] Ernst, A. T., & Krishnamoorthy, M. (1999). Solution algorithms for the capacitated single allocation hub location problem. *Annals of Operations Research*, *86*, 141–159.
[18] E. Talbi Metaheuristics on GPU, J.Parallel Distributed Comput 73(1)1-3(2013).
[19] Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, *64*(4), 1096–1109.
[20] Gen M, Katai O, McKay B, Namatame A, Sarker RA, Zhang B.-T (Eds.) (2009). Intelligent and Evolutionary Systems, SCI 187, pp: 19-216.Springer-Verlag.
[21] Horner, M. W., & O'Kelly, M. E. (2001). Embedding economies of scale concepts for hub network design. *Journal of Transport Geography*, *9*(4), 255-265.
[22] Ilić, A., Urošević, D., Brimberg, J., & Mladenović, N. (2010). A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, *206*(2), 289–300.
[23] Klincewicz, J. G. (1998). Hub location in backbone/tributary network design: a review. *Location Science*, *6*(1), 307-335.
[24] Knysh DS, Kureichik VM (2010). Parallel genetic algorithms: a survey and problem state of the art, Journal of Computer and Systems Sciences International.49(4): 579–589.
[25] Kratica, J., Stanimirović, Z., Tošić, D., & Filipović, V. (2012). Genetic algorithm for solving uncapacitated multiple allocation hub location problem. Computing and Informatics, 24(4), 415-426.
[26] Labbé, M., Yaman, H., & Gourdin, E. (2005). A branch and cut algorithm for hub location problems with single assignment. *Mathematical programming*,*102*(2), 371-405.
[27] Marić, M., Stanimirović, Z., & Stanojević, P. (2013). An efficient memetic algorithm for the uncapacitated single allocation hub location problem. *Soft Computing*, *17*(3), 445–466.
[28] Meier, J. F., & Clausen, U. (2015). Solving classical and new single allocation hub location problems on Euclidean data, 2015. *Available as: Optimization Online*, 03-4816.
[29] Meyer, T., Ernst, A. T., & Krishnamoorthy, M. (2009). A 2-phase algorithm for solving the single allocation p-hub center problem. *Computers & Operations Research*, *36*(12), 3143–3151.
[30] O'Kelly, M. E., & Miller, H. J. (1994). The hub network design problem: a review and synthesis. *Journal of Transport Geography*, *2*(1), 31–40.
[31] O'kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, *32*(3), 393-404.
[32] Pospichal P, Jaros J, Schwarz J (2010). Parallel Genetic Algorithm on the CUDA Architecture. C. Di Chio et al. (Eds.): Evo Applications, Part I, LNCS 6024, pp. 44–451.
[33] Rostami, B., Buchheim, C., Meier, J. F., & Clausen, U. (2015). Lower Bounding Procedures for the Single Allocation Hub Location Problem. *Electronic Notes in Discrete Mathematics, Accepted for Publication*, *320*.
[34] Silva, M. R., & Cunha, C. B. (2009). New simple and efficient heuristics for the uncapacitated single allocation hub location problem. *Computers & Operations Research*, *36*(12), 3152–3165.
[35] Skorin-Kapov, D., Skorin-Kapov, J., & O'Kelly, M. (1996). Tight linear programming relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research*, *94*(3), 582–593.
[36] Sohn, J., & Park, S. (1998). Efficient solution procedure and reduced size formulations for p-hub location problems. *European Journal of Operational Research*, *108*(1), 118–126.
[37] T.Van Luong T, Melab N and Talbi EG. GPU Computing for Parallel Local Search Metaheuristic Algorithms. IEEE Transactions on Computers 62(1) 173-185 (2013).
[38] Ting, C. J., & Wang, H. J. (2014). A threshold accepting algorithm for the uncapacitated single allocation hub location problem. *Journal of the Chinese Institute of Engineers*, *37*(3), 300–312.
[39] Topcuoglu, H., Corut, F., Ermis, M., & Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms. *Computers & Operations Research*, *32*(4), 967–984.

A CPU-GPU Parallel Approach to Solve Multiple Choice Multidimensional Knapsack Problem

M. BEN REJEB[1], S. HTIOUECH[2], and S. BOUAMAMA[3]

[1] COSMOS Lab, National School of Computer Science of Tunisia,
Faculty of Economics and Management, Nabeul
Benrejeb.medali@gmail.com
[2] COSMOS Lab, National School of Computer Science of Tunisia,
National school of engineering of Bizerte
Htiouech.Skander@enib.rnu.tn
[3] COSMOS Lab, National School of Computer Science of Tunis
Sadok.Bouamama@ensi-uma.tn

Keywords : combinatorial optimisation, multiple choice, knapsack problem, GPU

1 Introduction

The knapsack problem (KP) is a NP-hard combinatorial optimization problems used to model a wide variety of real life and literature problems[2]. The purpose is to maximize profit while having only limited resources. This problem generalizes different variants of other problems [4][5]. The multiple choice multidimensional knapsack problems (MMKP) are particular variants of the KP considered as one of the most complex member of his family [3][4].
Consider a set of item groups. Each item has a particular value in the objective function and requires a certain amount of resources. The MMKP consist to pick exactly one item from each group in order to maximize the total profit value of the pick without exceeding the capacities of the knapsacks [4]. Formally, the MMKP can be stated as follows :

$$\begin{array}{ll} \text{maximise} & \sum_{i=1}^{n} \sum_{j=1}^{n_i} c_{ij}^{k} x_{ij} \\ \text{Subject to} & \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij} \leq b^{k}, k = 1, ..., m \\ & \sum_{j=1}^{n_i} x_{ij} = 1, i = 1, ..., n \\ & x_{ij} \in \{0, 1\}, i = 1, ..., n, j = 1, ..., n_{ij} \end{array}$$

where $b = (b^1, b^2, ..., b^m)$ is the capacity vector of the multi-constrained knapsack resources, and a set of n disjoint item groups $N = (N^1, ..., N^i, ..., N^n)$ where each group $i, i = 1, ..., n$ has n_i items. Each item $j, j = 1, ...n_i$, of the i^{th} group has a non-negative profit value c_{ij} , and requires an amount of resources represented by the weight vector $a_{ij} = (a_{ij}^1, a_{ij}^2, ..., a_{ij}^k)$. Note that weight terms a_{ij}^k (with $1 \leq k \leq m; 1 \leq i \leq n; 1 \leq j \leq n_i$) are non negative.

The appearance of new architectures such as graphical processing units (GPUs) seems particularly interesting to reduce resolution time. The disjoint groups of MMKP allow the decomposition of the initial problem in several sub-problems. Each one will be therefore formed by a set of groups of the initial problem. Each set is considered as an MMKP. In this paper, we propose a new approach to solve the MMKP using CPU-GPU architecture using CUDA. To do this, we apply an efficient parallel implementation of the branch and bound algorithm developed by Hifi and Sbihi [1] on a CPU-GPU. For each sub-problem we associate a new thread to solve by Hifi and Sbihi s algorithm [1]. Therefor, we obtain a final solution with a reasonable quality that can be improved by using a local search.
A strategic management of GPUs memories and synchronization between GPU threads decreases significantly GPU-CPU communications. Such a process could make the approach solve larger sizes of instances.

2 Method description

First of all, the initial problem is decomposed into a set of sub-problems. The latter are pooled into different groups using a classification method. In a second step, and based on the amount of resources used by the sub-solutions, the available resources will be distributed on different sub-problems. Such distribution will guarantee to increase the quality of final solution.
In the third step, all instances will be therefore solved in parallel using an exact solution algorithm. Using such algorithms gives an optimal solution to each sub-problem. Hence providing a good solution to the initial problem. However, this solution is not necessarily optimal.
Therefore, to improve the solution obtained in third step, a second distribution of the remaining resources is performed. In fact, the sum of the remaining resources left by the solved sub-problems that are higher than a predefined threshold will be redistributed among the other sub-problems whom left fewer resources. These sub-problems will be subject of a second resolution using the new redistributed resources in order to improve their results.

The next algorithm makes up the framework of our method.
/* Initialization phase */
1. Decompose the initial problem into a set of sub-problems.
1.1. Divide the groups of items into several class of groups.
1.2. Assign each class of groups to a sub-problem.
1.3. Distribute the available resources between the different sub-problems.
/* Parallel resolution phase */
Repeat
2. Solve all sub-problems in parallel using Branch and Bound algorithm [1].
2.1. Copy data from host to device.
2.2. Solve each sub-problem with the algorithm approach using the best-first search strategy [1].
2.3. Copy the solutions of the sub-problems from device to host (Profit Solution = Sum of the profits of the sub-problems).
/* Intensification phase */
3. Repeat
3.1. Redistribute the remaining resources to the sub-problems which have used much of their resources.
3.2. Re-solve these sub-problems in parallel.
Until no improvements found
4. Intensification phase around the current solution to improve it (swap move [5]).
/* Diversification phase */
5. Swap groups between the sub-problems, which have used less of their resources.
Until the stopping criterion is met.

Our parallel approach is evaluated using medium size and large size benchmarks [7]. It improves the best literature known results and provides solutions with a total value on average equal to 1% close to the optimum and better performance than those obtained by Chao Gao et al. [6] and htiouech et al. [5].
Additionally, we define a new set of very large benchmark size (up to 64000 variables). Preliminary results show that our approach is always able to find very good solutions quality for new instances in a reasonable time of execution (less than 150 seconds).

References

[1] A. Sbihi : A best first search exact algorithm for the Multiple-choice Multidimensional Knapsack Problem, Paris (2007)
[2] Healy Jr : WC. Multiple choice programming. Operations Research ; 12 :122 38. (1964)
[3] Khan. S, K. Li, E. Manning, and M. Akbar : Solving the knapsack problem for adaptive multimedia systems. Studia Informatica, Special Issue on Combinatorial Problems, 2(2) :157-178 (2002)
[4] S.Htiouech, S.Bouamama, A Lagrangian and Surrogate information enhanced tabu search for the MMKP. 2014 IEEE World Congress on Computational Intelligence. IEEE WCCI. July 6-11, Beijing, China (2014)

[5] S. Htiouech, S. Bouamama and R. Attia, OSC : solving the multidimensional multi-choice knapsack problem with tight strategic Oscillation using Surrogate Constraints. The International Journal of Computer Applications . Volume 73(13) :1-7, 2013. Published by Foundation of Computer Science, New York, USA (2013)
[6] Youxin Xia, Chao Gao, JinLong Li : A Stochastic Local Search Heuristic for the Multidimensional Multiple-choice Knapsack Problem. 10th International Conference, BIC-TA 2015 Hefei, China, September 25-28, 2015,
[7] ftp://cermsem.univ-paris1.fr/pub/CERMSEM/hifi/

Metaheuristics for software engineering

Schema Theory for Software Test Data Generation

Abdelilah Sakti, Yann-Gaël Guéhéneuc, and Gilles Pesant

École Polytechnique de Montréal
Montréal, Quebéc
{abdelilah.sakti, gilles.pesant, yann-gael.gueheneuc}@polymtl.ca

Abstract. Genetic Algorithms (GA) have been effective in solving many search and optimization problems. Software testing is one field wherein GA have received much attention to solve the problem of test-data generation. However, few works analyze this problem to see whether GA are appropriate. Schema theory is an analysis tool that theoretically explains how and in which classes of problems GA work well. In this paper, we use schema theory to analyze the problem of test-data generation. Based on this analysis, we propose an enhanced GA framework for software test-data generation. It uses the schema analysis to identify performance schemata. To preserve these schemata during the GA evolution phase and maintain their diversity, our framework provides a new selection strategy called pairwise-selection as well as adaptive crossover and mutation operators. We evaluate our framework on a set of synthetic programs and compare it to the well known test-data generation framework, DaimlerChrysler system. We observe that a high branch coverage was obtained and the number of fitness evaluations was significantly reduced.

1 Introduction

Genetic algorithms (GA) are a family of global search algorithms that were initially proposed by Holland in the 1970s [1]. Genetic algorithms are often used as an optimizer, although they are becoming more widely used for practical problem solving and for scientific modeling. Evolutionary testing is one field wherein GA has received much attention [2–7].

In general, a search algorithm faces challenges in a large search space when the landscape of its fitness function contains many local optima and it is insufficiently informed. To tackle these challenges to GA for some classes of problems, Holland [1] proposed schema theory to analyze the problem and extract additional pieces of information and exploit them while the search goes on. On such a class of problems, GA guided by a fitness function based on schema theory is theoretically predicted to outperform simple GA and other search techniques, such as hill climbing. The *schema theory* or *building-blocks hypothesis* states that a GA can outperform other search algorithms if the chromosomes are distinguishable (can be represented by distinct schemata) and the combination of partial schemata (lower-order building blocks) can construct higher-order schemata [1, 8].

However, despite the theoretically predicted advantage of GA based on schema theory analysis, its application to evolutionary testing poses additional challenges. The schema theory analysis as defined by Holland is not directly applicable to evolutionary testing and some adaptations of the original schema theory analysis exist [5]. Also, to further increase its performance and to benefit from the power of schema theory analysis, in addition to an adequate schema theory form, the GA evolution operators must preserve schemata and maintain their diversity [9].

In this paper, we propose an enhanced GA framework for evolutionary testing (EGAF-ET) based on schema-theory analysis. We show that testing chromosomes are distinguishable according to the subset of individual conditions that they satisfy and may be classified in schemata. Conditional expressions of branches are used to define a general set of schemata for a test target. According to the proposed concept of schemata, we define a general expression of an evolutionary testing fitness function. After adapting the general form of schema theory for evolutionary testing, we identify structures and properties associated with better performance by schema analysis, then incorporate them in the evolution phase with all fundamental GA operators (selection, crossover, mutation). We report on the comparison of a simple GA framework for software test data generation and our EGAF-ET and show that the latter outperforms the former: EGAF-ET significantly reduces the number of evaluations needed to reach a given branch and achieves higher branch coverage.

Table 1: A set of possible schemata for a test target that is control dependent on three branches (b_1, b_2, b_3).

Schema	Meta-constraint	Pattern	Order
s_{11}	$\{(x_1, x_2, x_3) \mid b_1$ is satisfied$\}$	$b_1 \ * \ *$	1
s_{12}	$\{(x_1, x_2, x_3) \mid b_2$ is satisfied$\}$	$* \ b_2 \ *$	1
s_{13}	$\{(x_1, x_2, x_3) \mid b_3$ is satisfied$\}$	$* \ * \ b_3$	1
s_{21}	$\{(x_1, x_2, x_3) \mid b_1 \wedge b_2$ is satisfied$\}$	$b_1 \ b_2 \ *$	2
s_{22}	$\{(x_1, x_2, x_3) \mid b_1 \wedge b_3$ is satisfied$\}$	$b_1 \ * \ b_3$	2
s_{23}	$\{(x_1, x_2, x_3) \mid b_2 \wedge b_3$ is satisfied$\}$	$* \ b_2 \ b_3$	2
s_{31}	$\{(x_1, x_2, x_3) \mid b_1 \wedge b_2 \wedge c_3$ is satisfied$\}$	$b_1 \ b_2 \ b_3$	3

The remainder of the paper is organized as follows: Section 2 describes our framework and shows how to use schema theory on evolutionary testing. Section 3 we present experimental results comparing a simple GA framework with our EGAF-ET. Section 4 summarizes related work. The conclusions and directions for future work are in Section 5.

2 Schema Theory for Test-data Generation

Test-data generation aims to generate an input vector (i.e., individual) that satisfies a set of branch-conditions leading to a test target. Its search space can be split into subspaces where each subspace represents the set of individuals satisfying a given subset of branch-conditions. Thus, individuals are distinguishable by the subset of branch-conditions that they satisfy. Therefore, a schema can be defined as a set of individuals that satisfy a meta-condition (i.e., a conjunction of branch-conditions) and its order is equal to the number of conjuncts. An individual i is an instantiation of a schema if and only if i satisfies its meta-constraint. Thus, the set of schemata that represent a problem of test-data generation is all possible branch-conditions whereon the test target is control dependent. For example, given an Unit-under-test (UUT) that takes three input arguments x_1, x_2, and x_3, a test target t control dependent on three branches b_1, b_2, and b_3, can be modeled by a subset from the set of schemata given in Table 1. Each schema represents an exponential number of sub-schemata, where a sub-schema is defined by the meta-constraint of the main schema while fixing one input argument to a value from its domain.

A schema is a hierarchical structure: intermediate-order schemata (e.g., s_{21}, s_{23}) play the role of stepping stones to go from lower-order schemata (e.g., s_{11}, s_{12}) to higher-order schemata (e.g., s_{31}) [10]. The hierarchical structure is one essential feature for the *building-blocks hypothesis*, i.e., partial schemata construct higher-order schemata. Thus, individuals from a schema having a fitness value higher than the average are likely to produce fitter individuals [1]. Therefore, a fitness function based on the defined schema may work well under certain circumstances. Based on this schema, we can define different fitness functions for evolutionary testing. Mitchell et al. [10] termed this class of fitness functions *Royal Road functions*.

2.1 Royal Road Function

To define a royal road function for evolutionary testing, we use the concept proposed by Jones [11]. The fitness value is composed of two scores called PART and BONUS. PART considers each building block individually in such a way that each building block receives a fitness value. BONUS rewards optimal building blocks that reach their optimal fitness value. Because we define a building block by a branch-condition, we can define PART by a straightforward assignment of its normalized branch distance (η) [12]. If, for some reason, an exact evaluation of a building block is impossible (e.g., a non-executed branch and its expression is impossible to evaluate symbolically), then its PART can be overestimated by assigning it the value 1. In this case, if we ignore the BONUS (e.g., BONUS=0) then we fall on the Approach Level fitness function (f_{AL}) [7] and Symbolically Enhanced fitness function (f_{SE}) [13] expressions. In previous work [14], we studied branch-hardness (i.e., the difficulty to satisfy a branch) and showed that the Difficulty Coefficient (DC) is a good corrective metric to get comparable branch distances among different branches. In this paper, we use this metric to adjust branch distances. Using this adjustment, we fall on the Difficulty Coefficient fitness function (f_{DC}) expression [14].

BONUS allows the revised class of royal road functions to outperform the simplest versions of the royal road functions and other search algorithms. Thus, it is key to performance and should not be ignored. In [11], Jones defines BONUS as a way to classify individuals into distinct levels (i.e., order of schema). The BONUS must be a distinctive value, i.e., from the overall fitness value we can distinguish between individuals from different order of schemata. Thus, BONUS must be greater than the maximum value that an individual can have from the sum of all PARTs. Because we define PART by a normalized distance that is always upper-bounded by 1, BONUS can be defined by the number of branches. To take into account the branch-hardness, we use the Difficulty Level (DL) metric [14] to reward an optimal building block.

In general an evolutionary testing fitness function is a minimization function, so the BONUS of each block must decrease its objective. By default, the objective value must contain the maximum summed BONUS that may appear in an objective value and when a building block reaches its optimal value, BONUS is decreased. Because the optimal fitness value of a building block is equal to 0, a simple equivalent expression may be obtained by penalizing non-optimal building blocks instead of rewarding optimal building blocks. Therefore, an unsatisfied branch is penalized by its BONUS, so a minimization royal road function for evolutionary testing can be expressed as follows:

$$f_{RR}(i, C) = \sum_{c \in C'} \Big(BONUS(c) + PART(i, c) \Big)$$
$$= \sum_{c \in C'} \Big(DL(c, C) + \eta\big(\delta \cdot DC(c)\big) \Big)$$

Where C is the set of branches whereon the test target is control dependent; C' is the subset of C that contains unsatisfied branches by i; DL(c,C) and DC(c) as are defined in [14].

In evolutionary testing, the level of dependency between building blocks may be different from one instance to another. An instance may contain a set of strongly dependent building blocks, while another may contain a set of building blocks that are weakly dependent or totally independent. Because a hierarchical structure must take into account the conflicts among schemata, the inter-dependency between building blocks, and intra-dependency within building blocks [15], we cannot enforce a general subset of schemata (i.e., a plan to follow) for all instances of the problem of test-data generation.

The proposed formulation of the fitness function does not predict an exact subset of schemata. This is an advantage because it can deal with any instance of the problem of test-data generation, but, at the same time, this is an inconvenience because its landscape may significantly change from one instance to another. Therefore, as any general fitness function, the proposed one may be deceptive in some instances of evolutionary testing. The fitness function alone cannot perform well on every instance of evolutionary testing. To further increase its power, in addition to a good fitness function, GA must exploit features associated with good performance [1]. In Holland's words "unexploited possibilities may contain the key to optimal performance, dooming the system to fruitless search until they are implemented. There is only one insurance against these contingencies. The adaptive system must, as an integral part of its search of "(schemata)", *persistently test and incorporate structures properties associated with better performance."*

Therefore, in the following subsections, we propose adaptive GA operators that use schema analysis to detect and exploit some features related to evolutionary testing performance.

2.2 Pairwise Selection Operator

The hope of a GA is finding the desired schemata at each generation. A distributed population on schemata may answer this hope. Because fitter individuals may be from a same schema, the selection operator must balance between fitness, diversity, and schemata. In the literature, there are several selection operators that balance between fitness and diversity [16]. *Pairwise-Selection* is a selection strategy that we propose to boost the selection by taking into consideration the diversity in terms of schemata by identifying schemata and distinguishing between them. From the overall fitness value, it may be possible to know the order of schemata, however it is impossible to identify an exact schema because many schemata may have the same order.

In evolutionary testing, we know a priori the optimal value of a fitness function (generally branch distance is equal to 0), whence the schemata identification is possible by storing the individual

```
1  foo(int i,int j,int k,int l ) {
2      int x=0;
3      if(i>0)      //b1
4          x=j;
5      if (x>0 && k>0)  //b2
6          if(l>0)    //b3
7              return 1;
8
9    return 0;
10 }
```

Fig. 1: foo function

	b_1	b_2	b_3
i	1	1	0
j	0	1	0
k	0	1	0
l	0	0	1

Fig. 2: influence matrix for foo function

fitness value of each building block (branch distance of each individual branch) and using them during the GA evolution phase. A schema detector can be defined as a binary vector derived from the schema pattern by replacing the asterisks by 0 and constraints by 1. The scalar product of a branch distance vector and detector vector is equal to 0 if and only if the individual is an instance of the schema. Further, the branch distance vectors may be a good mean to measure the similarity or the distance between individuals in terms of schemata. The *similarity* between two individuals is equal to the scalar product of their branch distance vectors. The similarity is maximal if the two individuals are instances of the same schema, it is minimal if the two individuals complement each other (it is equal to 0).

The pairwise selection combines two selection strategies to select individuals in a pairwise manner. The first individual is selected according to a selection strategy that takes in consideration the fitness and the population diversity. The second individual is selected using a selection strategy that focuses on the diversity in terms of schemata. The first strategy sorts individuals according to a linear ranking method [17, 18], after which the first individual is selected using stochastic universal sampling [18, 19]. The second individual is selected to increase the likelihood of getting a complement of the first individual using a tournament selection [16] based on lowest similarity and worst fitness in case of ties.

Pairwise selection has a good likelihood of diversifying the population in terms of schemata. Furthermore, it couples two individuals that have a potential to generate fitter offspring.

2.3 Crossover and mutation Operators

It is well known that crossover and mutation operators disrupt schemata in a population. The schema theorem shows that crossover and mutation rates are the main source of the loss of schemata [1]. Those rates are constant in evolutionary testing and are randomly applied without any consideration for schemata. Adaptive rates may help to reduce losses and make evolutionary testing more effective and efficient. Because crossover and mutation operators perform on genes, comparison in terms of genes' performances is a natural way to define adaptive crossover and mutation operator. The contribution of each gene to the fitness may be an adequate indicator to adapt those two operators. We propose a schema analysis to identify the influence of each input variable on each schema.

We say that an input variable v influences a schema s if v influences any basic building-block component in s. Because a basic building block is defined by an individual branch, we keep Korel's definition of influence [20] and we extend it with *indirect influence*. A variable v indirectly influences a branch b if b is influenced by a variable w while a definition of w is in an execution block B and the execution of B depends on v value. Fig. 1 is a sample source code wherein the input variable i indirectly influences the conditional statement at Line 5.

The schema analysis generates an *influence matrix* that summarizes input variables influence on basic building blocks. An *influence matrix* is a UUT matrix that reflects the dependency relationship between conditional statements and input variables: its lines are input variables and its columns are conditional statements. A cell $[v, c]$ of an influence matrix is equal to 1 if the conditional statement c depends in any way on input variable v, else it is equal to 0. Fig. 2 presents the influence matrix for `foo` function.

The schema analysis is a sort of data flow analysis. With a slight modification to a data flow analysis algorithm, we can generate the influence matrix. In this paper, we assume that the control

flow graph does not contain cycles (loops are unfolded *kpath* times) and use an adapted version of the Basic Reach Algorithm [21] to generate the influence matrix.

Using the influence matrix, an estimated input variable contribution can be determined in each of the branch fitness. We assume that input variables involved in a branch contribute equitably to its fitness value. Therefore, a fitness value can be assigned to each input variable by summing its contributions among all branches involved in an influence relationship.

Adaptive Crossover Operator. The aim of any crossover operator is generating at least one offspring which is fitter than each parent. To achieve this objective a crossover operator needs to detect the origin of the fitness in each parent and exploit it to get fitter offspring. An adaptive uniform operator can be developed for this purpose that focuses only on one offspring wherein the fitter genes are recombined. The adaptive uniform operator generates a main offspring and a secondary offspring. The main offspring receives from the first parent all genes that have reached the optimal fitness value then the remaining genes are chosen from both parents according to fitness values of genes (i.e., the fitter gene of both genes is chosen). The secondary offspring receives each gene from one parent with an equal probability. Despite genes being linked by building blocks and the latter not being necessary independent, this adaptive uniform crossover has a good likelihood of making a fitter offspring and reducing the probability of losing schemata.

Adaptive Mutation Operator. Two features define a mutation operator: the probability of changing a gene and the way a gene is changed. A mutation operation can be considered successful if the new offspring is fitter than its parent. To achieve this a mutation operator needs to detect the sources of weakness from the fitness of the parent and change them in an adequate way. An adaptive mutation operator can be developed to answer this. Instead of using an equal probability $1/\ell$ to change each gene, the adaptive mutation operator uses a probability computed in terms of the fitness value scored by each gene. It makes a distinction between two groups of genes: optimal genes that have the optimal fitness value (0) and non-optimal genes that do not yet reach the optimal fitness value. Only genes from the second group are subject to mutation with an equal probability $\frac{1}{|Sg|}$, where $|Sg|$ is the number of genes in the second group (non-optimal). Thus a mutated value is computed in terms of the min and the max of fitness values of branches (building block) wherein the input variable (gene) is involved. A value r is randomly selected from one of three intervals $\{[1, min], [min+1, max], [max+1, uBound]\}$ with probabilities $\{0.6, 0.3, 0.1\}$, then this value is added or subtracted to the current gene value. A minimum width is required for each interval (e.g., for integers each interval must contain at least 50 values otherwise the interval is modified to contain this number starting from its lower bound).

3 Empirical Study

The main goal of our empirical study is to compare our proposed enhanced GA framework against a simple GA and to analyze its impact in terms of effectiveness and efficiency on evolutionary testing. Evolutionary testing is considered more *efficient* if the enhanced GA framework reduces the number of evaluations to achieve a given coverage; it is considered more *effective* if the enhanced GA framework covers more targets. To get a more revealing analysis, we evaluate separately the four components in our enhanced GA framework: the royal road function of evolutionary testing, the pairwise selection, the adaptive crossover operator, and the adaptive mutation operator. to determine the contribution of each component in the enhanced GA framework and to analyze the following questions:

- **(RQ1): Validation of evolutionary testing royal road function.** Can an evolutionary fitness function expressed according to the schema theory make evolutionary testing more efficient or effective? This shows the impact of a fitness function expressed according to the schema theory;
- **(RQ2): Validation of schemata diversity hypothesis.** Can the pairwise selection make evolutionary testing more efficient or effective? This shows the impact of maintaining the population diversity in terms of schemata;

– **(RQ3): Performance of proposed adaptive crossover.** Can the adaptive crossover operator that combines fitter genes enhance evolutionary testing in terms of efficiency of effectiveness? This shows the effect of preserving schemata by exploiting the sources of performance;
– **(RQ4): Performance of proposed adaptive mutation.** Can the adaptive mutation operator that assigns high probability to "weaker" genes enhance evolutionary testing in terms of efficiency of effectiveness? This shows the impact of incorporating structures and properties associated with weaker performance to preserve schemata and mutate individuals.

To ensure that we compare, indeed, with a good implementation od simple GA a careful implementation of the DaimlerChrysler system for evolutionary testing (DCS-ET) [6, 7] was set up by using the evolutionary computation for Java (ECJ) system [22]. We choose ECJ because it is founded on the same principles as DCS-ET: both systems support breeding based on multi-populations, or Breeder Genetic Algorithm [23]. The ECJ features make reimplementing DCS-ET easy. DCS-ET has been largely studied and reimplemented in the literature and it is always considered as the state of the art [5]. In the next paragraph we give a brief description of DCS-ET [5, 7].

The population contains 300 individuals that are uniformly distributed over six sub-populations. All sub-populations are evolved in parallel separately according to the following process: (1) Evaluation: according to a fitness function, a score is assigned to each individual; (2) Selection: individuals are selected using a linear ranking [17, 18] with a selection pressure equal to 1.7 followed by stochastic universal selection [18, 19]; (3) Crossover: parents are recombined by using a discrete recombination [23]; (4) Mutation: offsprings are mutated by the breeder genetic algorithm mutation strategy [23]; (5) Reinsertion: an elitist reinsertion strategy keeps the top 10% of the current population; (6) Random Exchange: every 20 generations sub-populations randomly exchange 10% of their individuals with one another; (7) Populations competition: every four generations, the populations are ranked according to their progress in terms of performance (mean fitness) and the size of each sub-population is updated with weaker sub-populations losing individuals in favor of stronger ones.

We enhanced DCS-ET with f_{RR}, pairwise selection, adaptive crossover, and adaptive mutation operators to get a first version of our framework EGAF-ET.

The study was performed on 440 synthetic test targets that were randomly generated. A synthetic test target is a simple Java program that contains a set of nested branches to be satisfied (i.e., only the deeper branch is targeted). The search space and the number of nested branches are the main factors in determining the difficulty of an evolutionary testing problem. The search space grows exponentially with the number of input variables. To get simple Java programs that represent different levels of difficulty the number of input variables is varied from 2 to 9 and each test target contains at least a number of nested branches equal to the number of input variables and at most is equal to double the number of input variables. For each combination of the number of input variables and number of nested branches 10 simple Java programs are generated. We define each input variable as an integer that takes its value from the large domain $[2^{-30}, 2^{30}]$. Each simple Java program denotes a separate search problem for which the size of the search space approximately ranges from 2^{62} to 2^{279}. To get realistic test targets, every test target is generated carefully, branch by branch. Every branch must: 1) keep the test target feasible; 2) involve two input variables (80%) or an input variable and a constant (20%); 3) not be implied by the current test target.[1]

Each search for test data that meets a test target was performed 20 times. This repetition allows reducing the random aspect in the observed values. A set of 20 random number seeds was used to seed the random number generators. If test data was not found after 30,000 fitness evaluations for both DCS-ET and EGAF-ET, the search was terminated.

Table 2 summarizes the results of 8,800 executions for each combination of framework, BONUS, and PART. An execution is considered successful if it could generate a test datum. To compute a maximum coverage (Max. Cov.), a branch is considered covered if test data were found during at least one of the 20 executions. To compute a average coverage (Ave. Cov.), a branch is considered covered if test data were found during at least teen of the 20 executions. The f_{RR} with PART equals to the adjusted branch distance ($\delta \cdot DC$) outperforms f_{RR} with the simple branch distance (δ). Thus in our framework (EGAF-ET) f_{RR} with BONUS different from zero works better than with

[1] Benchmarks are available at `http://www.crt.umontreal.ca/~quosseca/fichiers/23-Meta16Benchs.zip`

Table 2: Results of different expressions of the evolutionary testing royal road function on both frameworks.

f_{RR}		SUCCESS		EVALUATIONS		Max. Cov.		Ave. Cov.	
BONUS	PART	DCS	EGAF	DCS	EGAF	DCS	EGAF	DCS	EGAF
0	$\eta(\delta)$	5,084	8,511	175,619,700	43,269,900	78.18	99.77	58.63	97.27
	$\eta(\delta \cdot DC)$	5,262	8,521	173,914,800	42,747,300	80.45	99.09	60.00	97.72
$\mid C \mid$	$\eta(\delta)$	5,048	8,515	175,827,000	43,021,200	77.72	99.54	56.59	97.27
	$\eta(\delta \cdot DC)$	5,248	8,528	173,948,400	42,644,100	78.86	99.77	58.86	97.50
DL	$\eta(\delta)$	5,037	8,547	176,110,800	41,990,700	76.59	100.00	58.18	97.72
	$\eta(\delta \cdot DC)$	5,166	8,556	175,356,000	41,851,500	79.09	100.00	59.31	97.72

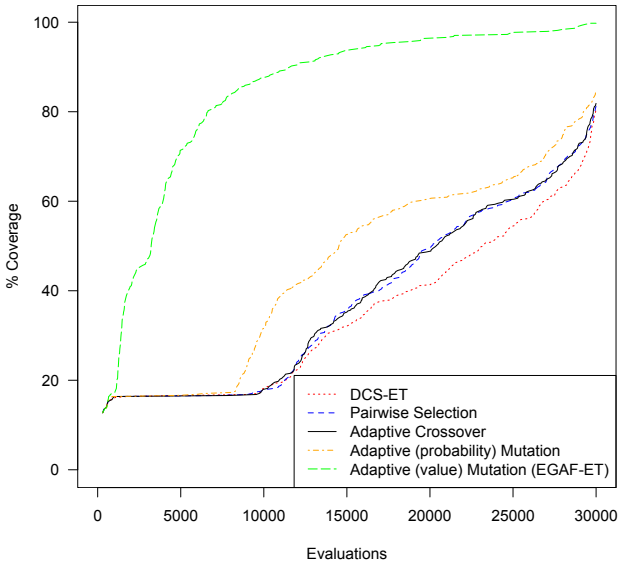


Fig. 3: The branch coverage in terms of the average number of fitness evaluations.

BONUS equal to 0. Further f_{RR} with DL is better than with $\mid C \mid$. Contrary to our framework on DCS-ET the f_{RR} with DL or $\mid C \mid$ slightly decreases the performance because the f_{RR} with BONUS different from 0 favors higher-order schemata and these latter overlapped, but DCS-ET uses simple crossover and mutation operators that disrupt schemata. Thus, **we answer RQ1 by claiming that the evolutionary testing royal road function makes evolutionary testing more effective and efficient if the schemata are preserved and their diversity is maintained**.

To measure the advantage of each proposed component the framework DCS-ET is modified step by step. First we replaced DCS-ET's selection strategy by our pairwise selection, then we changed the crossover operator to our adaptive one. After that we modified the mutation rate of DCS-ET by making it adaptive. Finally we replaced the mutation way by our proposed one. Figure 3 summarizes different changing steps that we did to pass from DCS-ET to EGAF-ET.

For our benchmark set of simple programs, in Table 2 and Figure 3, our framework significantly outperforms the DCS-ET framework both for branch coverage and number of fitness evaluations required to cover a same number of branches. Also, each proposed operator alone performs better than its simple counterpart. The pairwise selection could not show a significant enhancement in terms of branch coverage, but it is distinctly better than the selection strategy used by DCS-ET in terms of number of evaluations. The pairwise selection offers to the crossover operator a pair of individuals that complement each other. According to the results the complementary between the crossed individuals may diversify schemata and preserve them. Therefore, **we answer RQ2 by claiming that the pairwise selection maintains the population diversity in terms of schemata, thereby making evolutionary testing more efficient but not more effective**.

The proposed adaptive crossover does not show a meaningful difference compared to the discrete crossover because of the complicated relationship between schemata: the latter overlap and strongly

depend on each other. The dependency between schemata preserves the combination of fitter genes to generate fitter schemata, maybe due to the kind of UUTs tried, which essentially use integer data types and their branches are highly dependent on each other. More evidence is needed to draw a general conclusion about the proposed crossover. Thus, on the selected UUTs and the proposed crossover operator and **we answer RQ3 by claiming that combining fitter genes does not necessary generate fitter schemata**.

In the adaptive mutation operator, both the adaptive mutation rate (probability) and the adaptive mutation strategy (value) significantly enhance the performance in terms of fitness evaluations and branch coverage. The adaptive mutation rate orients the mutation operator on the "weaker" genes by assigning them a high mutation rate, then allows it to outperform its counterpart that uses equal probability regardless of gene performance. Thus, the adaptive mutation strategy orients the mutation operator to choose potential values by using the minimum and the maximum branch distances while favoring the neighborhood. This mutation strategy outperforms the breeder mutation operator in terms of branch coverage and fitness evaluations. Therefore, **we answer RQ4 by claiming that incorporating structures and properties associated with weaker or better performance to preserve schemata and mutate individuals significantly enhances evolutionary testing**.

4 Related Work

The last two decades have witnessed an increased interest concerning search-based test data generation, especially evolutionary testing. Many authors have considered genetic algorithms as potential search algorithms for test data generation [2–7]. These approaches keep the general concept of GA and focus only on two components: the representation of the feasible solutions and the fitness function. This makes the GA operators problem independent and those approaches very general. Our framework is different because in addition to these we propose to make GA operators problem dependent by taking into consideration the structure of test-data problem (schemata, branches and input variables performance) during the evolution phase.

Korel proposed a dynamic data flow analysis approach [20] for path coverage that may make the GA evolution phase problem dependent. His approach consists of analysing the influences of each input variable on the successful sub-path traversed during the program execution. Only input variables that influence the successful sub-path are subject of changing their values. Korel's approach does not take into account the non-executed sub-path and input variables that statically have an influence on the test target. However input variables that statically influence a test target in the whole path (executed or non-executed sub-path) may contain the key to reach a test target. Harman et al. [24] studied the impact of search space reduction on search based test data generation. In their study for a given test data generation problem (branch) they use the static analysis approach described in [25, 26] to remove irrelevant input variables form the search space. This approach ignores input variable value performance, So in a given context (individual) it cannot distinguish between relevant and irrelevant input variables, i.e., a input variables may be relevant for a given individual and irrelevant for another one "blindly" changing its value may decrease performances because it may be a component of a schema. Our approach is different from Korel and Harman et al. because it is based on schema theory and combines static and dynamic analysis to determine relevant and irrelevant input variables for each individual according to a test target.

An evolutionary testing schema theory has been previously defined and analysed by Harman and McMinn [5]. They define an evolutionary testing schema as a set of chromosomes that satisfy a constraint (branch) and its order is the number of variables involved (arity of the constraint). According to the schema theory, this means that a schema defined on a small number of variables (lower-order schema) is less fit than a schema defined on a larger number of variables (higher-order schema), thus the recombination of chromosomes satisfying lower-order schema builds chromosomes satisfying higher-order schema. Since they do not define a fitness function and there is none in the literature that reflects their schema concept their definition is useful only on some particular cases of branches and it is hard to be automated. As we claim earlier in this paper, it is more productive to model schemata for a class of problems while linking it with a fitness function. Our evolutionary testing schema definition is more general and it is reflected by a royal road function. In our definition the order of schema is the number of individual conditions, so any nested branch is a potential evolutionary testing problem wherein the schema theory may be applied.

Despite the large body of work on evolutionary testing, we believe this is the first paper to provide a complete evolutionary testing GA framework based on schema theory. Our framework rational differs from previous work in that it provides a new concept of fitness function and all fundamental GA operators.

5 Conclusion

In the last two decades, search-based testing and in particular evolutionary testing have been extensively applied to solve the problem of automated test data generation. However despite the importance of the schema theory analysis for GA, its application to evolutionary testing has posed many challenges. To establish a GA framework based on schema theory analysis, this paper provides a novel automated evolutionary testing framework. We have (1) adapted schema theory for evolutionary testing, (2) defined a royal road fitness function for evolutionary testing, (3) proposed a new selection strategy called pairwise selection that combines two selection strategies to maintain the population diversity in terms of schemata, (4) proposed an adaptive crossover that dynamically changes its rate according to the schemata, and (5) proposed an adaptive mutation operator that dynamically changes its rate and the way it mutates in terms of the schemata and their performances. We carried preliminary experiments to compare our framework to DaimlerChryster system on some randomly generated benchmark programs. A preliminary case study was carried out to frame the research questions. Results indicate that (1) the proposed royal road function can make evolutionary testing more effective and efficient if the schemata are preserved and their diversity is maintained; (2) the pairwise selection is able to maintain the population diversity in terms of schemata, thereby making evolutionary testing more efficient but not more effective; (3) the proposed adaptive crossover does not show a meaningful difference comparing to the discrete crossover, thereby combining fitter genes does not necessary generate fitter schemata; (4) incorporating structures and properties associated with weaker or better performance to preserve schemata and mutate individuals can significantly enhance evolutionary testing. Further the results obtained are promising but experiments with real world programs must be performed to provide additional empirical evidence on the potential of our proposed evolutionary testing framework. In the future we will focus on integrating the proposed framework in our tool JTExpert [27].

References

1. Holland, J.H.: Adaptation in natural and artificial systems, university of michigan press. Ann Arbor, MI **1** (1975) 5
2. Michael, C., McGraw, G., Schatz, M.: Generating software test data by evolution. Software Engineering, IEEE Transactions on **27** (2001) 1085–1110
3. McMinn, P., Harman, M., Binkley, D., Tonella, P.: The species per path approach to searchbased test data generation. In: Proceedings of the 2006 international symposium on Software testing and analysis. ISSTA '06, New York, NY, USA, ACM (2006) 13–24
4. Tracey, N., Clark, J., Mander, K., McDermid, J.: Automated test-data generation for exception conditions. Software-Practice and Experience **30** (2000) 61–79
5. Harman, M., McMinn, P.: A theoretical and empirical study of search-based testing: Local, global, and hybrid search. Software Engineering, IEEE Transactions on **36** (2010) 226–247
6. Baresel, A., Sthamer, H., Schmidt, M.: Fitness function design to improve evolutionary structural testing. In: Proceedings of the Genetic and Evolutionary Computation Conference. (2002) 1329–1336
7. Wegener, J., Baresel, A., Sthamer, H.: Evolutionary test environment for automatic structural testing. Information and Software Technology **43** (2001) 841 – 854
8. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley (1989)
9. Mitchell, M., Holland, J.H., Forrest, S., et al.: When will a genetic algorithm outperform hill climbing? Advances in neural information processing systems (1994) 51–51
10. Mitchell, M., Forrest, S., Holland, J.H.: The royal road for genetic algorithms: Fitness landscapes and ga performance. In: Proceedings of the first european conference on artificial life, Cambridge: The MIT Press (1992) 245–254
11. Jones, T.: A description of hollandś royal road function. Evol. Comput. **2** (1994) 409–415
12. Arcuri, A.: It does matter how you normalise the branch distance in search based software testing. In: ICST, IEEE Computer Society (2010) 205–214

13. Baars, A.I., Harman, M., Hassoun, Y., Lakhotia, K., McMinn, P., Tonella, P., Vos, T.E.J.: Symbolic search-based testing. In Alexander, P., Păsăreanu, C.S., Hosking, J.G., eds.: ASE, IEEE (2011) 53–62
14. Sakti, A., Guéhéneuc, Y., Pesant, G.: Constraint-based fitness function for search-based software testing. In Gomes, C.P., Sellmann, M., eds.: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 10th International Conference, CPAIOR 2013, Yorktown Heights, NY, USA, May 18-22, 2013. Proceedings. Volume 7874 of Lecture Notes in Computer Science., Springer (2013) 378–385
15. Watson, R.A., Hornby, G.S., Pollack, J.B.: Modeling building-block interdependency. In: Parallel Problem Solving from Nature-PPSN V. Volume 1498 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1998) 97–106
16. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. Urbana **51** (1991) 61801–2996
17. Whitley, L.D.: The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In Schaffer, J.D., ed.: ICGA, Morgan Kaufmann (1989) 116–123
18. Pohlheim, H.: GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB Documentation. `http://www.geatbx.com/docu/index.html` (2006) [Online; accessed 10-Feb-2013].
19. Baker, J.E.: Reducing bias and inefficienry in the selection algorithm. In Grefenstette, J.J., ed.: ICGA, Lawrence Erlbaum Associates (1987) 14–21
20. Korel, B.: Automated software test data generation. Software Engineering, IEEE Transactions on **16** (1990) 870–879
21. Allen, F.E., Cocke, J.: A program data flow analysis procedure. Commun. ACM **19** (1976) 137–
22. Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Bassett, J., Hubley, R., Chircop, A.: Ecj: A java-based evolutionary computation research system (2010)
23. Mühlenbein, H., Schlierkamp-Voosen, D.: Predictive models for the breeder genetic algorithm i. continuous parameter optimization. Evolutionary computation **1** (1993) 25–49
24. Harman, M., Hassoun, Y., Lakhotia, K., McMinn, P., Wegener, J.: The impact of input domain reduction on search-based test data generation. In: Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. ESEC-FSE '07, New York, NY, USA, ACM (2007) 155–164
25. Harman, M., Fox, C., Hierons, R., Hu, L., Danicic, S., Wegener, J.: Vada: a transformation-based system for variable dependence analysis. In: Source Code Analysis and Manipulation, 2002. Proceedings. Second IEEE International Workshop on. (2002) 55–64
26. Binkley, D., Harman, M.: Analysis and visualization of predicate dependence on formal parameters and global variables. Software Engineering, IEEE Transactions on **30** (2004) 715–735
27. Sakti, A., Pesant, G., Guéhéneuc, Y.G.: Instance generator and problem representation to improve object oriented code coverage. IEEE Transactions on Software Engineering **PP** (2014) 1–1

Automatically Generating Assessment Tests within Higher Education Context thanks to Genetic Approach

R. Ciguené[1], C. Joiron[2], and G. Dequen[2]

[1] Laboratoire SITERE, Ecole Supérieure d'Infotronique d'Haiti
richardson.ciguene@esih.edu
[2] Laboratoire MIS, Univ. de Picardie Jules Verne, 33 Rue Saint-Leu, 80039 Amiens Cedex 1
{celine.joiron, gilles.dequen}@u-picardie.fr

1 Introduction

The work partially described in this paper is at the crossroads of several research fields and specifically deals with higher educational learning assessments and Constraint Satisfaction Problems. The summative assessment [8, 1, 2] is done at the end of a curricula in order to evaluate learning outcomes [3]. When this type of assessment aims at delivering a certificate or a diploma, it is called *certificative*. Then a certificative assessment, which can be interpreted as a collection of tests, must guarantee fairness between all learners. However, teachers might have to create differentiated assessment *Tests* within various contexts : for distinct cohorts of learners, related to the same topics, and in various university sections or from a session to another; for the same cohort with multisession assessments; for a single assessment session with the aim to limit fraud (e.g. in a large amphitheater). As a consequence, the question is then : *How to guarantee fairness of a given assessment if we suppose it consists of distinct Tests ?* [3] Thus, we tackle in this paper the problem of assisting, thanks to an automatic process, the teacher in designing and generating assessments where each *Test* is structurally different from another. This work proposes a structural metric with the aim of characterizing the distance between two given *Tests*. This metric provides a dedicated fitness function that leads to define a genetic algorithm (GA) technique. We especially focus on Multiple Choice Questionnaires automatic generation. As a requirement, we suppose that the teacher has at his disposal a set of questions where each of them is coupled to a set of possible choices of answer (named *source database* in the following). The original use of GA allows to optimize this structural differentiation and thus guarantees the generation of collections of *Tests* with the largest distance possible while involving the smallest *source database*. The section 2 presents the structural distance between two *Tests* and defines its associated metric. The section 3 provides comparative experimental results about automatic generation of assessments using a genetic approach. Finally, section 4 gives some preliminary conclusions about this work and presents prospects.

2 About the structural distance : Dtest

A *Test* consists of an ordered sequence of *ItemTests* from the source database. Each *ItemTest* consists of one *EnonceItem*, the question statement, and an associated unordered set of *RepItems* from which possible choices will be extract during the generation process[4]. Each choice is semantically associated to a *true* or (exclusively) a *false* answer. If you associate each *EnonceItem* to a unique identifier, a *Test* can be likened as a word built on an alphabet. Drawing of this, related works on structural distance between words have been considered. As an example, Levenshtein proposed to estimate the distance between two strings by calculating the number of insertions, deletions and substitutions of characters that must be achieved in one of them to make it identical to the other [6, 5, 9].

2.1 From Levenshtein to structural distance between *Tests*

Practically Dtest is a score within the range $[0, 1]$ and represents the distance metric between a couple of *Tests* (t_1, t_2). From this range, 0 corresponds to $t_1 \equiv t_2$, meaning each *EnonceItem* from

[3] The notion of distinction between two *Tests* is then considered from a structural angle and as a first step, fairness of the assessments is out of the scope of this paper.
[4] associated sets are denoted respectively $EnonceItem_x$ and $RepItem_x^y$ for a *Test* x and an *ItemTest* y

t_1 belongs to t_2 at the same rank and is associated to the same set of *RepItems* with the same order. On the other hand, 1 is the maximal distance and corresponds to $t_1 \neq t_2$ meaning each *EnonceItem* from t_1 does not belong to t_2. Intermediate values represents a measure of 2 amounts applied to the sets of *EnonceItems* and *RepItems* respectively *Disparity* and *Permutation*. More precisely, Disparity measures the amount of distinct elements from two sets whereas Permutation expresses, within the framework of ordered sets, the mean rank difference between identical elements. Thus, $Disparity(a,b) = \frac{max(\#a,\#b)-\#(a\cap b)}{max(\#a,\#b)}$ and $Permutation(a,b) = \frac{\sum_{x\in(a\cap b)}|\mathcal{I}_x^a - \mathcal{I}_x^b|}{2\left\lfloor\frac{max(\#a,\#b)}{2}\right\rfloor\left\lceil\frac{max(\#a,\#b)}{2}\right\rceil}$ In both cases a and b are ordered sets and $\mathcal{I}_i^j$ denotes the rank of the element i in the set j. The figure 1 provides a schematic representation of the scale of our final metric which consists in combining 4 scores which are in order of relevance :

$$
\begin{array}{ll}
i) & \overline{cp(t_1,t_2)} = \forall_{r\in(EnonceItem_{t_1}\cap EnonceItem_{t_2})} Permutation(RepItem^r_{t_1}, RepItem^r_{t_1}) \\
ii) & \underline{qp(t_1,t_2)} = Permutation(EnonceItem_{t_1}, EnonceItem_{t_2}) \\
iii) & \overline{cd(t_1,t_2)} = \forall_{r\in(EnonceItem_{t_1}\cap EnonceItem_{t_2})} Disparity(RepItem^r_{t_1}, RepItem^r_{t_1}) \\
iv) & qd(t_1,t_2) = Disparity(EnonceItem_{t_1}, EnonceItem_{t_2})
\end{array}
$$

Thanks to this metric, we then are able to propose in the next section, thanks to a GA [7, 4], an original technique that generates collection of *Tests* with global distance (i.e. average distance of each *Test* couple) greater than randomization.

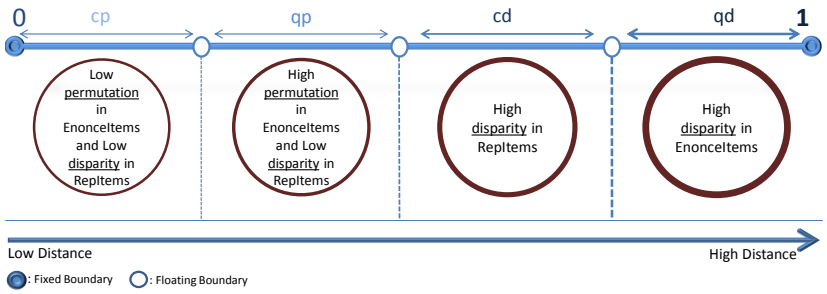


Fig. 1. Scale of the metric characterizing the distance between two *Tests*

3 Generating thanks to a genetic approach

The aim of our preliminar experimentation is to offer the highest structural distance as possible while allowing the teacher to imagine the smallest possible number of elements in the source database. To do so, our GA approach uses DTEST as a fitness function and is named GAwDTEST. Within the experimental context, we choose to work with a population of 1,000 individuals where each of them is initially a collection of *Tests* randomly sampled among those belonging to the source database. Following preliminary experimentations that we cannot detail in this paper, we empirically set the percentage of deletion and mutation to 8% and 3% respectively. We consider 1,000 generations or a timeout at 3,600 seconds. Some prior experiments lead us to consider the ratio $\frac{\#ItemTest \in Source\ Database}{\#ItemTest \in Test_i}$ as the relevant parameter correlated to the context of generation. In other words and as an illustration, this means that generating *Tests* with 10 (resp. 50) *ItemTests* chosen among a set of 20 (resp. 100) are identical contexts. When the ratio is high, this corresponds to a context where the teacher has a large amount of questions at his disposal so that building his assessment. Prior experiments show that in this case, a random selection results good enough structural distance according to DTEST. On the other hand, when the ratio is closed to 1.00, random generation is not sufficient to provide a great as differentiation between the generated *Tests* as the size of the collection grows[5]. Consequently, we choose to focus only on small ratios {1, 1.2, 1.5, 1.7, 2}. Figure 2 gives some details on the experimental contribution of GAwDTEST in comparison with random selection. Left side of the figure 2 shows comparative mean distances

[5] note: ratio equals 1.00 is only a shuffle on the source database

between random selection and GAwDTEST according to DTEST. For instance this guarantees to the teacher that if he has at disposal n question in the database to build Tests of n questions (i.e. ratio = 1.00), GAwDTEST enhances the classical random shuffle generation, according to the global metric DTEST, by 10%. Right side of figure 2 provides the distribution of DTEST metric for each couple of *Tests* belonging to a given assessment at a ratio 1.00. According to our scale (see figure 1), and wit[illegible] a high disparity in *EnonceItems* v

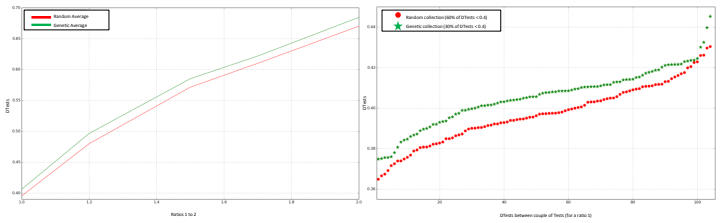


Fig. 2. Comparative results between a random sample and selected collection from GAwDTEST for ratios from 1.00 to 2.00

4 Conclusion

This paper has presented an original approach that can assist the teacher in automatically generating collections of *Tests* through MCQ, specifically within the context of certificative assessments. As a contribution, we propose an original metric partially inspired from Levenshtein distance[6, 9] that characterizes the difference between two *Tests* of a given collection. Since this metric can be used as a fitness function, we adapted a genetic approach to automatically generate collections that maximizes the distance between each couple of *Tests* of a given collection and that even if the teacher has few patterns in its source database. This approach, GAwDTEST, can be compared to a random selection and allows some substantial improvements on a difficult problem. GAwDTEST improvements are more significant to generate small collections. Analysis of such a fact cannot be developed within this paper. Nevertheless this leads us to consider a new parameter (i.e. constraint) during generation process : Topology of the classroom. Limiting fraud within this context consists in considering that each learner must have a (very) distant *Test* from his entire neighborhood. This problem can be partially understood as an allocation problem. One of our future work, in addition to handling fairness, will consider this new constraint.

References

1. Catherine Garrison and Michael Ehringhaus. Formative and summative assessments in the classroom, 2007.
2. Wynne Harlen. On the relationship between assessment for formative and summative purposes. *Assessment and learning*, pages 103–118, 2006.
3. Wynne Harlen and Mary James. Assessment and learning: differences and relationships between formative and summative assessment. *Assessment in Education*, 4(3):365–379, 1997.
4. Randy L Haupt and Sue Ellen Haupt. *Practical genetic algorithms*. John Wiley & Sons, 2004.
5. Wilbert Jan Heeringa. *Measuring dialect pronunciation differences using Levenshtein distance*. PhD thesis, Citeseer, 2004.
6. V. I. Levenshtein. *Binary codes capable of correcting deletions, insertions, and reversals.*, volume 10. February 1966.
7. Gen Mitsuo and Cheng Runwei. *Genetic algorithms and engineering optimization*, volume 7. John Wiley and Sons, 2000.
8. Maddalena Taras. Assessment–summative and formative–some theoretical reflections. *British Journal of Educational Studies*, 53(4):466–478, 2005.
9. Li Yujian and Liu Bo. A normalized levenshtein distance metric. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1091–1095, 2007.

Genetic Algorithm to Improve Diversity in MDE

Florian Galinier, Éric Bourreau, Annie Chateau, Adel Ferdjoukh, Clémentine Nebut

LIRMM, Université de Montpellier and CNRS, Montpellier, France
{lastname}@lirmm.fr

1 Introduction

Model Driven Engineering (MDE) provides tools for systematization and automation of models. Model transformations are nowadays widely used either in a maintenance context or in a more traditional development contexts. It allows realization at many stages during life cycle for a software: code generation, reverse engineering, etc. These models transformations are for example a way to change a model from a specific domain – *e.g.* a UML class model – to another domain – *e.g.* a Java model. Domains of models are specified by models of higher abstraction level: Meta-Models (MM).

Generation of models from MM as a testing purpose is a quite complex task. Several combinatorial approaches have already been proposed. In [1–3], the authors propose a generation based on Constraint Satisfaction Problems (CSP). These methods provide encouraging results, creating models that conform to meta-models and moreover, encoding constraints (OCL) that can be added to meta-models. Recently, in [4], simulated annealing generate a **set of models** which are the more dissimilar and which cover the meta-model. Meta-heuristics seem to provide interesting results in term of diversity, and, in our opinion, diversity is now a major issue to complete our generation process.

In this paper, we explain how to generate diverse models conform to a meta-model, using genetic algorithms.

2 Evolutionary algorithm

Evolutionary algorithm (EA) have been largely adopted to provide good solutions to combinatorial problems. In this section we will describe the differents components of our EA [5].

2.1 Chromosome representation

Representing models as a sequence in a chromosome is complex, moreover if we want to allow the possibility to crossover models. We base our approach on the CSP paradigm used by the authors in [1]. In this work, Ferdjoukh et al. propose to describe a model as a constraint network composed by variables, domains associated to these variables and a set of constraints. A valid model (*i.e.* which respect the meta-model and constraints) is an instantiation of variables that respect CSP constraints. We first choose to use vectors of instanciated values as chromosome representation. In a second time, we prefer to express in a chromosome a set of models (a list of size p) by concatenating p consecutive vectors representing p different models.

2.2 Initial Population

In [6], Ferdjoukh et al. introduce probabilistic simulation to generate realistic datas from a MM, based on a CSP instantiation. Their tool (named Grimm) is able to generate a set of valid models, conforms to meta-models, with variety in domain range instantiation due to probability distribution. A first set of 100 models, conforms to a specific meta model, is generated. Due to the value of p, we respectively considered 50 chromosomes (p=2), 25 chromosomes (p=4) or 20 (p=5).

2.3 Fitness function

Diversity between models is a rich problem. Indeed, it could be linked to a metrix representing similarity/dissimilarity. Some previous works in model driven engineering [7] used semantics data of models to provide a similarity measure. In our case, semantics data seems not sufficiently realistic, models are generated. Instead, we choose to focus on structure to define fitness.

As previously explained we reduced the problem by representing a graph as a vector including for each artefact target, relations and attributes. A simple Hamming distance or Levenshtein distance can be then used to compared two models. We choose to used a more recent metric from Data Mining domain: the cosine dissimilarity [8]. This metric is usualy applied to vectors that represent occurences of word in a text. It is defined as:

$$D_C(\overrightarrow{V_1}, \overrightarrow{V_2}) = 1 - S_C(\overrightarrow{V_1}, \overrightarrow{V_2}) \tag{1}$$

with $S_C(\overrightarrow{V_1}, \overrightarrow{V_2})$ the cosine similarity of two vectors $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ defined as :

$$S_C(\overrightarrow{V_1}, \overrightarrow{V_2}) = \frac{\overrightarrow{V_1} \cdot \overrightarrow{V_2}}{\|\overrightarrow{V_1}\|\|\overrightarrow{V_2}\|} = \frac{\sum_{i=1}^{n} V_1[i]V_2[i]}{\sqrt{\sum_{i=1}^{n} V_1[i]^2}\sqrt{\sum_{i=1}^{n} V_2[i]^2}} \tag{2}$$

For the simple case where one chromosome contain two models, this distance is quite easy to compute, but in other cases, we obtain a distance matrix. We follow different approaches to try to increase diversity. Indeed, we consider several fitness for the "more than two models" case:

(1) a fitness based on the *lowest* value of the distance matrix: with this approach, we consider that increase the minimal value will normally increase all the distances;
(2) a fitness based on *mean* value of the distance matrix: with this approach, we hope that trying to increase the mean will increase all the distances;
(3) a fitness based on a *sum* of (1) and (2), trying to increase both values;
(4) two different fitness functions: the (1) and the (2), trying to increase one of the two values.

If the three first approaches can be realized with all genetics algorithms, the last approach need the use of a multi-objective evolutionary algorithm. We choose the widely used *NSGA-II* [9, 10].

2.4 Crossover, mutation, feasibility repair and selection

We randomly pick couples of chromosomes and apply a simple one point crossover on them to generate new chromosomes.

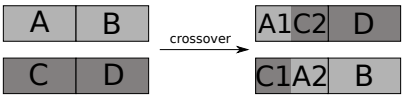


Fig. 1. One point Crossover between two chromosomes composed by 2 models.

We determine that a mutation chance of 0.5‰ for each gene is enough to avoid a too quick convergence and allow diversification. To check feasibility, first we store extra-data, like domains and constraints, to quickly check if new vectors (generated by crossover or mutation) are still valid. Our modified vector is reintroduced in CSP tool to check satisfiability and if necessary, repair it by looking valid instantiation in a classical depth first search exploration.

As selection criterion, when we generating 100 new chromosomes, we keep the 100 best over the overall 200 chromosomes

3 Computational results

We first consider a population with individual composed only by 2 models. In Fig. 2, we can observe that our method provides better results than the CSP approach. Indeed, population at generation 100 has a mean distance score better than the original population. Moreover, we can note that for both distances the lowest value of our final population is better than the higher value of initial one. The lowest values for cosine distance are inherent to the fact that some values of vectors are

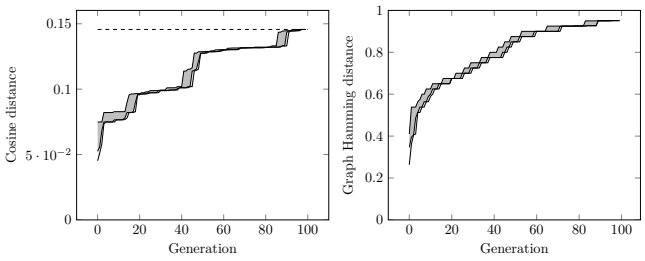


Fig. 2. Evolution of mean distance for population with chromosome with 2 models

too constrained – the dashed line indicates the best possible pairing at last generation, computed with a brute force algorithm.

Fig. 3 shows the different fitness approaches when $p = 5$. Trying min value (1) or mean value (2) brings quite similar results. The combination of two objectives, with sum aggregation (3) and even more with bi-objective vision (4), provides even better results. We also observe that our method provides better results than original one at the cost of an increased computation time to reach convergence.

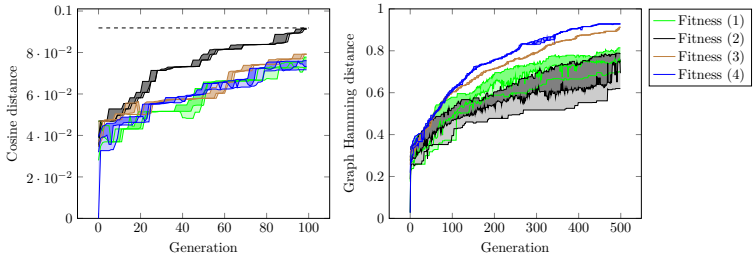


Fig. 3. Evolution of mean distance for population with chromosome composed by 5 models

4 Conclusion

Diversity of artefacts in model generation is an important point. The evolutionary meta-heuristic provide a tool to increase distance between models. With this approach, we are able to increase mean score for more than 50% for considered metrics. As perspectives, we will look at the case where chromosome are mono-model by considering a population fitness, and preliminaries computations seem to provide better results in terms of scalability.

References

1. A. Ferdjoukh, A.-E. Baert, E. Bourreau, A. Chateau, R. Coletta, and C. Nebut. Instantiation of Meta-models Constrained with OCL: a CSP Approach. In *MODELSWARD*, pages 213–222, 2015.
2. J. Cabot, R. Clarisó, and D. Riera. Verification of UML/OCL class diagrams using constraint programming. In *IEEE ICSTW Workshop*, pages 73–80, 2008.
3. C. A. González Pérez, F. Buettner, R. Clarisó, and J. Cabot. EMFtoCSP: A tool for the lightweight verification of EMF models. In *FormSERA*, 2012.
4. J. Cadavid, B. Baudry, and H. Sahraoui. Searching the Boundaries of a Modeling Space to Test Metamodels. In *ICST*, pages 131–140, 2012.
5. J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* The University of Michigan Press, 1975.
6. A. Ferdjoukh, E. Bourreau, A. Chateau, and C. Nebut. A Model-Driven Approach to Generate Relevant and Realistic Datasets. In *SEKE*, 2016.
7. J.-R. Falleri, M. Huchard, M. Lafourcade, and C. Nebut. Metamodel matching for automatic model transformation generation. In *MODELS*, pages 326–340, 2008.
8. A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43, 2001.
9. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
10. K. Deb and H. Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

$\mathcal{Q}$-learning strategies for action selection in the TESTAR automated testing tool

Anna I. Esparcia-Alcázar, Francisco Almenar, Mirella Martínez, Urko Rueda, and Tanja E.J. Vos

Research Center on Software Production Methods (PROS)
Universitat Politècnica de València
Camino de vera s/n 46022, Valencia, Spain
{falmenar,aesparcia,mmartinez,urueda,tvos}@pros.upv.es
http://www.testar.org

Abstract. TESTAR is an open source tool for automated software testing that generates test sequences on the fly based only on information derived from the Graphical User Interface (GUI). At the core of TESTAR is the way to automatically select which actions to test; finding the right algorithm to carry out this task can make significant differences to the testing outcome.
In this work we evaluate $\mathcal{Q}$-learning as a metaheuristic for action selection and carry out experiments with a range of paramenters, using random selection as a baseline for the comparison. Two applications are used as Software Undder Test (SUT) in the experiments, namely MS Powerpoint (a proprietary desktop application) and the Odoo enterprise management system (an open source web-based application). We introduce metrics to evaluate the performance of the testing with TESTAR, which are valid even under the assumption that access to the source code is not available and testing is only possible via the GUI. These metrics are used to perform statistical analysis, showing that the superiority of action selection by $\mathcal{Q}$-learning can only be achieved through an adequate choice of parameters.

Mots-Clefs. Automated GUI Testing,Testing Metrics, Testing Web Applications, $\mathcal{Q}$-learning

1 Introduction

The Graphical User Interface (GUI) represents a central point in any application from where the user may access all the functionality. Hence, testing at the GUI level means taking the user's perspective and is thus the ultimate way of verifying a program' s correct behaviour. Current GUIs can account for 45-60% of the entire source code [2] in any application and are often large and complex. Consequently, it is difficult to test applications thoroughly through their GUI, especially because GUIs are designed to be operated by humans, not machines. Moreover, they are inherently non-static interfaces, subject to constant change caused by functionality updates, usability enhancements, changing requirements or altered contexts. Automating the GUI testing process is therefore a crucial task in order to minimise time-consuming and tedious manual testing.

TESTAR is an open source tool that performs automated testing via the GUI, using the operating system's Accessibility API to recognise GUI controls and their properties, and enabling programmatic interaction with them. It derives sets of possible actions for each state the GUI is in and selects and executes appropriate ones, thus creating a test sequence on the fly. TESTAR has been successfully applied to various commercial and open source applications, both deskptop and web-based ones, as shown in e.g. [4, 6, 9, 10]; in most cases the action selection mechanism was random choice, a procedure also known as *monkey testing*

In citeBV2012,BV14 the first attempts to action selection in TESTAR based on metaheuristics, and especifically $\mathcal{Q}$-learning, are described. However, the performance metrics used for evaluation were the average time it took to crash the application under test and the reproducibility of the crashes. Although the results were promising, they revealed problems with this choice of metrics, which we try to address here. In this work we introduce four novel metrics specifically designed to testing via the GUI and without access to the source code of the applications. Using these metrics we compare various settings for $\mathcal{Q}$-learning and also use random testing as a baseline.

In order to carry out our study we chose two applications: the Odoo enterprise resource planning (ERP) system and the PowerPoint presentation software. They are two very different types of SUT: one is an open source web application and the other a proprietary desktop application.

We run experiments in three phases or iterations, refining the process after each phase, and carry out statistical analysis on the results of the third phase.

The rest of this paper is structured as follows. Section 3 describes the action selection mechanism using $\mathcal{Q}$-learning. Section 4 introduces the metrics used for quality assessment of the testing procedure. Section 5 summarises the experimental set up, the results obtained and the statistical analysis carried out; it also highlights the problems encountered. Finally, in section 6 we present some conclusions and outline areas for future work.

2 Related work

The existing literature in User Interface testing covers three approaches: *capture-and-replay* (C&R), which involves recording user interactions and converting them into a script that can be replayed repeatedly, *visual-based* which relies on image recognition techniques to visually interpret the images of the target UI [3] , and *traversal-based*, which uses information from the GUI (GUI reflection) to traverse it [1], and can be used to check some general properties.

Current practice of UI testing relies mainly on Capture-and-Replay (C&R, also called Record-and-Replay) tools. This is a mature technology, for which tools are widely available, be they commercial or open source. However, a major problem with this approach is maintenance, as changes in the UI usually render the created test scripts unusable. This problem becomes more severe with the new generation of Internet-based applications, as these adapt their layout dynamically according to the users' needs. Hence, in spite of some degree of automation, GUI testing still involves heavy load of manual work, which is costly and error prone [3].

Visual-based and traversal-based tools aim at solving the maintenance problem; the latter group, to which TESTAR belongs, is considered to be the most resilient to changes in the SUT.

In order to evaluate the quality and performance of the testing suitable metrics must be defined. For instance, in [7] Chaudhary et al propose metrics for event driven driven software. Memon et al [8] propose a coverage criteria for GUI testing. However, knowing what to measure is still an area that deserves further investigation. In this work we propose four such metrics which are suitable to measure the quality of the GUI testing on web applications, based on the assumption that source code is not available.

3 Using $\mathcal{Q}$-learning for action selection in TESTAR

The choice of an action selection mechanism is one of the two main inputs for the human tester in TESTAR (the other one being the custom protocol).

We have employed the $\mathcal{Q}$-learning algorithm to guide the action selection process. $\mathcal{Q}$-learning [11] is a model-free reinforcement learning technique in which an agent, at a state s, must choose one among a set of actions A_s available at that state. By performing an action $a \in A_s$, the agent can move from state to state. Executing an action in a specific state provides the agent with a reward (a numerical score which measures the utility of executing a given action in a given state). The goal of the agent is to maximise its total reward, since it allows the algorithm to look ahead when choosing actions to execute. It does this by learning which action is optimal for each state. The action that is optimal for each state is the action that has the highest long-term reward.

Our version of the $\mathcal{Q}$-learning algorithm, shown in Algorithm 1 is governed by two parameters: the maximum reward, R_{max} and the discount γ. Depending on how these are chosen the algorithm will promote exploration or exploitation of the search space. The R_{max} parameter determines the initial reward unexplored actions have; so, a high value biases the search towards executing unexplored actions. On the other hand, discount γ establishes how the reward of an action decreases after being executed. Small γ values decrease the reward faster and vice versa.

R is set as follows:

$$R(s,a,s') := \begin{cases} R_{max} & if\ x_a = 0 \\ \frac{1}{x_a} & otherwise \end{cases}$$

Algorithm 1 $\mathcal{Q}$-learning algorithm

Require: $R_{max} > 0$ /* *reward for unexecuted actions* */
Require: $0 < \gamma < 1$ /* *discount factor* */
1: **begin**
2: start SUT
3: $V(s, a) \leftarrow R_{max} \quad \forall (s, a) \in S \times A$
4: **repeat**
5: obtain current state s and available actions A_s
6: $a^* \leftarrow argmax_a\{V(s, a)|a \in A_s\}$
7: execute a^*
8: obtain state s' and available actions $A_{s'}$
9: $V(s, a^* \leftarrow R(s, a^*, s') + \gamma \cdot max_{a \in A_{s'}} V(s', a)$
10: **until** stopping criteria met
11: stop SUT
12: **end**

where x_a is the number of times action a has been executed and R_{max} is a large positive number (in order to make actions not executed before attractive for the agent)

4 Testing performance metrics

Finding appropriate metrics for assessing the quality of the testing has been a long standing issue. For instance, [8] defines a number of metrics for GUI testing, but these imply having access to the code of the SUT; one of the strengths of TESTAR is precisely not relying on the assumption that this is the case. However, this also implies that specific metrics must be defined.

In previous work [5] we used the number of crashes and the time to crash as a measure of the testing performance, but these pose problems too, because they reveal nothing about to what extent the SUT was explored, a fact particularly relevant if no crashes are detected. Aiming to circumvent that issue, in this work the metrics were chosen as follows:

- **Abstract states** This metric refers to the number of different states, or windows in the GUI, that are visited in the course of an execution.
- **Longest path** Any automated testing tool must ensure the deepest parts of the GUI are tested. To measure whether the tool has just stayed on the surface or it has reached deeper, we define the longest path as the longest sequence of non-repeated (i.e. excluding loops) consecutive states visited.
- **Minimum and maximum coverage per state** We define the *state coverage* as the rate of executed over total available actions in a given state/window; the metrics are the highest and lowest such values across all windows. This allows us to know to what extent actions pertaining to states were explored.

A consequence of not having access to the source code is that the metrics given above can be used to compare the efficiency of different testing methods, but not to assess the overall goodness of a method in isolation, because we do not know the global optima for each metric; for instance, we cannot know exactly how many different states there are.

5 Experiments and results

5.1 The software under test (SUT)

We used two different applications in order to evaluate our $\mathcal{Q}$-learning approach in TESTAR, Odoo and PowerPoint.

Odoo [1] is an open source Enterprise Resource Planning software consisting of several enterprise management applications that can be installed or not depending on the user needs. It can be

[1] See https://github.com/odoo/odoo for Odoo's git repository and issue tracker, including a manual with instructions on how to deploy the server and its requirements.

used to create websites, manage human resource (HR), finance, sales, projects and others. Odoo has a client-server architecture and uses a PostgreSQL database as a management system. Once deployed, we installed the mail, calendar, contacts, sales, inventory and project applications in order to test a wide number of options.

PowerPoint Microsoft PowerPoint is a slide show presentation program currently developed by Microsoft and part of its productivity software Microsoft Office. It is currently one of the most commonly used presentation programs available.

5.2 Procedure

In order to test Odoo with TESTAR a server version of Odoo must first be deployed[2]. Then TESTAR must be configured by supplying the URL that accesses the Odoo client and the browser that will be used to launch it.

On the other hand, to test PowerPoint with TESTAR we must first install it and then TESTAR must be configured by providing the specific command that would be used to run PowerPoint using the cmd (Windows command prompt).

Next, for both tools we run TESTAR in *spy mode*; this uncovers possible problems with items that may not be detected well, such as emergent windows. In addition, it helps detecting undesired actions that might be performed by TESTAR that may bring problems such as involuntary file deletion. A number of parameters must also be set up, which are given in Table 1. With these settings and a first version of the TESTAR *protocol*[3] we carried out three iterations of the testing process, improving the protocol each time so as to remove the problems encountered.

Table 1. Experimental set up. We carried out three iterations involving the five sets. After each iteration the results obtained were used to refine the TESTAR protocol so as to better adapt it to the application.

Set	Max. actions per run	Runs	Action Selection Algorithm	Parameters	
				R_{max}	γ
Q1	1000	30	$\mathcal{Q}$-learning	1	0.20
Q20	1000	30	$\mathcal{Q}$-learning	20	0.20
Q99	1000	30	$\mathcal{Q}$-learning	99	0.50
Q10M	1000	30	$\mathcal{Q}$-learning	9999999	0.95
RND	1000	30	random	N/A	N/A

5.3 Statistical analysis

We run the Kruskal-Wallis non parametric test, with $\alpha = 0.05$, on the results for the five sets. In iteration 3 the test shows that all the metrics have significant differences among the sets. Running pair-wise comparisons provides the results shown in the boxplots contained in Figures 1 and 2; these results are ordered in Table 2, where the shaded column is the best option. It can be seen that for each SUT and metric the best choices are different; also, random selection turns out not to be such a bad choice in most cases. This highlights the importance of an adequate choice of parameters when using $\mathcal{Q}$-learning for action selection.

One metric we have not considered in the statistical analysis is the number of failures encountered, shown in Table 3

In the case of Odoo we can see that although Q20 did not perform so well in the other metrics, it does on the other hand find the higher number of failures (which involve stopping the execution and hence having a lesser chance of increasing the value of other metrics); this must also be taken into account when evaluating the different algorithms. However, for PowerPoint no failures were encountered, so this metric reveals no information.

[2] See the source install tutorial available from https://www.odoo.com/documentation/8.0/setup/install.html

[3] For more details the reader is referred to the tutorial available from www.testar.org

Table 2. Results of the statistical comparison for the sets obtained in the third iteration. The shaded column represents the best choice, the remaining ones are in order of preference.

Metric (Odoo)	Set				
Abstract states	Q10M	RND	Q1	Q99	Q20
Longest path	Q10M	Q99	RND	Q1	Q20
Maximum coverage per state	Q10M	Q20	RND	Q1	Q99
Minimum coverage per state	Q99	Q20	Q10M	Q1	RND
Metric (PowerPoint)	**Set**				
Abstract states	Q99	Q20	RND	Q10M	Q1
Longest path	Q20	Q99	RND	Q10M	Q1
Maximum coverage per state	Q10M	Q20	Q99	RND	Q1
Minimum coverage per state	Q1	Q20	Q10M	Q99	RND

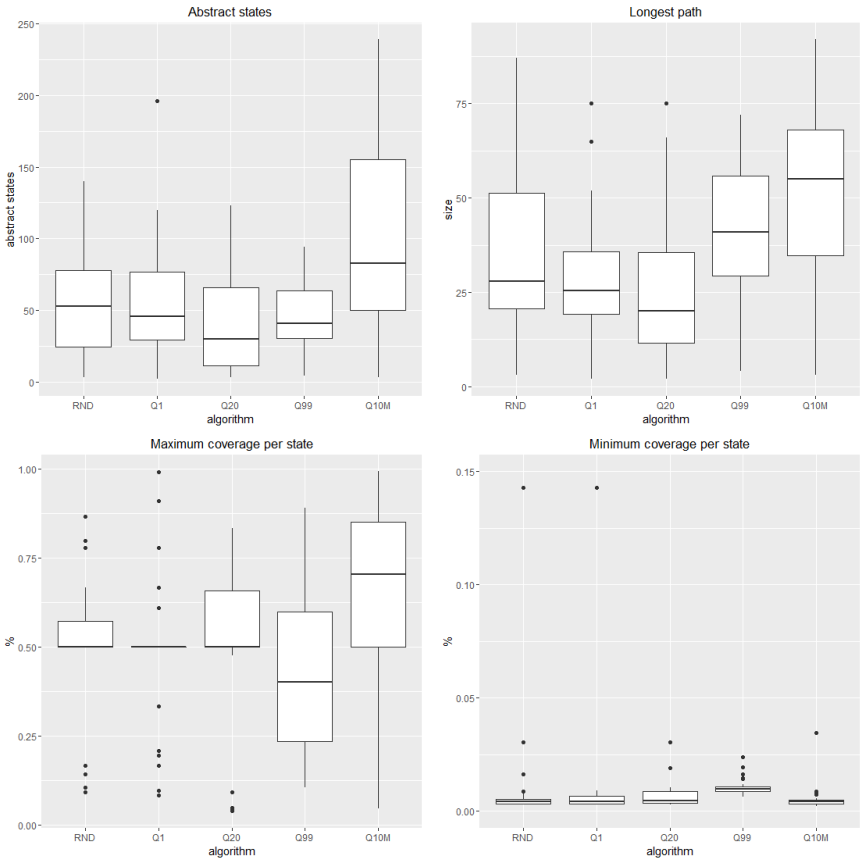


Fig. 1. Boxplots for the four metrics with the results obtained for Odoo in Iteration 3.

Table 3. Number of failures encountered per algorithm in the 3rd iteration when testing Odoo. No failures were encountered in PowerPoint.

Set (Odoo)	Total Failures	Unique failures
Q10M	3	1
Q99	0	0
Q20	6	2
Q1	2	1
RND	1	1

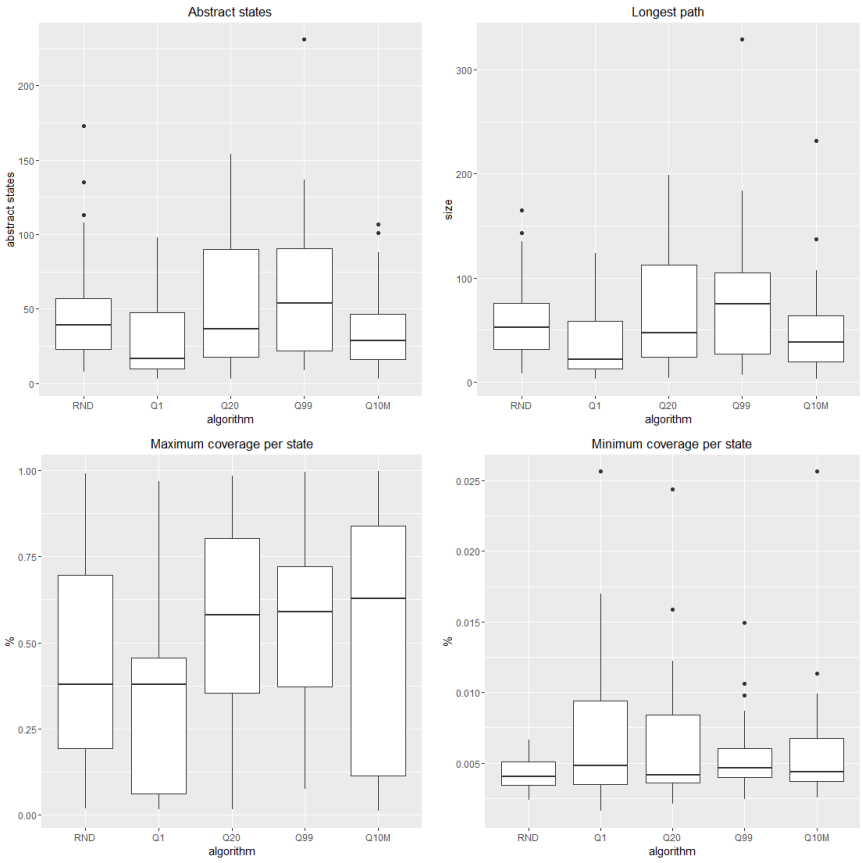


Fig. 2. Boxplots for the four metrics with the results obtained for PowerPoint in Iteration 3.

6 Conclusions

We have shown here the successful application of a $\mathcal{Q}$-learning action selection strategy within the TESTAR tool to the automated testing of the Odoo management software and the commercial application PowerPoint. $\mathcal{Q}$-learning was also compared to *monkey testing*, i.e. using random choice for action selection. Four metrics were defined in order to evaluate the performance. Statistical analysis reveals the superiority of the $\mathcal{Q}$-learning-based method, provided the parameters of the algorithm have been properly selected.

Further work will involve the improvement of the metrics. We will also explore more complex metaheuristics for action selection, especially population based ones (such as ant colony optimisation and genetic programming) in order to improve over the relatively simple $\mathcal{Q}$-learning algorithm.

Acknowledgments.

This work was partially funded by projects **SHIP** (*SMEs and HEIs in Innovation Partnerships*, ref: EACEA/A2/UHB/CL 554187) and **PERTEST** (TIN2013-46928-C3-1-R).

References

1. P. Aho, N. Menz, and T. Rty. Dynamic reverse engineering of GUI models for testing. In *Proc. 2013 Int. Conf. on Control, Decision and Information Technologies (CoDIT'13)*, May 2013.
2. Pekka Aho, Rafael Oliveira, Emil Algroth, and Tanja Vos. Evolution of automated testing of software systems through graphical user interface. In *International Conference on Advances in Computation, Communications and Services, Valencia*, 2016.
3. Emil Alegroth, Robert Feldt, and Lisa Ryrholm. Visual GUI testing in practice: challenges, problems and limitations. *Empirical Software Engineering*, 20:694–744, 2014.
4. S. Bauersfeld, A de Rojas, and T.E.J. Vos. Evaluating rogue user testing in industry: An experience report. In *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, pages 1–10, May 2014.
5. Sebastian Bauersfeld and Tanja E. J. Vos. User interface level testing with TESTAR: What about more sophisticated action specification and selection? In *Post-proceedings of the Seventh Seminar on Advanced Techniques and Tools for Software Evolution, SATToSE 2014, L'Aquila, Italy, 9-11 July 2014.*, pages 60–78, 2014.
6. Sebastian Bauersfeld, Tanja E. J. Vos, Nelly Condori-Fernández, Alessandra Bagnato, and Etienne Brosse. Evaluating the TESTAR tool in an industrial case study. In *2014 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2014, Torino, Italy, September 18-19, 2014*, page 4, 2014.
7. Neha Chaudhary and O.P. Sangwan. Metrics for event driven software. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(1):85 – 89, 2016.
8. Atif M. Memon, Mary Lou Soffa, and Martha E. Pollack. Coverage criteria for GUI testing. In *Proceedings of ESEC/FSE 2001*, pages 256– 267, 2001.
9. U. Rueda, T. E. J. Vos, F. Almenar, M. O. Martínez, and A. I. Esparcia-Alcázar. TESTAR: from academic prototype towards an industry-ready tool for automated testing at the user interface level. In J. H. Canos and M. Gonzalez Harbour, editors, *Actas de las XX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2015)*, pages 236–245, 2015.
10. Tanja E. J. Vos, Peter M. Kruse, Nelly Condori-Fernández, Sebastian Bauersfeld, and Joachim Wegener. TESTAR: tool support for test automation at the user interface level. *IJISMD*, 6(3):46–83, 2015.
11. Christopher J.C.H. Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8(3):279–292, 1992.

GENETIC ALGORITHM FOR OPEN SHOP SCHEDULING PROBLEM

Yacine BENZIANI *, Imed KACEM **, Pierre LAROCHE **
* University of North Dakota, USA
** LCOMS EA 7306, Université de Lorraine, Metz, France
mybenziani@aero.und.edu
{imed.kacem, pierre.laroche}@univ-lorraine.fr

KEYWORDS
Open Shop, Genetic Algorithm.

ABSTRACT

In this paper, we present a genetic algorithm for the open shop scheduling problem. We use a simple and efficient chromosome representation based on the job's occurrence and the fitness function reflect the length of the schedule. The solutions obtained after performing the different operators of the genetic algorithm are always feasible. Heuristic approaches are also developed to generate the initial population and to improve the obtained solutions. The algorithm was implemented and computational results show interesting result.

INTRODUCTION

The open shop scheduling problem consists in sequencing a set *J* of *n* jobs on a set *M* of *m* machines to minimize the length of the schedule (the makespan). A job is a set of *m* operations to be processed on the associated machines without interruption in an arbitrary order (the order of these operations is a decision variable for every job).

The open shop is NP-hard. In (Gonzalez and Sahni 1976) the open shop on three machines O3||Cmax is shown to be binary NP-hard. However, O2||Cmax, the open shop on two machines, is polynomially solvable in O (*n*). The open shop can be formulated by a disjunctive graph. Contrarily to the job shop where the order of operations is fixed, in the open shop all arcs of such a graph are disjunctive, and we have to select the direction of all arcs of the graph. This case implies that the open shop has more decision variables and larger solution space. A complete acyclic selection represents a feasible schedule and its length is the value of the longest path in the graph (Roy and Sussmann 1964).

Only few exact methods are proposed for the general open shop problem, a branch and bound procedure using the disjunctive graph formulation is developed by Brucker et al 1997. Their algorithm is based on the branching scheme of Brucker et al (Brucker et al 1994) and the immediate selection of Carlier and Pinson (Carlier and Pinson 1989) both applied to the job shop problem. This algorithm was improved by using intelligent backtracking (Guéret et al 1998) and according to constraint propagation to reduce the search space (Dorndorf et al 2001).

Due to the limitations of exact methods in term of computation time, various heuristics and meta-heuristics were proposed to the open shop problem and many of them are an adaptation from the job shop problem. Guéret and Prins (Guéret and Prins 1998) have examined the list scheduling heuristics and proposed two new heuristics based on list scheduling with priorities and matching construction algorithm. The shifting bottleneck procedure was generalized to the open shop (Ramudhin and Marier 1996), initially developed for the job shop problem. Two constructive heuristics were presented (Brazel et al 1993): matching algorithm based on the generation of Rank-Minimal schedules and insertion algorithm initially developed for the job shop (Werner and Winkler 1995). In (Liaw 1998) and (Liaw 1999) an iterative approach was proposed based on Benders decomposition and a tabu search. A tabu search is also considered by Taillard (Taillard 1993). A genetic algorithm was proposed to the open shop (Liaw 2000) and (Prins 2000). Recently, an ant colony algorithm with beam search was proposed (Blum2005), a particular swarm optimization is discussed (Sha and Hsu 2008) and a bee colony algorithm is developed (Huang and Lin 2011). The particle swarm of Sha and Hsu gives better results for difficult instances.

In this paper, we describe the basic definition of genetic algorithm in Section 2. In Section 3, we propose a genetic algorithm for the open shop. In Section 4, we discuss the computational results given by the genetic algorithm. And finally we conclude the paper by giving some perspectives in Section 5.

BASIC DEFINITIONS

In this section, we describe the principle of the genetic algorithm proposed to solve the open-shop scheduling problem.

Genetic Algorithm

The genetic algorithm (GA) is a generic search strategy. It is based on the natural selection and mutation of natural

evolution. GA was introduced to optimization by several researchers and became practical after the publication of the book of Holland (Holland 1975). The GA is a very simple process and it is used without knowing the characteristics or the mathematical formulation of the optimization problem.

To elaborate a genetic algorithm, we need an initial population. Each individual in the population is encoded by a chromosome, which represents, in the best case, a unique coding representation. The initial population is obtained generally by heuristics and random process. Each individual is evaluated by a fitness function; this function is related to the considered objective to be optimized

As a natural evolution, the GA is composed of some operators. The first operator is the natural selection: some individuals are chosen among the best ones, to become parents for individuals of the next generation On the selected individuals a mutation operator is applied. It consists on the permutation of chromosome elements. Another important operator is the crossover. This operator is used to obtain new individuals by recombination of the old ones. The new individuals, called children, will be added to the population by replacing the parents or the worst individuals in the population and if we want to keep a fixed number of individuals. The new children can also be added to the population without deleting the parents and the number of individuals grows for each generation.
The process is repeated until the stopping criterion is satisfied. The stopping criterion can be either the number of iterations, a time criterion or the gap between a certain lower bound and the GA solution.

Genetic Algorithm for Scheduling Problem

The genetic algorithm was applied to a wide range of scheduling problems. The chromosome represents generally the order of jobs on machines. The crossover operator is more complicated to implement due to the infeasibility of solutions resulted from GA operators since the job apparition in the sequence could be repeated. We need to rearrange the chromosome to obtain a feasible solution.

The following example describes a chromosome of the single machine problem and different operators of the genetic algorithm, such as the mutation and crossover.

As seen above, a chromosome is given by an order of jobs in the machine.

Chromosome	2	5	4	1	9	7	8	6	3

Figures 1: Chromosome Representation

To perform the mutation operator, in our algorithm, we choose randomly two points in the chromosome and we swap the two associated jobs. Another way to get a mutated child is to choose randomly three points in the sequence and swap the associated jobs in circular permutation. The two figures below illustrate these mutations:

Parent	2	5	4	1	9	7	8	6	3
Child	2	5	6	1	9	7	8	4	3

Figures 2: Simple Mutation Operator

Parent	2	5	4	1	9	7	8	6	3
Child	2	5	6	1	9	4	8	7	3

Figures 3: Multi-points Mutation Operator

Given two parents, the crossover operator is applied as follows: we choose randomly a point in the chromosome and we copy the first part of the first parent in the first child. We complete the remaining positions in the sequence with the missing jobs following the apparition order in the second parent. This ensures that the obtained children represent feasible solutions. We apply the same process for the second child. The following figure illustrates the process.

Parent 1	2	5	4	1	9	7	8	6	3
Parent 2	1	7	6	9	2	3	5	8	4

Child 1	2	5	4	1	7	6	9	3	8
Child 2	1	7	6	9	2	5	4	8	3

Figures 4: Crossover Operator

GENETIC ALGORITHM FOR THE OPEN SHOP PROBLEM

A population is an array containing several individuals. Each individual represents a solution of the open shop problem. The individuals are evaluated by a fitness function according to the objective function. At each step of the algorithm, some individuals are selected to reproduce new members by the crossover and mutation operators.

Solution Representation

A solution of the open shop is defined by ordering the operations on each machine and by fixing an order of operations. In the disjunctive graph representation, a feasible schedule is a complete acyclic selection obtained by an orientation of the disjunctive arc. The makespan equals to the longest path in the oriented graph. A feasible schedule is also an assignment of starting time for all operations. A solution can be represented by different ways. An example of 3x3 machines- jobs with processing times given in Table 1 will illustrate these different ways:

Table 1: Processing Time

		Job		
		1	***2***	***3***
Machine	***1***	4	5	8
	2	2	9	9
	3	4	7	2

Starting Time Representation.
The first way is obtained by taking n x m integer value array which represents the starting times (y_i) of operations. The operations are ordered job by job; we put in the first m position the operations of the first job and so on.

i	1	2	3	4	5	6	7	8	9
y_i	0	18	25	25	9	18	9	0	18
M	1	2	3	1	2	3	1	2	3
Job	1			2			3		

Figure 5: Starting Time Representation

Jobs Sequence Representation.
The second representation is defined by the sequence of jobs in the machine and operations in the job, called the job sequence. It consists of two 2-dimensional arrays: the first array contains the job sequence on each machine, and the second defines the precedence order of operations.

	Job Sequence				Precedence Order		
	J_1	J_2	J_3		J_1	J_2	J_3
M_1	1	3	2	M_1	1	2	3
M_2	2	1	3	M_2	2	3	1
M_3	3	2	1	M_3	3	1	2

Figure 6: Job Sequence Representation

Working Sequence Representation.
In the last representation, the solution is represented by the working sequence with two n x m arrays. The first array contains the job apparition called Inter-Job, and the second array contains the precedence order called Intra-Job. The Intra-Job defines the order of operations in the job, and the inter-Job defines the order of jobs through the machines.

Following the job order in Intra-Job, the job occurrence in Inter-Job defines the apparition order of its operations and the sequence in the machines is constructed. Each job appears m times (number of job operations).

Op	1-3	1-1	1-2	2-3	2-2	3-2	2-1	3-1	3-3
Inter	3	1	2	3	2	2	1	1	3
Intra	1	2	3	2	3	1	3	1	2

Figure 7: Working Sequence Representation

From each representation, we can find the others by constructing the schedule using the Gant Chart. The working sequence is obtained by ordering the starting time in non-decreasing order and we put the operations in the Inter-Job following this order and respecting the Intra-Job giving by job operations order.

Chromosome Representation and Fitness Function

In our genetic algorithm we use the working sequence representation. The chromosome is represented by two integer arrays. The first array is the Inter-Job and contains the job number, and the second is the Intra-Job and contains the machine number. Each solution has a unique coding representation and each chromosome corresponds to one solution.

Inter-Job	3	1	2	1	2	2	1	3	3
Intra-Job	3	2	1	2	3	1	3	2	1

Figure 8: Chromosome

The fitness function is the value of the makespan; the length of the schedule. Given a solution represented by its chromosome, we process as follows to compute the length of the schedule: we read the first operation and we set the associated job length and the loading machine with its processing time. We read the remaining operations, and at each position we update the associated job length and loading machine with the possible starting time added to the related processing time. When all operations are completed, the length of the schedule equal to the maximum of loading machine and job length.

Initial Population and Heuristics for the Open Shop

The initial population contains several different solutions. The solutions are obtained by some heuristics used to give a good gene. Some of them are obtained by using priority rules (Giffler and Thompson, 1960). The population is completed randomly to diversify the chromosomes.

In this sub-section, we will describe some heuristics for the open shop. It is very useful to improve the genetic algorithm solutions.

Greedy Search Technique.
This heuristic is based on the search tree generated by a branch and bound method (we do not describe the branch and bound algorithm in this paper, we refer the reader to the literature for more details (Brucker et al 1994) and (Dorndorf et al 2001)). We apply a Depth-First-Search for only $2m$ (n (n - 1)) (number of combinations to obtain a complete schedule) and we choose the vertex in the tree search having the best evaluation to find the first solution.

Neighborhood Search.
Given an initial solution for the open shop, we explore its neighborhoods by swapping two operations. After trying to swap all operations two per two, then we swap two machines and we reiterate the operations process until testing all combinations of machines.

Mutation Operator

The mutation operation is used to diversify the population, so that, if we are blocked in a local optima we can jump to another area in solutions domain. In the apparition order, we choose randomly two positions in the working sequence, containing two different jobs, and then we swap the selected operations. And in job precedence order, on each job we swap two machines (see Figure 9).

Parent

Inter-Job	3	1	2	3	2	2	1	1	3
Intra-Job	1	2	3	2	3	1	3	1	2

Child

Inter-Job	3	1	2	1	2	2	1	3	3
Intra-Job	3	2	1	2	1	3	3	2	1

Figure 9: Mutation

Crossover Operator

After choosing two arbitrary individuals from the population, we apply the crossover operator. We choose an arbitrary position in the apparition order of the first parent and we build the new individual by taking the first part of the first parent and we complete the apparition order by the genes appearing in the second parent. In the precedence order, we choose randomly a job; the first child obtains the first j precedence order of the first parent and we complete the precedence order by the last part of the second parent. We apply the same technique for the two parents to obtain two new individuals. In the example below, we generate two different solutions:

Parents

Inter-Job	3	1	2	3	2	2	1	1	3
Intra-Job	1	2	3	2	3	1	3	1	2

Inter-Job	3	1	2	3	2	2	1	1	3
Intra-Job	3	1	2	1	3	2	3	2	1

Children

Inter-Job	3	1	2	1	3	3	2	2	1
Intra-Job	1	3	2	2	1	3	3	2	1

Inter-Job	3	1	2	2	3	2	2	1	3
Intra-Job	3	1	2	1	2	3	3	1	2

Figure 10: Crossover

Process of the Genetic Algorithm

The genetic algorithm runs as follows: First, we produce individuals using the heuristics and we complete the population with random solutions. In our implementation we use 500 individuals. For each generation, we choose arbitrary, from the 30% best solutions, two individuals who represent the parents, on these parents, we perform the mutation and the crossover operators with a fixed probability, 0.1 for the mutation and 0.8 for the crossover, to obtain two new children. Then, we select randomly two other individuals among the 70% less well and we replace them with the new children. After that we select randomly with a probability of 10% an individual and apply the neighborhood search. We reiterate the process until the stopped condition is satisfied, in our test we stop after 1500 iterations.

COMPUTATIONAL RESULTS

In this section, we tested the genetic algorithm and the heuristics on the instances of Taillard (Taillard 1993) denoted Tai *-* (*-* : number of machines and instance number), and instances of (Guéret and Prins 1999) GP *-*. The PC used is an Intel Core 2 Quad Processor CPU 2.83 GHz under Windows 7 operating system, the program is coded in Visual C++. We report in the following table results the name of the instances, the lower bound, the best known solution BKS, the GA solution and the gap computed by (Cmax - AG)/AG.

Taillard's instances are composed of four types such that the number of machines is equal to the number of jobs $n = m = 4$, 5, 7, 10, each part contains ten different instances. The *lb* in the table represents the trivial lower bound; the maximum of maximum loading machines and maximum job length. The best known solution is the optimal solution and it is proven for all instances. The gap is calculated if the optimal solution is not reached. The optimal solution is obtained in all instances with 4, 5, 7 and 10 machines-jobs. We report in the table below instances with 10*10 machine- jobs.

Table 2: Taillard's instances 10*10

Instances	lb	BKS	AG
Tai 10-0	637	637	637
Tai 10-1	588	588	588
Tai 10-2	598	598	598
Tai 10-3	577	577	577
Tai 10-4	640	640	640
Tai 10-5	538	538	538
Tai 10-6	616	616	616
Tai 10-7	595	595	595
Tai 10-8	595	595	595
Tai 10-9	596	596	596

Guéret and Prins instances are composed of four types. The number of machines is equal to the number of jobs $n = m = 4$, 5, 6, 7, 8. The value lb is the best lower bound. The optimal solution is obtained in all instances of 3*3, 4*4, 5*5, 6*6, and 7*7 machines-jobs except for one instance. We reported in table 3 the solutions obtained by the GA for the instances of 7*7 and 8*8.

Table 3: Guéret and Prins instances 7*7

Instances	lb	BKS	AG
GP07-01	1000	1159	1159
GP07-02	1000	1185	1185
GP07-03	1000	1237	1237
GP07-04	1000	1167	1167
GP07-05	1000	1157	1157
GP07-06	1000	1193	1193
GP07-07	1000	1185	1185
GP07-08	1000	1180	**1181**
GP07-09	1000	1220	1220
GP07-10	1000	1270	1270

Table 4: Guéret and Prins instances 8*8

Instances	lb	BKS	AG
GP08-01	1000	1130	1160
GP08-02	1000	1135	1136
GP08-03	1000	1110	1111
GP08-04	1000	1153	1153
GP08-05	1000	1218	1218
GP08-06	1000	1115	1115
GP08-07	1000	1126	1126
GP08-08	1000	1148	1148
GP08-09	1000	1114	1114
GP08-10	1000	1161	1161

We note that instances of Taillard are less difficult to solve than instances of Guéret and Prins. This is due to the fact that the optimal solution has the same value as the lower bound.

CONCLUSION

In this paper, we have presented a genetic algorithm for the open shop. The chromosome representation was helpful for implementing of the GA algorithm. It also allows the operators of GA to be simpler and avoids the occurrence of infeasible solutions. The GA gave us good computational results and it can solve many instances. The gap obtained is about 5% for the hardest instances. We have also found the optimal solution for many instances.

For future research, we will focus on solving harder instances and the implementation of parallel GA algorithm for shop scheduling problem. It is also interesting to generalize the GA for the flexible scheduling problem since this problem has a similar formulation.

REFERENCES

Blum, C. (2005). Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling. *Computers & Operations Research*, *32*(6), 1565-1591.

Bräsel, H.; T. Tautenhahn; and F. Werner. (1993). Constructive heuristic algorithms for the open shop problem. *Computing*, *51*(2), 95-110.

Brucker, P.; J. Hurink; B. Jurisch; and B. Wöstmann. (1997). A branch & bound algorithm for the open-shop problem. *Discrete Applied Mathematics*, *76*(1), 43-59.

Brucker, P.; B. Jurisch; and B. Sievers. (1994). A branch and bound algorithm for the job-shop scheduling problem. *Discrete applied mathematics*, *49*(1), 107-127.

Carlier, J. and É. Pinson (1989). An algorithm for solving the job-shop problem. *Management science*, *35*(2), 164-176.

Dorndorf, U.; E. Pesch; and T. Phan-Huy. (2001). Solving the open shop scheduling problem. *Journal of Scheduling*, *4*(3), 157-174.

Giffler, B. and G.L. Thompson. (1960). Algorithms for solving production-scheduling problems. *Operations research*, *8*(4), 487-503.

Gonzalez, T. and S. Sahni. (1976). Open shop scheduling to minimize finish time. *Journal of the ACM (JACM)*, *23*(4), 665-679.

Guéret, C. and C. Prins. (1998). Classical and new heuristics for the open-shop problem: A computational evaluation. *European Journal of Operational Research*, *107*(2), 306-314.

Guéret, C. and C. Prins. (1999). A new lower bound for the open-shop problem. *Annals of Operations Research*, *92*, 165-183.

Guéret, C. ; N. Jussien ; and C. Prins. (2000). Using intelligent backtracking to improve branch-and-bound methods: An application to open-shop problems. *European Journal of Operational Research*, *127*(2), 344-354.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press.

Huang, Y. M. and J.C. Lin. (2011). A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems. *Expert Systems with Applications*, *38*(5), 5438-5447.

Liaw, C. F. (1998). An iterative improvement approach for the nonpreemptive open shop scheduling problem. *European Journal of Operational Research*, *111*(3), 509-517.

Liaw, C. F. (1999). A tabu search algorithm for the open shop scheduling problem. *Computers & Operations Research*, *26*(2), 109-126.

Liaw, C. F. (2000). A hybrid genetic algorithm for the open shop scheduling problem. *European Journal of Operational Research*, *124*(1), 28-42.

Prins, C. (2000). Competitive genetic algorithms for the open-shop scheduling problem. *Mathematical methods of operations research*, *52*(3), 389-411.

Ramudhin, A. and P. Marier. (1996). The generalized shifting bottleneck procedure. *European Journal of Operational Research*, *93*(1), 34-48.

Roy, B. and B. Sussmann. (1964). Les problemes d'ordonnancement avec contraintes disjonctives. *Note ds*, *9*.

Sha, D. Y., & Hsu, C. Y. (2008). A new particle swarm optimization for the open shop scheduling problem. *Computers & Operations Research*, *35*(10), 3243-3261.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *european journal of operational research*, *64*(2), 278-285.

Werner, F. and A. Winkler. (1995). Insertion techniques for the heuristic solution of the job shop problem. *Discrete Applied Mathematics*, *58*(2), 191-211.

Metaheuristics for intelligent transportation systems

A Genetic Algorithm for solving the multicriteria routing problem in public transit networks

O. Dib[1], M. Manier[2], L. Moalic[3] and A. Caminada[4]

1. IRT SystemX, 8 Avenue de la Vauve, 91120 Palaiseau France
omar.dib@irt-systemx.fr

2. OPERA – UTBM, Belfort France
{omar.dib ; marie-ange.manier ; Laurent.moalic ; alexandre.caminada}@utbm.fr

Keywords: Multimodal networks, Multicriteria shortest paths, Genetic algorithm.

1 Introduction

Route planning in urban public transport systems constitutes a common decision problem faced by travelers. As in real life, commuters do not only seek short time travels. However, they tend to consider other elements into their journeys such as monetary cost, comfort (quality of mode) and effort (walking distance, number of transfer, waiting time…). Therefore, there is a real need to develop a seamless application that provides passengers with efficient itineraries according to their needs and preferences.

Routing applications whether they arise in transportation area or other domains such as communication networks refer for solving Shortest Path Problems (SPPs). While solving some routing problems can be done in a straightforward manner, computing shortest paths under certain circumstances is not always an easy task. For instance, solving the one-to-one SPP in static networks can be easily accomplished by applying the well-known algorithm of Dijkstra. On the other side, computing multicriteria shortest paths appears to be more difficult especially in large-scale dynamic networks. Computing itineraries w.r.t several criteria refers to the Multiobjective Shortest Path Problem (MOSP), a fundamental problem in the field of multiobjective optimization. Solving the emerging problem consists of finding the set of non-dominated journeys from which the user chooses his/her most preferred one. Given two journeys j_1 and j_2, we say that j_1 dominates j_2 if there is at least one criterion for which j_1 has a better value than j_2 and there is no criterion for which j_2 has a better value than j_1. A journey j is then called Pareto-optimal if it is not dominated by any other journey. The main difficulty in multiobjective contexts stems from the fact that, in many optimization problems, determining the entire set of nondominated solutions is a tedious task since one problem may have a huge number of nondominated solutions (even in case of two objectives). Additionally, and in contrast to single criteria search, one cannot abort the search after finding a first optimal solution. Even after finding all Pareto-optima, search algorithms require a substantial amount of time to guarantee that no further solution exists. Therefore, in many optimization problems, especially those requiring real time answers, we do not focus on finding the optimal Pareto solution set. Rather, we try to use approximate methods whereby we compute near optimal solutions in reasonable computational time.

The Genetic Algorithm (GA) that belongs to the population-based metaheuristics and was introduced by Holland in 1987 [1] is one of the approximate approaches that has been efficiently used for solving a wide range of multicriteria problems. For instance, [2] used a GA to solve the one-to-one SPP in large-scale road networks. [1] and [4] worked with evolutionary algorithms to compute single source shortest paths using single-objective fitness. [5] proposed a GA to find shortest paths in computer networks. [6] and [7] also worked with GAs to find shortest paths in data networks. [8] proposed effective crossover operators for the all-pairs SPP. In this work, we develop a GA for solving the multi-criteria routing problem in time-dependent public transit network. As optimization criteria, we use travel time, travel cost, number of transfers and total walking time. As transportation modes, we use Railway, Bus, Tram, Metro and pedestrian. We assess the performance of the proposed GA by solving a wide range of real-world routing queries defined on the public transport network of the City of Paris and its suburbs. We compare the performance of the introduced GA with an exact multi-criteria shortest path algorithm. Experimental results indicate that the proposed GA is efficient enough to give near optimal solutions in reasonable computational time.

2 Modeling Approach

This section considers modeling a time-dependent public transit network. It should be clarified that the term "transport network" is used in the sense of multiple fixed scheduled transport services. A key difference to static networks is that public transit networks are inherently time-dependent, since certain segments of the network can only be traversed at specific, discrete points in time. As such, the first challenge concerns modeling the timetable appropriately in order to enable the computation of journeys. Roughly speaking, a timetable consists of a set of stops (such as bus stops or train platforms), a set of routes (such as bus or train lines), and a set of trips. Trips correspond to individual vehicles that visit the stops along a certain route at a specific time of the day. Trips can be further subdivided into sequences of elementary connections, each given as a pair of (origin/destination) stops and (departure/arrival) times between which the vehicle travels without stopping. Each network in our approach is modeled as a separate directed graph. An additional work is then done to integrate all sub-graphs into one larger graph. As a first step of modeling, we introduce three types of nodes that correspond to stations, platforms and departure events. A station comprises a set of platforms where passengers wait for vehicles. An edge is inserted between a platform and its parent station; its weight represent the minimal time required for accessing that platform from the entrance point of the station. A platform cannot belong to more than one station, however, a station can contain one or several platforms. Each platform has also a type (Bus, railway, tram...) to differentiate between modes. Since a timetable consists of time-dependent events (e.g., a vehicle departing at a stop) that happen at discrete points in time, we use the idea of the time-expanded model that builds a space-time graph to unroll time. Roughly speaking, the model creates a vertex for every event in the timetable that consists of vehicle departing from a platform *x* at *dt* and arrives to another platform *y* at *at*. Timestamps are inserted into event nodes to account for the departure and arrival times. Event nodes are ordered in the way that a higher-level node refers to an earlier event. In addition, waiting, boarding and alighting arcs are inserted between event nodes and platforms. To account for transfers between and inside stations, we inserted transfer edges with transfer time between platforms.

3 Genetic Algorithm

After introducing the modeling approach in the previous section, we explain in this section the proposed GA. As in most GA's schemes, the first challenge lies in the genetic representation. A solution (individual) in our work is any route that allows going from the starting node at the user's departure time to the destination node. To represent such solution, we have used the permutation encoding. Typically, each chromosome consists of a string of positive integers that represent the IDs of edges included in the route. The size of chromosomes is not fixed since several routes with different nodes and edges may exist to go from an origin point to a destination one. After encoding, we explain the generation of initial solutions. The composition of the initial population in our approach is remarkably different compared to traditional GAs. Our initial solutions are a set of feasible paths generated using a construction heuristic based on a double search algorithm. More precisely, we simultaneously run a forward search from the starting node with respecting the departure time *dt* and a backward search from the destination node. A feasible path is then found when the two searches intersect. As a result of this operation, we get feasible routes having different lengths to go from one node to another. When it comes to the evaluation phase, we do not use a simple scalar value like in single criterion optimization. However, the fitness function in our work is represented by a p-dimensional vector where in each dimension we evaluate the value of one criterion. As in traditional GA scheme, the proposed algorithm evolves through three operators: *selection*, *crossover* and *mutation*. To accomplish the crossover, a single tournament selection is employed. After ordering the individuals in the population, each chromosome in the odd position is mated with the next chromosome in order to produce new individuals. By doing so, we produce a new population having twice the size of the current population. The best half individuals are then selected for the next generation and the rest are ignored. Duplicated individuals are replaced with newly generated chromosomes to avoid reprocessing the same individual. Single point crossover technique has been used in our approach in order to produce offspring. An intersection node between two individuals is selected to be the crossover point. Current individuals exchange then part of genes with each other before or after the crossover point to generate offsprings. To accomplish the mutation, we randomly select one individual and we try to replace one subsequent route by a new feasible one. By doing so, we increase the diversity in the population and thereby, we prevent the algorithm

from premature convergence. Finally, to allow the convergence of the proposed method, two terminating criteria are use i) when the maximum number of evolutions is reached (Generation Number) ii) when no better solutions are found during several evolutions (Fitness Convergence).

4 Experimental Results

To assess the performance, we have applied the proposed algorithm to solve 10000 routing queries based on the real data of the French region Île-de-France that includes the city of Paris and its suburbs. The start time, departure and arrival stations are uniformly picked at random. Transportation data are provided by the transport organization authority that controls the Paris public transport network and coordinates the different transport companies operating in Île-de-France. Data comprise geographical information, as well as, timetable information for four transport modes (Bus, Metro, Railway, and Tram). More precisely, data encompass 17950 stations; 41047 platforms; 195000 transfers; 303000 trips and 6800000 events for one day. We use in this GA the following parameters: the initial population size is 5; the probability of crossover is 0.9; the probability of mutation is 0.1; the maximum number of generation is 500; The number of generations used to ensure a fixed state in the population is 100. We run algorithms on an Intel core I5 machine of 8 GB RAM and we used java as a programming language.We compare the performance of the proposed GA with an exact shortest path algorithm regarding two axes: the CPU computation time and the quality of solutions. Results indicate that the average running time to solve routing queries w.r.t four criteria (travel time, cost, number of transfers and walking time) is 390 seconds when using the exact algorithm. However, the GA spends less than 117 milliseconds to accomplish its search process. The average gap to the optimality of the GA w.r.t the exact approach may reach a maximum of 4%. Therefore, the proposed GA is efficient enough to be used in a real world routing system. As future works, we have planned to integrate other transportation modes such as Car and Bike Sharing, as well as, consider some stochastic parameters such as accidents and delays.

Acknowledgments

This research work has been carried out in the framework of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program “Investissements d’Avenir”.

References

[1] Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. Machine learning, 3(2), 95-99.

[2] Dib, O., Manier, M. A., & Caminada, A. (2015, September). A Hybrid Metaheuristic for Routing in Road Networks. In Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on (pp. 765-770). IEEE.

[3] Surender Baswana, Somenath Biswas, Benjamin Doerr, Tobias Friedrich, Piyush P. Kurur, Frank Neumann: Computing single source shortest paths using single-objective fitness. FOGA 2009: 59-66

[4] Benjamin Doerr, Edda Happ, Christian Klein: Tight Analysis of the (1+1)-EA for the Single Source Shortest Path Problem. Evolutionary Computation 19(4): 673-691 (2011)

[5] Kumar, Rakesh, and Mahesh Kumar. "Reliable and Efficient Routing Using Adaptive Genetic Algorithm in Packet Switched Networks." IJCSI International Journal of Computer Science Issues 9.1 (2012): 1694-0814.

[6] Behzadi, Saeed, and ALIA ALESHEIKH. "Developing a Genetic Algorithm for Solving Shortest Path Problem." NEW ASPECTS OF URBAN PLANNING AND TRANSPORTATION (2008).

[7] Kumar, Dr Rakesh, and Mahesh Kumar. "Exploring Genetic algorithm for shortest path optimization in data networks." Global Journal of Computer Science and Technology 10.11 (2010).

[8] Benjamin Doerr, Daniel Johannsen, Timo Kötzing, Frank Neumann, Madeleine Theile: More Effective Crossover Operators for the All-Pairs Shortest Path Problem. PPSN (1) 2010: 184-193

A hybrid algorithm for finding an integer route assignment

L. Idres[1] and M.S. Radjef[1]

Research Unit LaMOS, Department of Operational Research, Exact Sciences Faculty, Bejaia University
06000, Algeria.
ilahna@yahoo.fr, radjefms@gmail.com

1 Introduction

Route assignment is one of the central problem in transportation field. It was firstly modeled by Beckmann et al.[1] as a mathematical program. However, this model suffers from several drawbacks such that the solution integrity. In fact, when the obtained solution is not integer, Rosenthal[3] shows that it is not a good approximation of the equilibrium. He suggested to use a non-cooperative game to describe the interaction among the road users. Following Rosenthal proposal, an asymmetric network congestion game modeling route assignment is proposed. Since the problem of finding a pure Nash equilibrium (PNE) in such games is NP-complete[2], an hybrid algorithm consisting of two steps is built. The first step is constituted by a Greedy Best Response (GBR) algorithm which has two major advantages: polynomial complexity and returns an integer solution. The second step is inspired from the ant colony algorithm.

2 Model and Preliminaries

Let us consider a road network as a directed graph $G = (V, E)$, where each vertex $v \in V$ represents an intersection of the road network and each arc $e \in E$ represents a route. On this graph, we distinguish l particular pairs of vertices that represent the different Origin-Destination (O-D) pairs are defined. Each (O-D) is linked by network roads that are modeled by a set of path $\Phi_j = \{\sigma_{j1}, \sigma_{j2}, \dots \sigma_{j|\Phi j|}\}, j = \overline{1,l}$. N road users compete over this road network, such that each user chooses a path linking his (O-D) pair in order to minimize his travel time. This situation is modeled as the asymmetric network congestion game $\Gamma =< I, \mathcal{R}, \{X_j^i\}_{i\in I, j=\overline{1,l}}, \{\mathcal{C}^i\}_{i\in I} >$, where

- $I = \bigcup_{j=1}^{l} N_j$, denotes the set of players representing the road users, with N_j the subset of the players sharing the j^{th}(O-D) pair and $N_j \cap N_i = \varnothing$, $\forall i, j = \overline{1,l}$, $i \neq j$, $\sum_{j=1}^{l} |N_j| = N$.
- The resource set $\mathcal{R}$ represents the set of the arcs of the graph G, i.e. $\mathcal{R} = E$.
- $X_j^i = \Phi_j$ is the strategy set of the i^{th}player ($i \in N_j$) traveling through the j^{th} (O-D) pair.
- The cost function $\mathcal{C}^i$ of the i^{th} player using his strategy $x_j^i = \sigma_j^i \in X_j^i$ represents his travel time through the path σ_j^i and depends on the paths of the other players $\sigma^{-i} \in X_j^{-i} = \prod_{k=1,k\neq i}^{N} X_j^k, j = \overline{1,l}$. The following BPR function is used to calculate it,

$$\mathcal{C}^i(\sigma^{-i}, \sigma_j^i) = \sum_{e\in\sigma_j^i} tf(e)\left(1 + \gamma\left(\frac{n(e)}{K(e)}\right)^{\alpha}\right), \tag{1}$$

where, $tf(e)$(respectively, $K(e)$) is the free travel time (respectively, the capacity) of the arc $e \in E$. γ and α are coefficients depending on the type of arcs (roads). $n(e)$ is the number of players including the arc (resource) e in their path (strategy) σ.

Solution Concept

Wardrop [4] was one of the first to describe a principle that makes the road network in a steady state. This principle is known as User Equilibrium (UE) or "*selfish equilibrium*" and aims to

minimize the travel time of each user, concretely ***"the journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route"***[4]. In game theory, the commonly used concept is Nash equilibrium at which no player has interest to deviate unilaterally from his Nash strategy. It is easy to see that any UE is a Pure Nash Equilibrium (PNE).
In practice one can be satisfied by an approximation of the PNE, hence in this work the route assignment must satisfy the conditions of an $\epsilon-$PNE. That is to say, a player does not deviate to another strategy if this one does not brings him a benefits greater than ϵ.

3 Algorithm

In order to find an $\epsilon-$PNE of the game Γ an hybrid algorithm is proposed. This latter contains two steps. The first one consists of a Greedy Best Response (GBR) algorithm, and the second one is an adaptation of ant colony.

3.1 First step: GBR algorithm

GBR is a simple algorithm that considers the players one by one proceeding as follows: selects a player, calculates his travel time through his paths and affects him to the path with the lowest cost. Then, another player is selected and the above method is applied taking into account the assignment of the previous player. This procedure is repeated until there is no player to assign. GBR has two major advantages, the first one is the computing complexity (polynomial) and the second one is that provides an integer assignment. However, the obtained assignment does not necessarily converge to an $\epsilon-$PNE, that is why the second step is added.

3.2 Second step: Adaptation of ant colony(AC)

In order to improve the obtained assignment by the GBR algorithm, a second step inspired from the ant colony is built. Unlike the classical ant colony algorithm where the ants move from a common origin to a common destination looking for the shortest path, herein there are several origins and destinations. One can consider several ant colonies (one colony for each origin-destination pair) however these colonies are interdependent. That is to say the deposed pheromone by an ant of a specific colony influences the ants of the other colonies. The final solution must satisfy the following conditions for each (O-D) pair,

$$\max_{\sigma_j \in \Phi_j^u} Pr(\sigma_j) - \min_{\sigma_j \in \Phi_j^u} Pr(\sigma_j) \leq \epsilon, \quad j = \overline{1,l}, \tag{2}$$

$$\max_{\sigma_j \in \Phi_j^{-u}} Pr(\sigma_j) \leq \min_{\sigma_j \in \Phi_j^u} Pr(\sigma_j), \quad j = \overline{1,l}. \tag{3}$$

where, $Pr(\sigma_j)$ denotes the probability to choose the path σ_j and Φ_j^u (respectively. Φ_j^{-u}) denotes the set of used paths (respectively. unused paths).

4 Numerical Example

Let us consider the network of Fig.1 containing two origins (O_1, represented by the vertex 1 and O_2, represented by the vertex 3) and two destinations (D_1, represented by the vertex 7 and D_2 represented by the vertex 14).

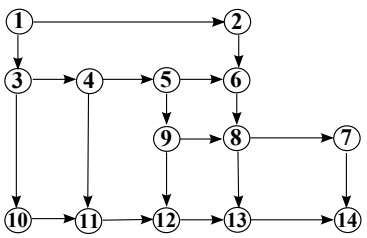


Fig. 1. Network example

The assignments are computed with both GBR algorithm and the proposed hybrid algorithm (GBR+AC). The obtained results are shown in the table bellow.

Table 1. The obtained results

		Number of players		Travel time				Number of players		Travel time	
	Paths	GBR	GBR+AC	GBR	GBR+AC		Paths	GBR	GBR+AC	GBR	GBR+AC
Pair	σ_1^1	4643	4643	24.76	24.76	Pair	σ_2^1	908	908	23.13	23.13
$(O_1 - D_1)$	σ_1^2	0	0	27.14	27.14	$(O_2 - D_1)$	σ_2^2	0	0	25.42	25.42
	σ_1^3	0	0	29.49	29.49						
		Number of players		Travel time				Number of players		Travel time	
	Paths	GBR	GBR+AC	GBR	GBR+AC		Paths	GBR	GBR+AC	GBR	GBR+AC
Pair	σ_3^1	929	917	26.76	24.02	Pair	σ_4^1	3949	2868	25.14	22.07
$(O_1 - D_2)$	σ_3^2	39	39	26.00	26.05	$(O_2 - D_2)$	σ_4^2	0	1081	22.00	22.05
	σ_3^3	0	0	33.01	33.00		σ_4^3	0	0	22.01	22.06
	σ_3^4	0	12	26.01	26.06		σ_4^4	0	0	27.42	26.91
	σ_3^5	0	0	28.35	26.14		σ_4^5	1532	1532	21.02	21.06

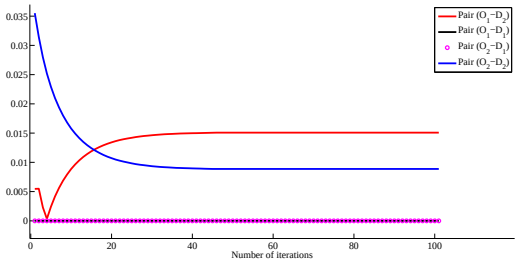


Fig. 2. $\max_{\sigma_j \in \Phi_j^u} Pr(\sigma_j) - \min_{\sigma_j \in \Phi_j^u} Pr(\sigma_j)$.

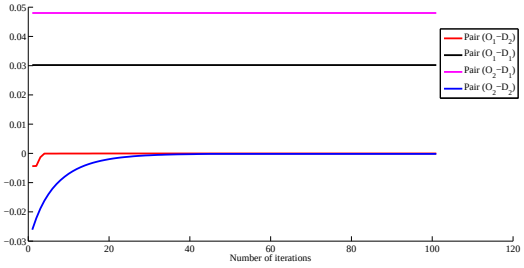


Fig. 3. $\min_{\sigma_j \in \Phi_j^u} Pr(\sigma_j) - \max_{\sigma_j \in \Phi_j^{-u}} Pr(\sigma_j)$.

Discussion

From Table 4. one can remark that the obtained results using GBR algorithm converge to an $\epsilon-$PNE for only two (O-D) pairs, whereas the proposed hybrid algorithm improves the results and makes the assignments closer to $\epsilon-$PNE for a greater number of (O-D) pairs. From the graphs of Fig. 2 and Fig. 3 we deduce that the solution converge quickly to $\epsilon-$PNE. Finally, one can say that the obtained results using the hybrid algorithm are encouraging however it could be ameliorated. Indeed, concerning the pair $(O_1 - D_2)$ the difference between the probabilities of the used paths decreases until iteration 5 where it reaches its minimum, then increase until iteration 40 where it becomes steady (see Fig. 2).

References

1. Beckmann, M. and McGuire, C.B. and Winsten, C.B.:Studies in the economics of transportation. Yale University Press:New Haven. London: Geoffrey Cumberlege: Oxford University Press (1956)
2. Fabrikant, A. and Papadimitriou, C. and Talwar, K.:The complexity of pure Nash equilibria. In proceeding of the 36 th Annual ACM Symposium on Theory of Computing, Chicago. 604–612 (2004)
3. Rosenthal, R.W.:The network Equilibrium Problem in Integers. Networks (1) 53-59 (1973)
4. Wardrop, J.G.:Some theoritical aspect of road traffic research.Proceedings of the Institute of civil engineers. Part II (1) 325-378 (1952)

Real-life resolution for Costumers-Dependent Dial-A-Ride-Problem

S. Nasri[1] and H. Bouziri[2]

[1] LARODEC Laboratory, ESC, University of Manouba, Tunisia
sonia2015.nasri@gmail.com
[2] LARODEC Laboratory, ESSECT, University of Tunis, Tunisia
hend.bouziri@gmail.com

Abstract

Dial-A-Ride Problem (DARP) is one of the vehicle routing pickup and delivery problems. A basic variant is the multi-vehicles DARP which is proposed by Cordeau and Laporte [3]. This problem is NP-Hard as it is stated by Healy and Moll [9]. Much effort has been devoted to the problem resolution seeking for near optimal solutions. However, in our days, transport on demand problems need to be applied to realistic frameworks to deal with real-life applications. Hence, we propose a new model for the multi-vehicles DARP. This model is the Costumers-Dependent DARP (CDDARP) which imposes new real-life criteria on the DARP constraints where the objective is to minimize total transit time. Further, a new Tabu Search (TS) method named TS-CDDARP is provided to solve CDDARP. An efficient insertion heuristic is proposed to satisfy all requests. In addition, a flexible routing plan is made over a new neighborhood strategy that optimizes the total vehicles travel time. To validate our model, we investigate numerical experimentations on real-life instances of Chassaing et al. [19] which apply an evolutionary local search approach (ELS). The results indicated improvements based on a comparative study between TS-CDDARP and ELS. We also highlight that our approach is robust given various input parameters.

1 Introduction

DARP is one of vehicle routing pickup and delivery problems which is based on transportation requests where each request is associated with an origin and a destination. DARP is distinguished by the passengers transportation and especially as the handicapped persons transportation problem [4]. Furthermore, one classification is proposed in [2] including three classes for DARP. First, many-to-many problem where any node can be a source or destination for any commodity. Second, one-to-many to one problem where requests are from depots to costumers, then to depots. Third, in one-to-one problem, each commodity is a request, and has a given origin and a given destination. The famous definition is that of Cordeau and Laporte [3] where DARP consists in satisfying all requests while minimizing total travel costs under a set of constraints including vehicle capacity, maximal total duration, maximal ride time of costumers, precedence between pickup and delivery, and time windows. It is an NP-Hard problem as it is shown in [9]. Moreover, much effort has been dedicated to the development of metaheuristic solution approaches to solve this problem.

A recent work of Chassaing et al. is proposed in [14], in which authors provide an Evolutionary Local Search based approach with dynamic probabilities management in local search for DARP. The supposed model is proposed in [7], and it is defined on a complete weighted digraph with a homogeneous fleet vehicles and a set of requests. In this approach, authors minimize three criteria including the total riding time, the total duration, and the total waiting time. A relaxation of two constraints has been proposed in order to allow the search in regions of non-feasible solutions. These constraints are related to time windows and maximal ride time of costumers. They replaced them by total time windows violation and total riding time violation in the objective function. The requests insertion is made thanks to a constructive heuristic which randomly generates feasible solutions to the DARP. Results are obtained using 20 instances introduced by Cordeau and Laporte [7] who apply TS as a resolution method. A comparative study is established between ELS and TS. This comparison is based on the time efficiency for the algorithms. The dynamic probabilities management improves the convergence with a neighborhood search leading to an efficient exploration of the search space. The ELS results has indicated improvements related to TS in a rather short computation time.

This ELS approach is also tested by the same authors in new real-life framework of DARP [19]. Realistic measures for DARP are supposed on nodes' loads, maximal ride time, and time windows, contrarily to other applications that respect the offered model described in [7]. This latter considers constant loads equal to one, constant maximal ride time for all costumers, and fixed time windows on pickup nodes. However, authors in [19] supposed that loads transported by each vehicle can be greater than one and they are costumers-dependent, whereas the maximal ride time of costumers can vary related to each costumer and the distance to be crossed. In addition, time windows are proposed for either origins or destinations and they have been also considered to be variable related to costumers.

Our aim is to propose a new variant of one-to-one DARP that we call Costumers-Dependent DARP taking into account new realistic criteria. This new model aims to optimize service quality under new costumers dependent constraints. Further, we provide a new metaheustic resolution to this new variant which is TS since it was performant in the basic variant of the DARP proposed by [7]. Hence, a new method named TS-CDDARP is then applied on the CDDARP, and tested on several intances. We compare our results to those produced by the ELS of Shassaing et al. [19]. Hence, we propose a new insertion heuristic that is used to create an efficient initial routing plan for CDDARP. A new neighborhood strategy is proposed for the vehicles routing optimization. This method contributes in routing plan improvement providing several changes as well as flexibility in the transport on demand plan. We performed a comparative study between TS-CDDARP and ELS. This comparison is mainly based on the total transit time (TRT).

The paper is organized as follows: Section 2 is focused on a literature review. Section 3 is dedicated to the problem presentation and formulation. Section 4 is focused on TS based metaheuristic for computing solutions of good quality for the CDDARP. Section 5 reports the numerical results on instances from the literature with a compartive study. Finally, concluding remarks are set in Section 6.

2 Literature review

Many variants of DARP are provided in [3] to deal with various routing transportation situations depending on static or dynamic, and single vehicle or multiple vehicles. In static situations, all transportation requests are known beforehand, costumers ask for service in advance and plan is made before. Whereas, in dynamic version of DARP, initial requests are created, then requests are gradually inserted throughout the day and vehicle routes are adjusted in real-time to meet demand costumers asking for service during time. Consequently, the routing plan is updated at each time. On the other hand, single vehicle DARP supposes that all users are served by a single vehicle contrarily to multiple vehicles DARP where users are served by multiple vehicles.

One variant of DARP that is frequently used in the majority of researchs is multi-vehicles static DARP provided in [7]. It aims to design a set of least cost vehicles routes capable of accommodating all requests with precedence relationships on pickup and delivery. The objective is to minimize the total routing cost of the vehicles, the total violation of loads, and duration under time windows and ride time constraints. Tabu search algorithm is proposed by the authors as a solution methodology. A route optimization is applied to perform solution based on intra-route exchanges at every set of iterations. Computational experiments consist of randomly generating instances that contain between 24 and 144 requests. Results show the capability to produce feasible solutions as well as to improve the overall quality of the solutions.

Several DARP applications are offered based on various models that are proposed for DARP providing different objectives as well as constraints. These models are produced to deal with several DARP resolutions either in health care organization or in public transport domains. A first example is in [1] where a tabu search heuristic for solving the multi-depot, multi-vehicle, double request DARP faced by a healthcare organization is proposed. The objective function consists in minimizing a weighted sum of the two objectives including total cost and total client inconvenience time based on vehicles riding time, and the extent of earliness due to client delivery before service and late pickup after service. Constraints ensuring multiple commodity flow conservation and compatibility requirements between routes and time are proposed. Time windows constraints are embedded in a flexible lower bound and rigid upper bound. Thus a vehicle does not have to wait, but the client may wait and any deviation from time windows is penalized in the objective function. To

demonstrate the usefulness of their model, they applied it to a real-world problem in center for addictive behavior health and recovery services. To optimize its current transportation practices, a tabu search heuristic is provided. Results show an improvement of the solution due to the intensification technique for large-sized problems.

Another model of DARP is proposed in [18] to deal with emergency transportation problems in ambulance services where it is inherently to improve service quality and patients flow in crowded hospitals. The authors define a variant for DARP describing the problem as the determination of a set of routes for m salesmen whom all start from and return to a single home city (the hospital as a depot). Hence, they used a model for Travelling Salesman Problem and particularly the standard multiple Travelling Salesman Problem over an integer linear programming formulation. The objective is to minimize the total travel costs under a set of constraints related to the number of salesmen (vehicles), flows conservation, time windows, and the minimum number of costumers which are visited by each vehicle. Computational experiments are performed on real-case from a Tunisian hospital using CPLEX solver. Results show the robustness and efficiency of the methodology, especially for problems with moderate high size.

Works are severely concentrated on DARP formulations in public transportation services qualified as door-to-door problem and especially as handicapped persons transportation problems. To optimize this class of problems applications, authors focus on the importance of proposal models contributing in the costs minimization. Many factors act on the DARP results mainly based on service quality improvements and costs reductions. These factors are correlated with the number of used vehicles. For example, authors in [20], aim to maximize the average of the square of vehicle efficiency measuring the exploitation of a vehicle's capacity. Indeed, the improvement of this factor's contribution in costs reduction leads to optimize mainly features of DARP that are related to services quality including user inconvenience and users ride times. A grouping genetic algorithm is proposed to find optimal routes for transporting handicapped people in terms of service quality and number of used vehicles. Real-life instances are applied from the city of Brussels by the Inter-Communal Transport Company of Brussels. The experimentation is made in a reasonable computation time that positively acts on services qualities allowing better use of available resources. Another example aims to directly optimize the quality of services as well the case in [21] where the maximum total ride time and the total waiting time are the aim to minimize. The problem is solved by a two-step approach with firstly metaheuristics (Tabu Simulated Annealing and genetic algorithm) that generate a set of feasible routes and secondly by a bi-objective set partitioning formulation. Efficient solutions are produced under computational results that are carried out on instances taken from the literature.

3 Dial-A-Ride Problem with Costumers-Dependent criteria

DARP is a specific variant of pickup and delivery problems based on human perspective and quality of service [3]. The problem consists in planning a set of minimum cost vehicle routes capable of satisfying all requests subject to vehicle availability and side constraints related to requests, vehicles, as well as quality of service. From one-to-one DARP class, a request is defined by a pickup point, a corresponding delivery point, and a demand to be transported between these locations.

3.1 Presentation of Costumers-Dependent DARP

DARP can be formulated using several objectives [6] such as minimizing total route duration, total waiting time, or total transit time, or total distance as well as a combination of them. Effectively, constraints can be changed related to the problem's nature. Here, we assume that a vehicle may pickup more than one passenger, maximal ride times depend on costumers and distances to travel, and time windows are costumers dependent. Hence, new time windows are generated for each request related to its associated pickup node and delivery node. As it is stressed in [19], time windows vary related to customers charges and distances producing new times intervals.

The CDDARP is defined on a complete weighted digraph with a homogeneous (with the same capacity) fleet of m vehicles and a set of n transportation requests. The set of nodes $\in \{0..2n\}$ includes the depot (0), the set of pickup nodes $o \in \{1..n\}$ and the set of delivery nodes d $\in$

$\{n+1..2n\}$. The depot is represented in two copies, 0 and (n_v+1) for the depot respectively at the beginning and at the end of each trip related to a vehicle v. A request is defined by 2 nodes that are respectively a pickup node and a delivery node. We define $tr^v_{p(i),p(i+1)}$ as the transit time between two successive nodes in positions (i) and $(i+1)$ accorded to a trip of a vehicle v where n_v is his visited nodes' number. Each vehicle's tour starts from depot 0 and return to it. Each node p is visited exactly once. The origin and the destination of each request are visited by the same vehicle.

A solution (S) is a set of tours accorded to a set of vehicles. The objective is to satisfy all requests with the given resources while minimizing the total transit time. We exploit the same DARP formulation found in [17] and [14] with new assumptions for some constraints related to costumers loads (passengers to pickup or to delivery), maximal ride time, and time windows. Whereas, the remaining constraints are maintained the same for the CDDARP including precedence between pickup and delivery operations, the maximal vehicles route duration, the maximal number of vehicles to be used as well as the maximal vehicles capacity.

3.2 Costumers-Dependent DARP formulation

The objective of CDDARP consists in minimizing the sum of all transit times between nodes p in successive positions i and $i+1$ accorded to all vehicles tours in the solution and the total time windows violation. The parameter λ is initially set to 1 at the beginning of the search as it is stressed in [3], this value is then adjusted in a dynamic way. In general, this parameter increases in the case of violation otherwise it decreases. The penalty increase is expressed by $(\lambda = \lambda(1+\delta))$, whereas, in the case of decrease, we use $(\lambda = \frac{\lambda}{(1+\delta)})$ as penalty term of time windows violation. The value of δ is randomly chosen between 0.05 and 0.1 as it is used in [10]. The set of positions accorded to each vehicle's tour starts from the depot with position $(i=0)$, then a set of positions $i \in \{1..n_v\}$, and finishes at position $(i = n_v+1)$ that also represents the depot.

As it is stated by Cordeau and Laporte [7], the constraint related to time windows can be tight as a partial relaxation of the problem. Hence, we add the total waiting time as a part of the objective function. Hence, it is considered as a penalty for each time windows violation on a pickup node or a delivery node. Here, we allow a waiting time $w^v_{p(i)}$ before any beginning of service $B^v_{p(i)}$ and it is forbidden after the end of service as it is supposed in [14]. Thus, the objective function penalty in (2) is the total waiting times multiplied by penalty coefficients λ as in [14]. Given that, a waiting time on a node $p(i)$ is the difference between the begining of service $B^v_{p(i)}$ on node $p(i)$ and the arrival time $A^v_{p(i)}$ to it expressed by equation (1). The arrival time is the total transit time from the depot to the arrival node. Here, no waiting time is considered on the depot. Then, waiting times accorded to the depot are respectively $(w^v_{p(0)} = 0)$ at the departure and $(w^v_{p(n_v+1)} = 0)$ at the end of a tour.

$$w^v_{p(i)} = B^v_{p(i)} - A^v_{p(i)} \qquad \forall i \in \{1 \dots n_v\}; \quad \forall v \in \{1 \dots m\} \tag{1}$$

Let f(S) be the objective function to minimize and it is represented by equation (2).

$$Min \quad f(S) = \sum_{v=1}^{v=m} \sum_{i=0}^{i=n_v} tr^v_{p(i),p(i+1)} + \lambda w^v_{p(i)} \tag{2}$$

Vehicle's load should not never exceed the maximal vehicle's capacity (C_v) and it is supposed to be the same for all vehicles. These loads denoted by (x_o) can be greater than one passenger for all pickup nodes o $\in \{1..n\}$. the same pickup value is also the number of persons to delivery in the corresponded delivery node $(d = o+n) \in \{n+1..2n\}$. These constraints are respectively represented by equation (3) and (4).

$$x_o \geq 1 \qquad \forall o \in \{1 \dots n\} \tag{3}$$

$$\sum_{o=1}^{o=n_v} x_o^v \leq C_v \qquad \forall v \in \{1 \dots m\} \tag{4}$$

Maximum Ride Time (MRT) is the maximum time that each passenger can remain on board of the vehicle and it depends on costumers. It is represented by (L_o) given it is considered the same in [19] for a pickup node o and a delivery node d. Hence, travel time of each costumer (request) must not exceed this upper bound from his origin's position named pos(o) in a vehicle's tour v $\in \{1..m\}$ and his destination's position pos($o+n$) in the same tour. this constraint is expressed in equation (5).

$$\sum_{i=pos(o)^v}^{i=pos(o+n)^v-1} tr_{p(i),p(i+1)}^v \leq L_o \qquad \forall v \in \{1 \dots m\}; \quad \forall o \in \{1 \dots n\} \tag{5}$$

Time windows constraints are defined by lower bounds (e_o;e_d) and upper bounds (l_o; l_d) on respectively origins and destinations. Hence, vehicles service times $B_{p(i)}^v$ on nodes p accorded to positions i related to vehicles tours must respect these bounds. They are redefined in [19] taking into account maximal ride time (L_o), and costumers distances expressed by transit times between origins and destinations. Moreover, these bounds depend on costumers loads (x_o) that are supposed to be equal to times needed to load pickup node's passengers on vehicles. Time windows on origins are represented by equations(6) and (7) for respectively lower bounds and upper bounds.

$$B_{o(i)}^v \geq max(e_d - L_d - x_d; e_o) \qquad \forall o \in \{1 \dots n\}; \quad \forall i \in \{1 \dots n_v\}; \quad \forall v \in \{1 \dots m\} \tag{6}$$

$$B_{o(i)}^v \leq min(l_d - x_d - c_{o,d}; l_o) \qquad \forall o \in \{1 \dots n\}; \quad \forall i \in \{1 \dots n_v\}; \quad \forall v \in \{1 \dots m\} \tag{7}$$

Time windows on destinations are defined by equations (8) and (9) for respectively lower bounds and upper bounds.

$$B_{d(i)}^v \geq max(e_o + x_o + c_{o,d}; e_d) \qquad \forall d \in \{n+1 \dots 2n\}; \quad \forall i \in \{1 \dots n_v\}; \quad \forall v \in \{1 \dots m\} \tag{8}$$

$$B_{d(i)}^v \leq min(l_o + x_o + L_o; l_d) \qquad \forall d \in \{n+1 \dots 2n\}; \quad \forall i \in \{1 \dots n_v\}; \quad \forall v \in \{1 \dots m\} \tag{9}$$

Several experimental studies are done on benchmarked DARPs in which authors imposed time windows only on one of two costumer's nodes as it is the case in [7] and [8]. Moreover, time windows are generally fixed without considering costumers' features such maximal ride time, transit time, and loads to pickup or to delivery. A new realistic assumption for time windows must be imposed on routing plans to produce more precisions on vehicles travel time. These time windows limitations contribute in the service quality improvement as waiting time and user inconvenience (time spent in the vehicle). In fact, this time windows redefinition participates in the vehicles tours strategy while seeking for better and realistic costumers' shedule.

4 TS for Costumers-Dependent DARP

In multiple vehicles DARP, the solution space is considerably larger than in the single vehicle DARP, and needs much efforts to solve it [16]. Thus, we aim to seek for near optimal solution

to multiple vehicles DARP using TS metaheuristic. An initial solution must be generated as a first routing plan to be optimized. Furthermore, there are different procedures and heuristics in literature while seeking for initial transport on demand plans. These heuristics can produce feasible or non-feasible solutions. Heuristics construct feasible solutions satisfying all problems constraints. We cite, in this first class, random methods where initial solutions are completely random as in [3], and [7]. This heuristic starts by assigning every request to a randomly selected vehicle, and sequentially inserting requests at the end of partial routes. However, all constraints are violated except the precedence constraint and the beginning and arrival points of requests (the depot). Others non-feasible heuristics start by sorting requests based on time windows as well as the beginning of services, then by sequentially inserting requests at the end of partial routes while respecting minimum distances criteria. To obtain an initial solution to the problem, authors in [10] provide a random insertion of requests in good positions as well as vehicles if it is not possible. It seeks for the solution with a minimum number of additional vehicles and a minimum total distance.

According to heuristics that create initial feasible solutions, many authors maintain the feasibility while constructing initial solutions. For example, a regret insertion method is used [8] to construct the population as well as the initial solution. This method improves a basic greedy insertion method. It involves that the basic greedy insertion method iteratively selects the request with the lowest insertion cost, and inserts it in to the corresponding route after improving the first method. The feasibility of this insertion heuristic is examined by a multi-trip schedule procedure. Another example is in [22], where a feasible initial schedule is created for DARP. Constraints are used in the solution construction such the time window constraint which is used to cluster the customers into different groups. Other constraints are used to create initial trips from each route. The feasibility is then verified by the rest of constraints.

We investigate a new Tabu Search algorithm for the CDDARP. This algorithm is based on new efficient heuristic for the requests' insertion. This heuristic is used to initially construct the initial feasible routing plan satisfying the related constraints. The feasibility check is maintained by the heuristic in a new flexible and effective neighborhood strategy that enhances good strategic decisions about the schedule of transport on demand problems.

4.1 TS-CDDARP algorithm

TS is a metaheuristic that guides a local search procedure to explore the solution space beyond local optimality. It is characterized by flexible memory structures using strategic restrictions and aspiration levels as means for exploiting search spaces [15]. Hence, we cite basic elements of TS that we adapt for our TS-CDDARP.

Neighborhood. A neighborhood ($N(S)$) is constructed using *generate_neighborhood*() providing adjacent solutions that can be reached from a current solution. Neighbors are produced using move operations accorded to (neighborhood_length). Each neighbor is characterized by a couple of solution's attributes ($S\_attributes$) giving another objective function value for a solution (S) in the search space.
Move operation. The couple of attributes describes a change on the requests' schedule and especially the part of the routing plan that is affected by a change. This part is resumed in two promoted requests that are a random pickup node related to a request and his nearest predecessor that is also a pickup node. Details are offered in figure 2.
Tabu list. To exploit short memory in TS-CDDARP, we add a subset of moves in the tabu list (Tabu_list) as forbidden. In our TS-CDDARP resolution, we add solution's attributes in Tabu_list to not be applied for a fixed number of iterations that corresponds to the tabu list's length (tabu_length). The structure of Tabu_list is embedded in an array of elements each of them saves a couple of solution's attributes.
Long term memory. It saves best solution's attributes having participated in generating good past solutions. It is represented by a list (list_moves) of best $S\_attributes$ during the entire tabu search.
Diversification. It consists in setting Tabu_list as empty after a fixed number of iterations (divers_iter). The used method is *empty_Tabu_list*() that allows TS-CDDARP to escape from local optimums. This technique gives a possibility for $S\_attributes$ that are in Tabu_list to be selected for another time.

Aspiration criterion. When a tabu move provides an attractive objective function value of a solution (Sol_aspired) and it would result in a solution better than any visited so far, then its tabu classification may be overridden.

The main objective of TS-CDDARP in the algorithm showed in figure 1 is to construct a flexible and an efficient routing plan capable of serving all requests satisfying the constraints. Further, it seeks for a better objective function value that minimizes (2). The fundamental steps of the algorithm in figure 1 are explained below.

Algorithm 1. TS-DARP

Input: Max_iter, tabu_length, neighborhood_length, divers_iter
Output: Best

Begin

```
S_init ← insertion_heuristic ( );

/* parameters initialization
S ← S_init; Sol_aspired ← S_init; Best ← S_init;

for i=0 to i= Max_iter do
      /* diversification criterion
      if (nb_iter = divers_iter) then
            empty_Tabu_list ( );

      N (S) ← generate_neighborhood ( );
      Best_neighbor ← select_best (N (S));

      /* Aspiration criterion
      if ( f(Sol_aspired) > f(Best_neighbor) ) then
            Sol_aspired ← Best_neighbor;

      if (Best_neighbor is not in Tabu_list) then
            if ( f(Best_neighbor) < f(S) ) Then
                  S ← Best_neighbor;
                  /* Tabu long term memory
                  add (list_moves, S_attributes);
                  update (Tabu_list);
end for

if ( f(S) < f(Best) ) then
      Best ← S;

if ( f(Sol_aspired) < f(Best) ) then
      Best ← Sol_aspired;
```

End

Fig. 1. TS-CDDARP.

- A greedy construction of the initial solution (S_init) is the result of the insertion heuristic named *insertion_heuristic*() of TS-CDDARP that provides a feasible initial vehicles' routing plan.
- A set of feasible routing plans is created in the neighborhood $N(S)$ by way of the procedure *generate_neighborhood*(). In fact, a request can change from a route to another during the neighborhood strategy. Hence, vehicles travel times also change according to new requests routes' positions.
- Local searches are executed during TS-CDDARP process seeking for (Best_neighbor) found in each $N(S)$. If this local optimum is not in Tabu_list, and has a better objective function (2), then it becomes the current solution (S). Hence, it is added to Tabu_list as forbidden.

Consequently, the procedure *update*($Tabu\_list$) updates the tabu list by adding or removing solutions attributes according to neighbors input or output.

- A diversification technique is made thanks to the procedure *empty_Tabu_list*() that gives opportunities to requests to be allocated to other vehicles routes in other routing plan situations.
- The use of the long term memory is made thanks to the procedure *add*($list\_moves, S\_attributes$). It lets us the save of the moves history that contributes in the vehicles routing plan improvement.
- At the end of the process when the maximal number of iterations (Max_iter) is yet achieved, the best solution (*Best*) is given as a better solution during all search spaces. The solution *Best* can be changed by the aspired solution (Sol_aspired), if it can be improved by this one.

4.2 Insertion heuristic of TS-CDDARP

As it is stated before, the solution of the present TS-CDDARP is embedded in a set of tours from the depot to it accorded to a set of vehicles. Each tour represents a subset of satisfied requests. We propose a new heuristic *insertion_heuristic*() that creates an initial solution S_init producing a feasible routing plan for the CDDARP. This solution is feasible while satisfying all the problem constraints except time windows constraints that are considered as a penalty for the objective function (2). The steps of *insertion_heuristic*() are resumed below.

- The request's allocation starts by successively dispatching vehicles while their capacities are not violated with the respect of the maximal tour's duration. At the departure, the heuristic starts by allocating nearest pickup nodes to each dispatched vehicle. It also initializes each vehicle's travel time by a transit time between the depot and the first allocated pickup node. For each dispatched vehicle, this node is added to the vehicle's tour and becomes the current one.
- Each dispatched vehicle continuous his tour by seeking for the nearest non-visited pickup nodes to the current one while constraints on the vehicle's capacity (4) and the maximal vehicle's duration are not violated. Moreover, transit times $tr^{v}_{p(i),p(i+1)}$ are computed between current $p(i)$ and successor $p(i+1)$ according to vehicles' tours.
- Constraints (5), (6), and (8) are controlled by computing each arrival time accorded to each request in a vehicle's tour. In the case of MRT violation, a dispatched vehicle proceeds to the delivery operation related to each request having minimal MRT. This technique allows us to ensure that no costumer exceeds his MRT.
- Due to arrival times computation, we also deduct possible time windows violations expressed by constraints (8), and (6). Then, waiting times are derived from equation (1) and applied in the penalty term of the objective function (2).
- A vehicle's tour is interrupted if its capacity constraint (4) is violated or its total route's duration is at the maximum. In the case of the capacity violation, the vehicle delivers all pickup nodes that are visited in the tour while respecting the nearest non-delivery node technique. These nodes are added to the tour while computing additional transit times to the vehicle's travel time.
- The end of the procedure *insertion_heuristic*() is allowed when requests are totally satisfied. Otherwise, dispatching vehicles will try to pickup maximal number of nodes in their routes respecting residual capacities updated after each pickup or delivery operation. However, a vehicle tour termination is set when the maximal route's duration is achieved.

This heuristic is efficient and allows adjustments to other situations in transport on demand problems since it is based on an iterative requests' allocation of n requests to m vehicles in a polynomial time θ(n,m).

4.3 Neighborhood strategy for TS-CDDARP

The neighborhood strategy of TS-CDDARP is an optimization technique that operates tactical changes on the vehicles routing while transporting users from origins to destinations. It aims to make flexibility to the plan of requests by introducing some orders while dispatching vehicles. We focus on the fact that any pickup node can be a successor for multiple other pickup nodes. However,

this node can be the most nearest node for many other pickup nodes. But, according to the general DARP definition, it is allocated to only one node during a vehicle routing. Then, it becomes visited and forbidden to be chosen by other vehicles. In this situation, a change in the vehicles routing that consists in allocating a random pickup (*rand-pk*) node to a nearest predecessor (*pred*) which is a pickup node visited by another vehicle. To deal with this move operation, we maintain *rand-pk* as a visited node until encountering pred. Hence, no node that is different from *pred* can be a predecessor to *rand-pk*. Figure 2 describes the neighborhood strategy that contributes in the solutions improvement of TS-CDDARP .

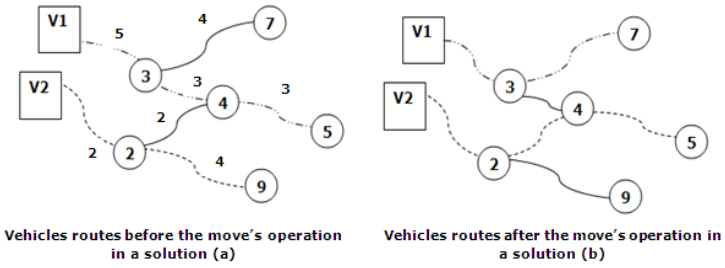


Fig. 2. Neighborhood strategy for TS-CDDARP

In the example of figure 2, we suppose to have two separated vehicles routes from a current solution (a) produced by TS-CDDARP. These routes have various distances. The pickup nodes 3, 4, and 5 are visited by vehicle V1 and the pickup nodes 2, and 9 are visited by vehicle V2.

Move's attributes are a random pickup node *rand-pk* and a nearest predecessor *pred* to it. Hence, we suppose that node 4 is randomly chosen to be removed from its corresponding vehicle's route. Node 4 is chosen as a nearest successor for node 3 in solution (a). But, it is also a nearest successor to node 2 in vehicle's route of V2 in (a). This node is not allowed to be chosen by V2 after node 2 in the solution (a). After a move operation, node 4 changes from route V1 to V2 given it is also another alternative to optimize vehicles routing by TS-CDDARP producing new travel times. Consequently, a new neighbor solution (b) is provided in figure2 where V1 visits node 3, and 7. Whereas vehicle V2 visits 2, 4, and 5 in the same neighbor (b). We obtain new routing plan for TS-CDDARP as well as new total transit time in the objective function of equation (2). The neighborhood strategy allows vehicles routing plan diversification and enhances new TS-CDDARP solutions to appear in various local searches. Hence, each new solution is qualified by a couple of attributes that are respectively *rand-pk*, and *pred* in the search space.

5 Experimental study

We are based on instances proposed by Shassaing et al. in [19] having realistic data such as distances between regions. The data format based on geographical measures allows us to adapt the benchmark to future real-case studies. The instances are relatively similar to those proposed by Cordeau and Laporte in [3] with specific characteristics. Distances between nodes are supposed as geographical measures, customers are located on big areas varying between 100km and 8500 km, vehicles speed is supposed to be constant equal to 1.33km/min. Maximal travel of tours is fixed to 480, vehicles capacity is 8, and loads to charge or discharge are between 1 and 4. To validate our TS-CDDARP algorithm, series of preliminary tests have been run over 1000 TS-CDDARP iterations using small instances of [19]. The tabu list length is supposed equal to 7 with a neighborhood length equal to 5. Experimental results in table 1 shows objectives functions related to selected instances from [19].

Because of the focus on the satisfaction as soon as possible of customers requests, we limit our study to the total transit time TRT comparison between results of ELS and those produced by TS-CDDARP. Comparing the results of the objective function (2) reached by the TS-CDDARP algorithm showed in table 1, we can easily see the improvement of the TRT provided by TS-CDDARP. The comparison is established between the total transit time (TRT_TS) provided by

Table 1. Results of TS-CDDARP on instances of [19]

Instances	m	2 * n	TRT_TS	TRT_ELS
SF_RL_d1	7	92	2074.359	2178.423
SF_RL_d10	4	68	1033.963	1524.163
SF_RL_d70	6	78	1614.723	1842.603
SF_RL_d92	2	34	257.923	590.998
SF_RL_d93	2	40	323.611	568.202
SF_RL_d94	2	46	353.287	506.644

TS-CDDARP and the total transit time (TRT_ELS) provided by ELS in [19]. Consequently, we outpoint that the move's operation proposed in figure 2 positively acts on the results given by the improvement seen in the objective functions values related to the instances. We highlight the routing plan adaptation of our TS-CDDARP resolution on various instances presented in table 1 and his robustness given various input parameters such as vehicles' number and costumers' number.

6 Conclusion

This paper proposes a new tabu search algorithm called TS-CDDARP for new real-life benchmark instances of DARP. We have investigated a new approach to produce flexible routing plan that satisfies customers requests as soon as possible. We have operated a new operation move to deal with more flexibility while dispatching vehicles seeking for better routes to fulfill as well all requests. Good results were produced for many instances showing improvement in total transit times related to those of ELS. Our method was validated with various experimental tests of TS-CDDARP. Our model can be expanded to include uncertain elements related to time windows as well as travel times between nodes. Future research may accommodate request changes dynamically through real-time framework.

References

1. E. Melachrinoudis, H. Min (2011). A tabu search heuristic for solving the multi-depot, multi-vehicle, double request dial-a-ride problem faced by a healthcare organisation, International Journal of Operational Research, 10(2), 214-239.
2. A. Gerardo Berbeglia, J. Cordeau, G. Laporte (2010). Dynamic pickup and delivery problems, European Journal of Operational Research, 202, 8-15.
3. J. Cordeau, G. Laporte (2003). The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms, Quarterly Journal of the Belgian, French and Italian Operations Research Societies , (1), 89-101.
4. P. Toth, D. Vigo (1996). Fast local search algorithms for the handicapped persons transportation problem. Meta-heuristics: theory and applications (pp. 677-690). In I. H. Osman J. P. Kelly (Eds.), Boston.
5. A. Lemouari, O. Guemri (2014). A Multi Objective TS-Method for the Static DARP, International Conference On Metaheuristics And Nature Inspired Computing.
6. J. Cordeau, G. Laporte, (2007). The Dial-a-Ride Problem: Models and Algorithms, Annals of Operations Research, 153(1), 29-46.
7. J. Cordeau, G. Laporte (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem, Transportation Research Part B, 37, 579-594.
8. Z. Zhenzhen, L. Mengyang, L. Andrew (2015). A memetic algorithm for the patient transportation problem, Omega, 54, 60-71.
9. P. Healy and R. Moll (1995). A new extension of local search applied to the Dial-A-Ride Problem. European Journal of Operational Research, 83(1), 83-104.
10. N. Sophie, F. Doerner, F. Hartl (2010). Variable neighborhood search for the dial-a-ride problem, Computers Operations Research, 37(6) 1129-1138.
11. S. Ropke, J. Cordeau, G. Laporte (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows, Networks, 49 (4), 258-272.
12. J. Cordeau (2006). A branch-and-cut algorithm for the dial-a-ride problem, Oper. Res, 54 (3), 573-586.
13. B. Kris, A. Caris, K. Janssens (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots, Transportation Research Part B: Methodological, 67, 166-186.

14. M. Chassaing, Ph. Lacomme, Ch. Laforest (2016). An ELS-based approach with dynamic probabilities management in local search for the Dial-A-Ride Problem, Engineering Applications of Artificial Intelligence, 48, 119-133.
15. F. Glover (1995). Tabu Search Fundamentals and Uses, Revised and Expanded, Technical Report, Graduate School of Business, University of Colorado, Bolder, CO.
16. G. Berbeglia (2009). Complexity Analyses and Algorithms for Pickup and Delivery Problems, HEC MONTRAL university.
17. J. Paquette, J.F. Cordeau, G. Laporte, M. M. Pascoal, (2013). Combining multicriteria analysis and tabu search for dial-a-ride problems, Transportation Research Part B: Methodological, 52, 1-16.
18. J. Jlassi, J. Euchi, H. Chabchoub (2012). Dial-a-ride and emergency transportation problems in ambulance services, Computer Science and Engineering, 2(3), 17-23.
19. URL : http://fc.isima.fr/ chassain/Phd/Reallifeinstances.php
20. B. Rekiek, A. Delchambre, H. Saleh (2006). Handicapped person transportation: An application of the grouping genetic algorithm, Engineering Applications of Artificial Intelligence, 19(5), 511-520.
21. F. Guerriero, F. Pezzella, O. Pisacane, L. Trollini (2014). Multi-objective optimization in dial-a-ride public transportation, Transportation Research Procedia, 3, 299-308.
22. Z. Xiang, C. Chu, H. Chen(2006). A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints, European Journal of Operational Research, 174(2), 1117-1139.

Optimal urban traffic management at the crossroads "Rond Point Hassan II", in Casablanca by genetic algorithms

Amal MERBAH[1], Abdelilah MAKRIZI[1], and EL Hassan ESSOUFI[1]

[1]Laboratory of Mathematics, Computing and Engineering sciences, Faculty of Sciences and Technics, SETTAT Hassan First University, Km 3, POB: 577 Route de Casablanca
merbah2007@gmail.com
a_makrizi@yahoo.fr
e.h.essoufi@gmail.com

1 Abstract

We develop a multi-objective mixed integer nonlinear programming model minimizing the three criteria: the waiting time (R) and the number of passenger cars (NV) , the waiting time of Tramway (R_{tram}) . By taking account of the constraints bound to the security of the users : the release red, the maximal and minimal borders on the durations of the red and the green , and constraints bound to the structure of the crossroads : antagonism , the absolute priority of Tramway (tram). In the application of our model on a crossroads "Rond Point Hassan II" (C) in Casablanca, due to the complexity caused by the consideration of several criteria and constraints, the number of variables and the non-linearity of certain equations, the use of the genetic algorithm is imperative and supplies satisfactory real time results compared with the semi adaptive system applied to the crossroads (C) .

Key words: urban traffic in Casablanca, multi-objective optimization, genetic algorithm.

2 Introduction

The literature enriched by methods and systems allowing to regulate road traffic in urban areas of crossroads on fires on real time , from the measures supplied by sensors places on crossroads. The way of using these measures to regulate the traffic, the taking into consideration of the constraints , the criteria to be minimized and the methods of resolution are to differ from a system in the other one. In [1] the only criterion is the waiting time and the constraints are: borders on the durations of the phases and antagonists' notion, the variables are : the successive phases apply to the system. In [2] the criterion is the queue length, the variables are: the start time of the green and the duration of green . Model in [3] is based on a mesoscopic modeling, the variables are: the sequence of passage groups. In [4] shows that acyclic systems are more flexible than the cyclic systems and develops three models of three criteria and variables different from a model to the other one , by basing itself on methods exact for the resolution.
In the development of a regulatory system similar to the Moroccan reality we present our first model in section 3 minimizing the three criteria: the waiting time (R), the number of passenger cars (NV), and the waiting time of Tramway (R_{tram}), using a multi-objective approach, adapted to operate in real time. In Section 4 we present a comparison of the results provided by our system of regulation determined by the genetic algorithm with the semi adaptive system applied to the crossroads (C).

3 mathematical model

3.1 key variables

– the date of the p^{ième} passage of fire f ,from the moment : $d_{p,f}$

– The fire is presented by state :

$$\begin{aligned} x_{p,f} &= 0 \quad \text{if the fire is red} \\ &= 1 \quad \text{if the fire is green} \end{aligned}$$

3.2 additional variables

$xa_{tram} = 1$ if and only if $ta_{tram} \in [dp, dp + 1]$
along with ta_{tram} the moment of arrived from the tramway and tq_{tram} the moment of its departure.

3.3 objective function

$$FA = \min(\alpha R + \beta NV + \rho R_{tram}) \tag{1}$$

$$R = \sum_{f=1}^{Nf} \sum_{p=1}^{NP} 1/2 * (l_{p,f} + l_{p-1,f}) * \min(\Delta d_p, \Delta d^0_{p,f}) \tag{2}$$

$$l_{p,f} = \max(l_{p-1,f} + (DA_f - (x_{p-1,f} * DS_f)) * (d_{p,f} - d_{p-1,f}), 0) \tag{3}$$

$$NV = \sum_{f=1}^{Nf} \sum_{p=1}^{NP} NV_{p,f} \tag{4}$$

$$NV_{p,f} = (DA_f * \min(\Delta d_p, \Delta d^0_{p,f})) + \int_{d_p}^{min(d_{p+1}, d_p + \Delta d^0_{p,f})} l_f(t)dt \tag{5}$$

$$\Delta d_p = d_{p+1} - d_p \tag{6}$$

$\Delta d^0_{p,f}$ is the time required to sell the queue $L_{p,f}$ on the section f to leave of $d = d_p$. This time is in the green case$l_{p,m}/DS_{p,m}$ and is worth more infinity in the case of red.

$$R_{tram} = (tq_{tram} - ta_{tram}) \tag{7}$$

3.4 Constraints

– antagonists

$$x_{p,f} + x_{p,f'} = 1 \tag{8}$$

– Borders on the minimal and maximal duration of the red

$$DRmin * (1 - x_{p,f}) \leq d_{p+1} - d_p \leq DRmax * (1 + M * x_{p,f}) \tag{9}$$

– Borders on the minimal and maximal duration of the green

$$DVmin * x_{p,f} \leq d_{p+1} - d_p \leq DRmax * x_{p,f} + M(1 - x_{p,f}) \tag{10}$$

– The red of release DR

$$\min(d_{c,f'} - d_{p,f}) = (x_{p-1,f} - x_{p,f}) * DR \tag{11}$$

– The priority absolved from the Tramway with f_j the set N fires of the crossroads containing the Tramway

$$\sum_{j=1}^{n} x_{p+1,fj} <= (1 - x_{tram}) \tag{12}$$

$$d_{p+1,fj} = d_{tram} \tag{13}$$

–

$$x_{p,f} \in 0, 1 \tag{14}$$

4 Application Model

4.1 The crossroads (C)

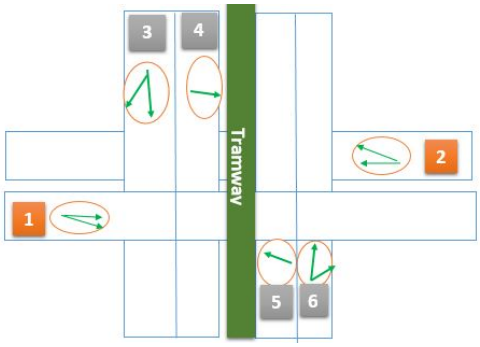


Fig. 1. The crossroads (C) drawn with the path of the Tramway

Fig. 2. Structure of the crossroads (C) from Google Maps

The crossroads (C) (See Figure 1 and 2) Consists of six fires using a semi S adaptive system with durations of phases depends on the hours of the day as Follows :

Table 1. The durations of the phases of the fires of the crossroads (C)

Hour	peak hour	Other hours
Fires 1 et 2	50	35
Fires 6 et 3	40	40
Fires 4 et 5	20	20

4.2 The input data

See Table 2

Table 2. The values supplied by the sensors and the constants

Variable	Symbol	Value
The minimal and maximal duration of the green for fires 1 and 2	$DV_{min}DV_{max}$	25 45
The minimal duration of the green for the other fires	$DV_{min}DV_{max}$	10 20
the red of release	DR	8
The lengths of the queues initial expectations on fires	$l_{p-1,f}$	0 0 0 0 0 0
The debits of entrances at the hours of points	DA_f	0.5 0.5 0.2 0.2 0.2 0.2
The debits of entrances at the other hours	DA_f	0.3 0.3 0.1 0.1 0.1 0.1
The debits of exits of six fires The output rates of the six lights	DS_f	0.4 0.4 0.3 0.3 0.3 0.3
The state of fire between d_{p-1} et d_p	$x_{p-1,f}$	0 0 0 0 1 1
The last switching dates	d_{p-1}, f	8
The arrival time of the tram during the cycle	ta_{tram}	20 50 80

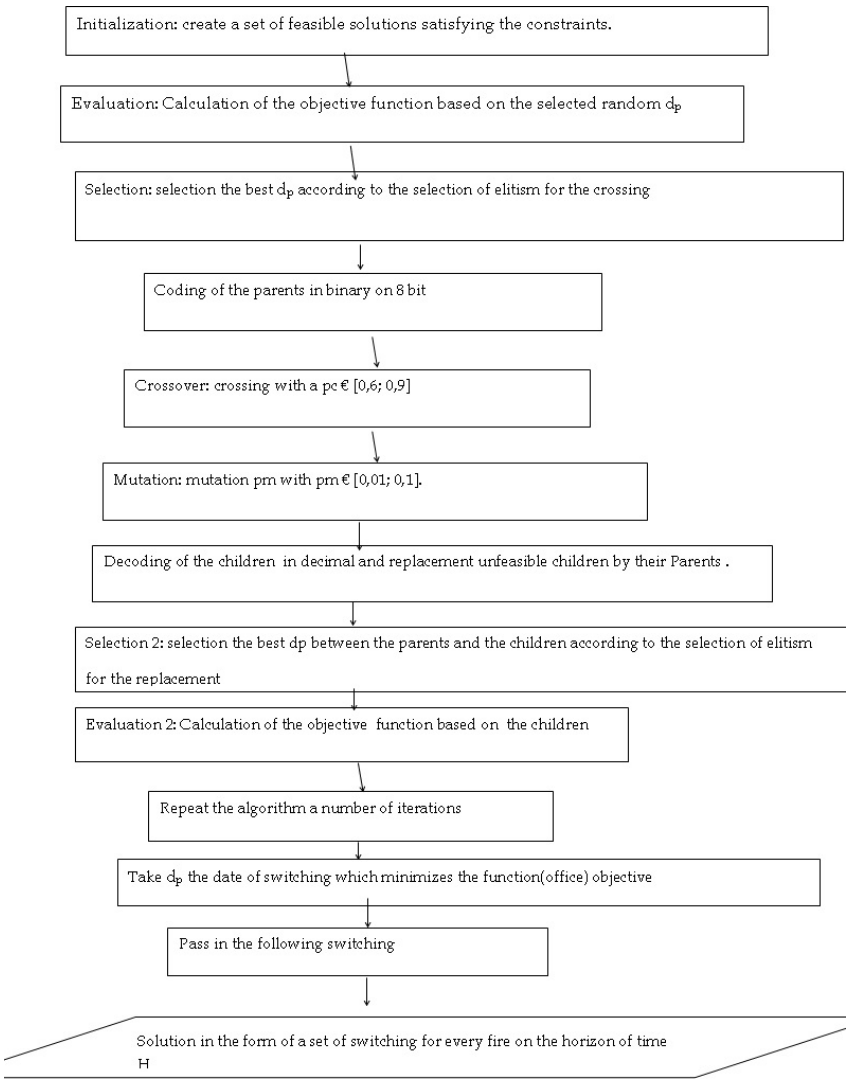


Fig. 3. The Scheme of the applied genetic algorithm

4.3 The genetic algorithm applied (GA)

See Figure 3

4.4 Comparison

Our model is solved by the GA applied to the crossroads (C). The table 3 gives a comparison of the objective functions found by both systems and shows that the GA gives the best results compared to semi-adaptive system S.

Table 3. Comparison of FA supplies by the AG and S in seconds

	FA supplied by S	FA supplied by AG	Difference
Dense debit	4688	1561	3127
	4688	984	3704
Fluid debit	1539	777	762
	1539	829	710

5 Conclusion

The genetic algorithm GA is best suited to solve our model of urban traffic management at the crossroads (C), and provides satisfactory results compared to the semi adaptive system. The generalization of the model on a network of crossroads, the use of the appropriate evolutionary methods to the optimization multiobjective will be our next works.

References

1. L. MAGNE *Commande optimale décentralisée du trafic urbain* L'univeristé d'Evry Val d'Essone et L'institut national de recherche sur le transport et leurs sécurité, 2001.
2. S. Kachroudi *Commande et optimisation pour la régulation du trafic urbain multimodale sur de grands réseaux urbains.* L'école nationale supérieure de l'aéronautique et de l'espace, 2010.
3. F. YAN *Contribution à la modélisation et la régulation du trafic aux intersections: Intégration des communications Véhicule-Infrastructure.* Université de Technologie de Bélfort-Montbéliard, 2012.
4. Y. DUJARDIN *Régulation adaptative multi-objectif et multi-mode aux carrefours à feux.* Université de Paris-Dauphine, 2013.
5. D. Hermawanto *Genetic Algorithm for Solving Simple Mathematical Equality Problem* Indonesian Institute of Sciences (LIPI), Indonesia

Towards Optimised Large Scale Deployments of Electric Bus Systems with On-Route Charging

Marcin Seredynski[1], Djamel Khadraoui[1], and Francesco Viti[2]

[1] Luxembourg Institute of Science and Technology,
Esch-sur-Alzette, Luxembourg
firstname.lastname@list.lu
[2] University of Luxembourg
Luxembourg, Luxembourg
francesco.viti@uni.lu

1 Introduction

Sustainable urban development motivates investments in environment-friendly and user-centred Public Transport (PT) services. Three trends towards next generation PT systems are observed. The first one is an introduction of greener vehicles such as battery electric/hybrid buses (e-buses). The second one focuses on high service quality (e.g. increased ride comfort via mitigation of stop-and-go driving pattern). The last trend is in reduction of emissions and bus operating costs related to energy consumption and equipment wear and tear during bus operations. These trends however bring new real-time optimisation challenges. The first one is created by different operational characteristics and constraints of e-buses, e.g. they need to periodically recharge batteries at charging stations placed in terminals and optionally in bus stops. This brings additional complexity into PT operations and its cost dynamics. Since several PT operators share the same e-charging infrastructure and complex electricity pricing schemes are usually applied (e.g. peak demand charges) optimisation of charging plans is necessary. The second challenge, relating trends number two and three, is how to provide comfort- and cost-related benefits without negatively impacting general traffic performance. Relying solely on strategies such as Transit Signal Priority (TSP), which prioritise PT vehicles at signalised intersections, might cause congestion effects that could backfire on the PT system itself. In this paper we demonstrate how these challenges can be addressed by the emerging Cooperative ITS (C-ITS) methods. A novel framework addressing the problem via combination of cooperation, negotiation and multi-objective optimisation is introduced.

The next section presents current situation of e-bus operations and research problems currently addressed in the literature. This is followed by an overview of our approach solving the problems of the next generation PT systems.

2 Current situation

The first above-mentioned challenge—optimal recharging—is not yet addressed in the literature, partly due to the fact that current e-bus systems are still deployed in small-scale trials. The existing approaches lack the required degree of modelling detail necessary to capture the complex interactions emerging between bus operations and charging infrastructure. The existing research efforts currently focus on developing a proper system design such as deploying strategic locations of charging stations [1, 2]. Energy efficiency is also addressed via energy management strategies for the engine [3], and regenerative breaking technologies [4]. The second challenge—support of a bus in operations—also requires new research efforts as, due to technological limitations, the existing methods enhancing PT operations such as Transit Signal Priority (TSP) [5] and holding strategies [6] have been designed to only support the simple punctuality objective. Hence, they cannot handle the complexity of next generation systems, which not only need to be "on time", but also have to be comfortable, cost-efficient, and allow e-buses to recharge battery on-route without impacting the service level.

2.1 Solution

We jointly address constraints and control capabilities of all entities of the PT ecosystem, which consists of signal control, (e-)buses, and e-bus charging infrastructure. The methods that we developed combine cooperation and negotiation between all ecosystem actors thanks to connectivity, in order to effectively achieve mutual goals. Such connectivity—provided by the emerging Connected Vehicle (CV) technology—allows us to shift from currently used standalone ITS systems such as TSP to the C-ITS paradigm. In particular, we use Signal Phase and Timing (SPaT) [7] information obtained from traffic signal controllers to switch the PT-oriented control emphasis from signal-centric strategies (TSP) towards vehicle-centric. Our approach has two layers–the first one represents infrastructure of an e-bus system—coordinated signal control managed by a traffic management centre and charging infrastructure. The second layer corresponds to the cooperative environment composed of interconnected vehicles. The goal of our approach is to optimise the interplay between three interacting components—in-vehicle control managed by a Cooperative Bus System (CBS), signal control and a centralised PT backoffice system. Key element of our solution is a novel energy-aware Automated Vehicle Location system (eAVL) hosted by the backoffice. Unlike traditional AVL, not only it collects information about fleet locations, but also about battery status of each e-bus. Consequently, the backoffice, via eAVL assures cost-efficient use of e-charging infrastructure with minimisation of impact on operations—optimised e-charging schedules are feed to individual buses. These schedules are used as constraints to the CBS systems of buses. Each CBS uses a Cooperative Driver Assistance System (C-DAS) which aims at similar objectives as TSP (reduced stop frequency at signalised interesections) via optimisation of speed and dwell time profiles. The latter include time required to recharge batteries. Traffic signals broadcast not only SPaT information, but also level of available PT priority (negatively correlated with general traffic demand). This means that a CBS knows what is the "priority cost in a given time. The priority is requested via negotiation. For this, the Priority Request Server (PRS) component of TSP is extended with negotiations capability with the TSPs Priority Request Generator (PRG) located in a PT vehicle. PRS will receive information about alternative scenarios varying in arrival time. Via the negotiation, the two control domains try to iteratively reach an agreement in terms of green time allocation for PT. The optimisation of charging schedules and speed/dwell time profiles will be carried out using multi-objective optimisation algorithms [8].

Acknowledgment

This work has been founded by MOBITRAFF Project (INTER-PolLux programme) financed by National Research Fund (FNR, Luxembourg) and National Centre for Research and Development (NCBR, Poland).

References

1. G. Fusco, A. Alessandrini, C. Colombaroni, and M. P. Valentini, "A model for transit design with choice of electric charging system," *Procedia - Social and Behavioral Sciences*, vol. 87, pp. 234–249, 2012.
2. M. Pternea, K. Kepaptsoglou, and M. Karlaftis, "Sustainable urban transit network design," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 276—291, 2015.
3. J. Peng, H. He, and R. Xiong, "Rule based energy management strategy for a series–parallel plug-in hybrid electric bus optimized by dynamic programming," *Applied Energy*, 2016 (in press).
4. L. Li, Y. Zhang, C. Yang, B. Yan, and C. M. Martinez, "Model predictive control-based efficient energy recovery control strategy for regenerative braking system of hybrid electric bus," *Energy Conversion and Management*, vol. 111, pp. 299–314, 2016.
5. J. Hu, B. B. Park, and Y.-J. Lee, "Coordinated transit signal priority supporting transit progression under connected vehicle technology," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 393–408, 2015.
6. O. Ibarra-Rojas, F. Delgado, R. Giesen, and J. Munoz, "Planning, operation, and control of bus transport systems: A literature review," *Transportation Research Part B*, vol. 77, no. 38–75, 2015.
7. M. Seredynski, D. Khadraoui, and F. Viti, "Signal phase and timing (SPaT) for cooperative public transport priority measures," in *Proc. 22nd ITS World Congress*, 2015.
8. E.-G. Talbi, M. Basseur, A. J. Nebro, and E. Alba, "Multi-objective optimization using metaheuristics: non-standard algorithms," *International Transactions in Operational Research*, vol. 19, pp. 283–305, 2012.

A Hybrid Bin-Packing algorithm for solving the yard optimization problem.

Razouk Chafik [1], Youssef Benadada [2], Jaouad Boukachour [3]

[1] PhD Student at ENSIAS Cedoc ST2I – ROL, Rabat, Morocco
[2] PhD, Professor at ENSIAS Team Leader of Operations research and logistics, Rabat, Morocco
[3] PhD, Associate Professor of Computer Science University of Le Havre, Le Havre, France
{Chafik.razouk@gmail.com, yss.Benadada@gmail.com, jaouad.boukachour@univ-lehavre.fr}

Abstract. Maritime container terminals are facilities where cargo containers are transshipped between ships or between ships and land vehicles (tucks or trains). They can be automated; semi-automated or static ones depend on the type of the equipments that they contain. There are crucial resources at container terminals; the yard, cranes and the vehicles. The yard refers to the temporary storage area where inbound, outbound containers can be stored. While we have two kinds of cranes depending on the type of the yard we are working on: yard crane and quay crane. Vehicles are used to transfer the container between the seaside and yard side (the external trucks XTs are not considerable in our study).

The decision of the container stacking position is the most important operational task for the incoming containers which affects not only the productivity of the stacking but also for the later retrieval and also to avoid the unproductive moves to access to the requested containers stored in the stack. Our aim is to define a new approach to design the container stacking problem using an approach of optimization and simulation. Thus in this paper we will define a new MIP which has as objective the minimization of the number of the stacks used to store a given number of inbound containers, and to define afterwards a simulation model using a simulation software (Arena) which has as input the result of the optimization model. So we will try first to define a relocation model using the standard (FCFS) then the Best fit decrease (BFD) from the bin packing algorithm to store and maintain the storage area and to avoid unproductive moves and reshuffles to find out the requested container sequence.

The main inputs are high, weight, destination and the delivery time of each container. Our objective is to have, based on heuristics, a guide to the planner to obtain an optimized stacking / reshuffling plan, given a stacking state and a container demand. We will try to minimize the number of reshuffles as well as the number of stacks used and obtaining a conclusion about possible yard configurations.

Keywords : Containers terminals; optimized plan; reshuffle; container stacking problem; heuristics; optimization; simulation.

1. Introduction:

Maritime transportation has increased remarkably over the last few decades according to Steenken et al. (2004) **[1]**, the use of containers reduces the amount of the product packaging and the possibility of damage.

A container terminal is an interim storage area, where container vessels dock on berths to unload inbound containers during the import operations and load outbound containers for the export operations. To be more competitive containers has been standardized in the world wide, an ISO standardized metal box, so the most famous ones are: 20feet, 40feet (TEU : twenty feet equivalent units) . They can be stacked then on top of each other following a 'last-in, first-out' (LIFO) strategy. Once the container will be transferred from the seaside to the yard side via internal vehicles, the yard crane will store this container in a specific location based on a certain criterions: destination, estimated departure, weight, size and destination. Figure 1 illustrates the complete process of typical operations and equipment in container terminals, including quay cranes for loading and unloading containers from vessel, internal vehicles for carrying containers within the terminal area and RTGC (Rubber Tyred Gantry Crane) for stacking containers in the storage yard.

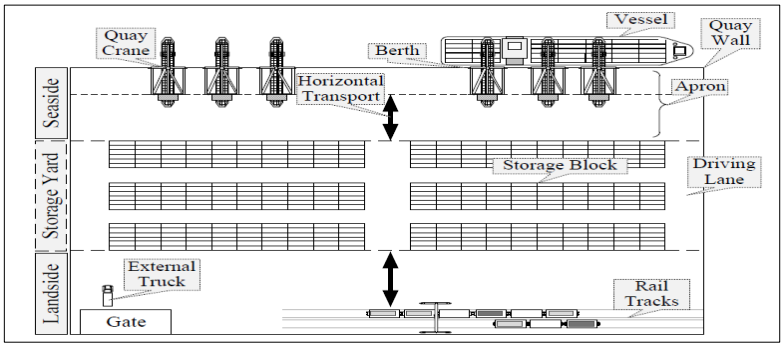


Fig.1. Container Terminal Main Area

In order to access a container which is not at the top of its stack, those above it must be relocated. It occurs since other ships have been unloaded later or containers have been stacked in the wrong order due to lack of accurate information. This reduces the productivity of the cranes. Maximizing the efficiency of the current process leads to several requirements:
1. Each incoming container should be allocated a location in the stack which should be free and supported at the time of arrival. This location can be identified either by the worker or by the state of the stack before each planning horizon.
2. Each outgoing container should be easily accessible, and preferably close to its unloading position, at the time of its departure. This process can be also optimized by storing containers of the same vessel in the same yard bay for example.

The number of containers handled by the container terminals is growing then the competition between containers terminals is increasing. The using of the scientific methods and operational research tools became crucial in container ports.

According to Stahlbok and Voss (2008) **[2]** activities in a containers terminal are conducted in two main areas: the quay and the yard area. In the quay area, we can found various optimization problems discussed in the literature such as:

- Berth allocation problem (BAP);
- Container storage;
- Quay crane scheduling;

Others one in are interested to the yard area:

- Yard crane scheduling;
- Storage Yard Planning;
- Container truck scheduling;

Based on this, Rashidi and Tsang (2013) **[3]** divided the operations in container terminals into 5 operations as in Figure 2 that effect the efficiency of the problems in container terminals:

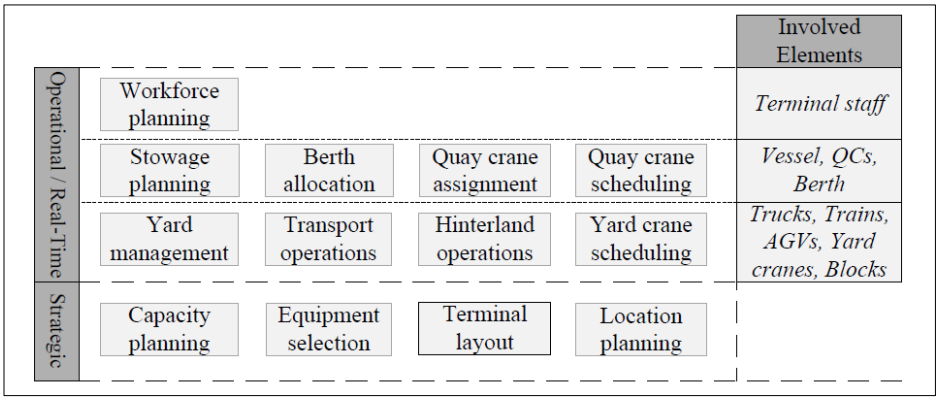


Fig.2. Decision problems in container terminals **[4]**

The majority of decision containers terminal can be divided into three levels of planning and control: 1) strategic, 2) Tactical and 3) operational. Strategic decisions are long-term decisions that include the structure of the terminal, handling equipment and handling procedures. Tactical decisions are medium-term decisions that are interested in determining the number of quay cranes, yard cranes, straddles, etc. And finally operational decisions are short-term decisions and include the process to be followed by the quay cranes, the yard cranes, straddles, etc. In this paper we will focus on the operational operations, so the objective is to have storage containers planning to be flowed by gantry cranes in non automated container terminals for the inbound containers by minimizing the total distance and the reshuffles number in the yard area.

The rest of this paper is organized as follows: Section 2 presents a literature review, section 3 defines the problem and section 4 presents the solving methods to our MIP. The results of numerical simulations and a comparative study are presented in Section5. Finally Section 6 is devoted to conclusion.

2. Literature review:

In this paper we focus on the storage space assignment already presented in the previous section and study the existing research for container stacking positions for import operations in automated containers terminal. Each yard area had a specific level to store containers; containers on the top of the stacks must leave earlier to avoid unproductive movements (reshuffles). These moves have a negative effect on the operational efficiency of the container terminal. Therefore, while determining the stacking positions of the containers, the objective of the problem is defined to minimize the global distance between the sea side and the yard area and also to reduce the number of the reshuffles.

The factors involved in determining the stacking plan are: stack level, stacking strategies for storage and retrieval containers for import, weight of containers, size and EDT. In our study to stack containers, the RTGC will be used to pick up the container from the internal vehicle to deliver it in the stacking position**.**

According to Steenken et al. (2004) and the extended study of Stahlbock and Voss (2008) the main logistics operations at container terminals can be classified as follow: Berth allocation, stowage planning, crane scheduling, terminal transport optimization, and storage and stacking logistics. There are static and stochastic cases of containers. In the static storage, the location of each container in the storage yard was allocated and reserved before the vessel arrival. The reservation for containers could be based on discharge port, container type and container weight, depending on the stacking strategy. On the other hand, stochastic case, container's location was not determined before the vessel's arrival. Instead, the storage location was determined in real time and after the vessel berthed. The EDT and the container destination are not taking into account for the inbound containers in this study.

According to Carlo et al. (2014) **[5]** the containers stacking problems can be divided into three types: storage of individual containers, of group of containers and the storage space assignment in relation with the other yard process. we will introduce under each type the related researches.

2.1 Storage space assignment of individual containers:

The main idea of this type of research is to first assign containers to blocks and secondly to specific locations within the selected block. Guldogan et al. (2010) **[6]** considers an integrated policy that takes into account the work balance, number of used trucks and the travel distances, the aim of the new strategy is to define clusters which group containers according to their departure dates, weight and type of the stored containers are not considered also the number of reshuffles can grow with the big instances. Park et al. (2011) **[7]** first select blocks to balance the GC's workload, and secondly, select the specific storage location based on a weighted function of space and GC utilization so departure dates of containers are not considered which will cause additional unproductive moves. Chen and Lu (2012) **[8]** aim to avoid rehandles in their approach to assign outbound containers, but workload balance of GC's are not considered. Ng et al. (2010) **[9]** used a new heuristic for the problem with import containers on ports with cyclical calling patterns, weight of containers are not considered.

2.2 Storage space assignment of groups of containers:

The main idea of the related research to this part is to propose methods to assign groups of containers to storage locations, where groups are, for example, based on the destination vessel, departure time, and type of containers. The heuristic's performance is assessed for variations in type of berth allocation, the transfer vehicle travel times, and loading/unloading times. An SA-based heuristic for this problem is proposed by Huang and Ren (2011) **[10]** that require enumerating all possible assignment permutations for the three container groups of import containers, result is not compared with existing storage policies. Jeong et al. (2012) **[11]** define a method to decide for each block how many import containers will be stored there, the consigned strategy take into account only the destination of containers the EDT is not considered. Nishimura et al. (2009) **[12]** propose a new MIP, and they use a new heuristic to minimize the weighted total container handling time no additional constraint for the destination type of the stored containers.

2.3 Storage space assignment in relation to other yard processes:

The aim of this part is to study the interaction between several marine operations in a containers terminal, each operation is a related to another adjacent operation so they can be optimized together. Here we will highlight papers that address more integrated decision making.

Murty (2007) **[13]** proposes a storage space assignment policy for a terminal using a consignment strategy. The policy is an adaptation of the best fit on-line heuristic for the bin packing problem, this is a limited MIP we will extend the proposed model by adding additional constraints and using the bin packing approach. Lee et al. (2006) **[14]** study storage space assignment to minimize the total number of GC shifts required to assign a group of containers to a set of blocks the workload can be reduced but the number of reshuffles can grow as we don't have a limitation of the EDT of the stored containers. Laik et al. (2008) **[15]** use the consignment strategy and the gantry crane dispatching problem their objective is to minimize the total handling and storage cost over a planning horizon, and to balance the workload between blocks the limitation of the proposed model the consignment strategy take into account only the inbound and outbound containers and the authors didn't define a container category to distinguish between export and import containers this may cause additional reshuffles in the storage area.

3. Problem definition : Container Stacking Problem (CSP)

Container terminals are divided into automated and non automated ports. The automated ports (AP) use the RMGC (Rail Mounted Gantry Crane), the AGV (Automated Guided Vehicles), the ALF (Automated lifting vehicles…Therefore non-automated ones (NAP), use the RTGC and the internal vehicles to transfer the containers from the berth to the storage space. So the AP needs a higher investment cost comparing with the NAP, but they have a lower operational cost, a faster crane speed and have more storage area. After the study of Lee et al. (2013) **[16]** 63.2% of the container terminals studied are non automated so they use the RTGC and the internal truck. So, this is the reason why we will focus on the NAP in this paper.

After the vessel's berth, quay cranes are assigned to each arrived docked vessel. Inbounded containers are unloaded by quay cranes (QCs) during the import operation. Then, containers are transported from berths to the storage area by trucks or other vehicles, the assignment of trucks to the storage area must verify the model's constraints and also starting by the containers which will have the smaller EDT and store them as close as possible to the transfer point. Once a container arrives at its stacking yard bay in the storage area, the stacking crane lifts the container from the truck and stacks it to the storage position. Afterwards GC can re-stack containers according to their EDT and destinations.

A container's position in the yard is then denoted by its block, bay, row and tier identifiers. A block is defined by its bays (length: basically [20, 30] per block) and rows (width: basically 6 per block, one is reserved to the internal truck) and the tiers (4levels per yard bay). Stacking yards are usually divided into multiple blocks, each consisting of a number of bays. A bay is composed of several stacks (row and tiers) of a certain size, and holds containers of the same size. This problem is called as the Container Stacking Problem (CSP), which is one of the most common and important problems at container terminals. Figure 3 show a schematic overview of container terminal, the theatrical container capacity of the yard is: 15 x 30 x 5 x 4 = 9000 TEUs (twenty-feet-equivalent units), which correspond to 9000 standard 20-feet containers.

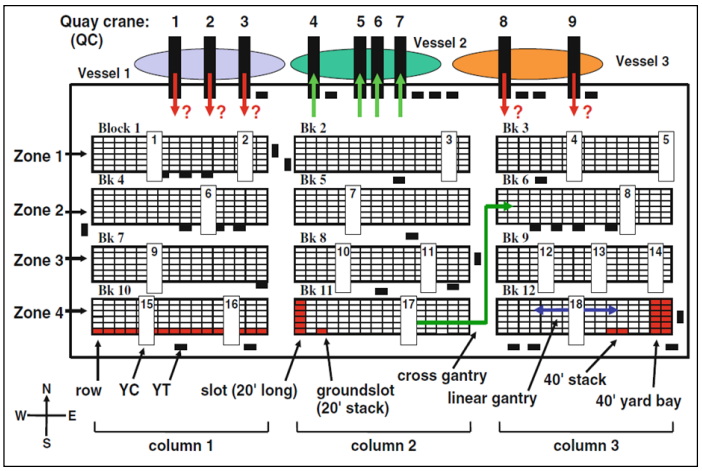


Fig.3. Schematic overview of the container terminal **[17]**.

3.1 Mathematical formulation

In a single stack, we can stores only containers of the same size, destination. The proposed stacking strategy is used to decide where to store newly arrived containers. The incoming containers belong to one of two types, namely 20-feet and 40-feet. A 20-feet container occupies 1 TEU in the storage area, while a 40-feet container occupies 2 TEUs. However, yard bay can only accommodate containers with the same size this is a physical restrictions to prevent possible damage. Therefore, the containers terminal assigns each position in the yard bay to a type of containers before the beginning of each planning horizon; this is done automatically, even for empty stacks.

Let R = (K, B) be a graph where K = {1, ..,N} is the containers set and B = {1, …, Nb} is the sub blocks set. The distance matrix cost is denoted by C_k^b. It represents the shortest way between the position of the block b and the ship of container k. The number of empty position of each block is denoted by B.

We propose a new mathematical model that reflects reality and takes into account most of the constraints imposed by port authorities. This model treats the following hypotheses:

1 We do not mix on the same block and in the same period the loading and the unloading containers.

2 Before the beginning of each period, we know the state of the storage area. For each block, we know: the number of container stored, the departure time of each container and the type of the block (dimension of stored containers).

3 For each stack, containers must be stored in the decreasing order of their departure time.

4 Containers are stored by respecting the constraint of dimension compatibility. All containers stored in the same block have the same dimension.
The CSP is formulated as a new and original mathematical programming model. This model is applied on each period in order to determine an optimal storage strategy.

3.2 Model Parameters

The proposed model is applied to each period of the working day in a containers terminal. The planning horizon is consisting of several shifts (3shift per day). In our study we will suppose that the range of shifts will be [1, 10].

Our objective is to minimize the global distance for the unloading containers from the berth to the storage space and then to balance the workload between the adjacent yard bays. The model parameters are defined as follow:
k: Container index.
b: Yard-bay index.
r: Index of row in the yard bay b.
e: Index of the level (tier) in the bay-row.
N: Total number of containers to be stored during the planning horizon.
B: Total number of yard-bays that may contain the containers during the planning horizon.
$\mathbf{w_k}$**: The weight of the container k.**
$\mathbf{C_b}$**: The free storage space in the yard-bay b (free storage positions).**
$\mathbf{r_b}$ **: Type of the yard-bay b (Three types: 20, 40).**
$\mathbf{R_k}$**: Type of container k.**
$\mathbf{D_k}$**: Destination of container k.**
$\mathbf{D_b}$**: Destination of yard-bay b.**
$\mathbf{t_k}$**: Estimated departure time of container k.**
C_k^b**: The assigned cost to choose / store a container k in the corresponding yard-bay b in the yard area.**

Decisions Variables:

$$\mathbf{X_{bk}^{re}} = \begin{cases} \mathbf{1} & \text{if k is assigned to the bay b, row r and tier e.} \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

$$Min \sum_{k=1}^{N}\sum_{b=1}^{B}\sum_{r=1}^{R}\sum_{e=1}^{E} C_k^b X_{bk}^{re}$$

$$\sum_{k=1}^{N}\sum_{r=1}^{R}\sum_{e=1}^{E} X_{bk}^{re} \le C_b \; ; b \in \{1,..,B\} \quad (1)$$

$$\sum_{b=1}^{B}\sum_{r=1}^{R}\sum_{e=1}^{E} X_{bk}^{re} = 1 \; ; k \in \{1,..,N\} \quad (2)$$

$$\sum_{k=1}^{N}\sum_{r=1}^{R}\sum_{e=1}^{E} (R_k - r_b) X_{bk}^{re} = 0 \; ; b \in \{1,..,B\} \quad (3)$$

$$\sum_{k=1}^{N}\sum_{r=1}^{R}\sum_{e=1}^{E} (D_k - D_b) X_{bk}^{re} = 0 \; ; b \in \{1,..,B\} \quad (4)$$

$$\sum_{k=1}^{N} X_{bk}^{re} = 1; \; b \in \{1,..,B\}, r \in \{1,..,R\}, e \in \{1,..,E\} \quad (5)$$

$$(w_k - w_{k'})(X_{bk}^{re\prime} - \sum_{e=e'+1}^{E} X_{bk'}^{re}) \le 0, b \in \{1,..,B\}, \forall r, \forall e, k \# k' \quad (6)$$

$$(t_k - t_{k'})(X_{bk}^{re\prime} - \sum_{e=e'+1}^{E} X_{bk'}^{re}) \le 0, b \in \{1,..,B\}, \forall r, \forall e, k \# k' \quad (7)$$

$$X_{bk}^{re} \in (0,1) \quad (8)$$

3.3 Model explanation :

In order to simplify our model, the related cost of the assignment of a specific container to the yard bay will be equal to distance between the berthing location and the final position in the yard bay. The reason behind this choice is to have a reference to which we can compare the obtained result. In the presented problem, only one vessel will be taken into account in each planning horizon, the planning horizon is defined before the ETA of vessel. It will be based on the vessel length and the number of containers to be stored.

The objective function aim to minimize the global distance for the total inbound containers to be stored during the planning horizon. This horizontal distance is calculated by adding all the distance of the stored containers between the berth location and the storage area (the assigned yard bay to each container). In the constraint (1) we define a total free positions per each yard bay, so the total number of the assigned containers must not exceed the numbers of free positions (this number is known at the beginning of the planning horizon).Constraint (2) ensures that each container is stored in a unique bay during the planning horizon. Constraint (3), (4) ensure that the size and the destination of the bay and the stored container are the same in each planning horizon, the aim of those constraints is to allow the proposed method to define a clustering of the available containers based on their destination and type, then we can allocate in the storage area a specific space to the those containers (consignment strategy). Each free position in the storage space can have only one container stored at the same time, this is what the constraint (5) present. The constraint of weight between the stored containers is defined in the next step (6) so the container on the top must have a weight less than the one stored in the previous level (this data is known for each inbound container). The estimated departure time is also known per each inbound container we will take it into account during the containers storage in the stacking area constraint (7).

4. Container stacking strategy :

Container stacking strategies are solution algorithms used to determine the storage position of each individual container, considering several operational constraints. In other words, stacking strategies are used to decide where to store newly arrived containers. We can find several studies in the literature which are interested in the container stacking problem study, Genetic algorithm (GA), Ant Colony (AC), Simulated annealing (SA) and hybrid between them are already applied to the CSP. No study was interested on the application of other heuristics such as: Bin Packing (BFD) to the CSP by defining a cluster with a set of containers taking into account the defined constraints this is what justifies the choice of this resolution strategy.

In this strategy, we consider several attributes of the containers while determining a stacking position. The first attribute is the expected departure time (EDT), which is defined as the time when a container will be removed from the stack. Secondly, the category of each container is considered the destination of each container and the weight.

The algorithm of the proposed strategy inspired from the bin packing (BFD) heuristic is defined as follow:

1. Get the relevant information of the container: Container type, weight, EDT and container destination.
2. Check each bay of matching container type in order, in ascending order and check whether selected bay is empty or not.
- **a**. If empty, place the arriving container to the first tier of the first stack in ascending order.
- **b**. If not empty, check each stack in order in ascending order of the selected bay for empty positions.
  - **i**. If stack empty, place the arriving container in this stack.
  - **ii**. If not empty, check whether category of the arriving container is the same as the category of the containers in the stack, or not.
    - **1**. If category is same, check if the Destination of the topmost container of the stack is equal to the destination of the incoming container.
      - **a**. If yes, check for the weight constraint
        - **i.** If satisfied.
          - **a**.If the ETD is less than the ETD of the topmost container of the stack, then stack the container.
          - **b**.If not return to the step 2.b
        - **ii**. If violated, return Step 2.b to check another stack of the selected bay.
      - **b**. If no, return Step 2.b.
    - **2**. If category is not same, return to Step 2.b

CLUSTRING

CONSIGNMENT

The aim of the proposed algorithm is to define an optimized plan to store the incoming containers by minimizing the global distance and reducing the number of reshuffles. As a summary, we will try to take into account to the below assumptions:

- Concentrate on the import operation to locate, store containers in the yard side.
- Use the available terminal equipments: QC, IT (trailer), YC and the straddle carrier to carry and handle the containers between the yard side and the gate.
- The productivity of the QC is known at the beginning of the unloading operation.
- The objective is to minimize the global distance between the yard side and the storage location.
- Reduce the number of reshuffles and avoid the housekeeping operation.
- The key factors of the terminal containers management: ETA (expected time of arrival), ETB (expected time of berthing), and ETD (expected time of departure).
- The assignment of the containers to the blocks will be performed using: the destination, the ETD, the size of the containers.
- A container can be stored in the vessel after the berthing [0,18h] before the unloading operation by the QCO. And during the transfer operation the container can stay in the IT [0, 1h]. And in the yard area between [1,5days].

5. Results:

Our problem interested in the deterministic state of the container stacking problem, the input is supposed to be known beforehand: State on the storage area and also the number of the containers to be stored in a specific planning horizon. The used instances verify the below technical constraints which are related to our port of study. Several instances have been tested using the bellow data:

- There are two types of containers (20 feet, 40 feet).
- We will consider a maximum number of 8QCs.
- Each stack can contain a maximum of four containers.
- Expected Departure date of containers to be stored is randomly chosen between [2h, 96h].
- Each container can wait [0,18h] for unloading from the vessel.
- The productivity of the quay crane is [30, 50] moves per hour.
- Planning Horizon: [0, 36h], Destination Type: 6
- The productivity of one QC which can perform 30moves per hour can be explained by: 5IT + 2.5 YC.
- The available IT in the port: [5, 25] depending on the type of the operation.
- During the transfer by IT, the move time between the seaside and the landside can be [0, 1h].

The model is coded with the Java programming language on Eclipse Standard Edition. The developed program is executed on a Core i5 2,7 GHz Win7 PC with 8-GB memory.

EDT Destination	**10**	**20**	**40**	**50**	**60**	**80**
1	50	40	40	25	24	-
2	20	35	11	-	64	25
3	45	75	56	35	60	45
4	50	50	60	64	-	0
5	50	-	50	24	30	30
6	30	30	30	-	-	15

Fig.4. Dispatching of arrived containers based on the destinations and the EDT.

In order to be able to compare with other results existing in the literature, we tried first to solve the model with an exact method using ILOG cplex, the proposed method (Branch & Cut) provide relevant result only for the small and medium instances. For the big ones, we tried to relax some constraints (10) & (13) to be able to compare with the result provided by Moussi et al (2015) **[18]** in his paper using the HAC/SA (Hybrid Ant Colony / Simulated Annealing).

In fact the initial model proposed by Moussi et al (2011) **[19]**, exclude the incoming containers with a residence time more than the one of all containers stored in the stacks. And also the author didn't take into account the destination and the type of the stored and adjacent sub blocks. So the proposed method has to be adjusted in order to be aligned with what we have in the literature.

The exact method (Branch & Cut) was tested and confirmed for the instances with small and medium parameters. In fact, this exact method can only used to verify the performance of our proposed method for this kind on instances.

The utilization is defined as the ratio of the total storage space occupied by all the unloading containers to the total storage space in the storage yard.

Utilization	Computation Time (s)	Results	Objective Value	Branch & Cut	HAC/SA
0.05	1	Optimal	15350	15350	15350
0.1	1	Optimal	10500	10500	10500
0.15	1	Optimal	17350	17350	17860
0.2	1	--	12004	19980	19980
0.25	1	Optimal	21345	21345	21558
0.3	274	--	21700	20065	20065
0.35	340	Optimal	31500	31500	31500
0.4	1	Optimal	31100	31100	32899
0.45	34,196.03	Out of memory	17564 (gap=12,18%)	33004	30345
0.5	3,833	Optimal	20399	20399	20399
0.55	110	Optimal	21344	21344	21344
0.6	13,304	Optimal	25345	--	25867
0.65	114	Optimal	27234	--	27445
0.7	6	Optimal	32897	--	32988
0.75	11,602	Optimal	39456	--	40156
0.8	6	Optimal	30231	--	--
0.85	32	Optimal	45676	--	47809
0.9	16	Optimal	50769	--	52344

The obtained result for the proposed method is relevant for the majority of the instances mentioned above. This means that the proposed method is more efficient than the one proposed in the study already mentioned above. For the future researches, this method can be combined with a local research method as an hybrid algorithm.

The new developed algorithm will be known as: Constraint satisfaction for minimizing stack numbers (CSMSN)[18]. The idea of CSMSN can be described as: Firstly, value selection rule to stack goods into an available stack location tentatively. Secondly, the domain of the remaining variables is changed through applying Best Fit Decreasing (BFD) and the look-ahead strategy of constraint propagation technology.

After performing the previous experimental evaluation and demonstrated that the effectiveness and feasibility of our approach compared with the best previous approach BFD. The decision maker was able to routinely solve the large & small scale discrete combinatorial optimization problems.

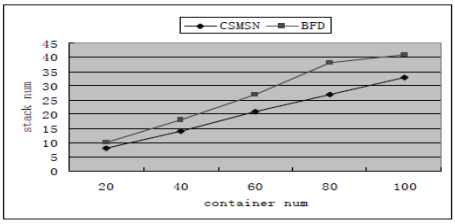


Fig.5. Comparison in stack numbers.

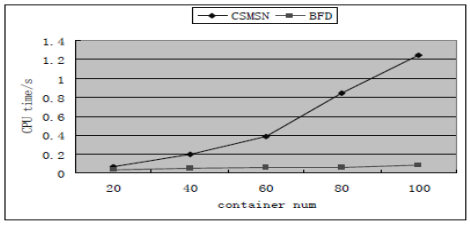


Fig.6. Comparison in CPU time.

6. Conclusion

In this paper we tried to introduce the containers terminal by introducing his structure and also the related researches to each component. Then we proposed a new mathematical formulation of the CSP with new constraints and objective function.

The proposed strategy defines a new method for unloading containers by using the bin packing heuristic as approach of resolution. The obtained result mentions a good performance for the small and medium instances. While it blocked for the big instances.

Future researches can add additional attributes such as the closest transfer point, which requires selecting an empty position closest to the transfer points from which the container will leave the stack, for instance import containers should be stacked as close to the transfer point landside as possible to decrease the traffic caused by external trucks. And also work with several vessels at the same time and also for export and transit in parallel to import operation. Finally the workload between adjacent bays in a storage block can be evaluated after the storage period.

1. REFERENCES

[1] Steenken, D., Voss, S., & Stahlbock, R. (2004). Container terminal operation and operations research-a classification and literature review. OR spectrum, 26(1), 3-49

[2] Stahlbock, R., & Voss, S. (2008). Operations research at container terminals: A literature update. OR Spectrum, 30(1),1-52.

[3] Rashidi, H., & Tsang, E. P. K. (2013). Novel constraints satisfaction models for optimization problems in container terminals. Applied Mathematical Modelling, 37(6), 3601-3634.

[4] Carlo et al. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. European Journal of Operational Research 235 (2014) 412–430.

[5] Guldogan, E. U. (2010). Simulation-based analysis for hierarchical storage assignment policies in a container terminal. Simulation, 87(6), 523–537.

[6] Park, T., Choe, R., Kim, Y. H., & Ryu, K. R. (2011). Dynamic adjustment of container stacking policy in an automated container terminal. International Journal of Production Economics, 133, 385–392.

[7] Chen, L., & Lu, Z. (2012). The storage location assignment problem for outbound containers in a maritime terminal. International Journal of Production Economics, 135(1), 73–80.

[8] Ng, W. C., Mak, K. L., & Li, M. K. (2010). Yard planning for vessel services with a cyclical calling pattern. Engineering Optimization, 42(11), 1039–1054.

[9] Huang and Ren (2011). Research on SA-based addressing model of slot in container terminal. Applied Mechanics and Materials, 97–98, 985–989.

[10] Jeong, Y. H., Kim, K. H., Woo, Y. J., & Seo, B. H. (2012). A simulation study on a workload-based operation planning. Industrial Engineering & Management Systems, 11(1), 103–113.

[11] Nishimura, E., Imai, A., Janssens, G. K., & Papadimitriou, S. (2009). Container storage and transshipment marine terminals. Transportation Research Part E, 45, 771–786

[12] Murty, K. G. (2007). Gantry crane pools and optimum layouts for storage yards of container terminals. Journal of Industrial and Systems Engineering, 1(3), 190–199.

[13] Lee, L. H., Chew, E. P., Tan, K. C., & Han, Y. (2006). An optimization model for storage yard management in transshipment hubs. OR Spectrum, 28, 539–561.

[14] Laik, N., & Hadjiconstantinou, E. (2008). Container assignment and gantry crane deployment in a container terminal: A case study. Maritime Economics & Logistics, 10, 90–107.

[15] Lee, B. K., & Kim, K. H. (2013). Optimizing the yard layout in container terminals. OR Spectrum, 35, 363–398.

[16] Ceyhun Güven, Deniz Türsel Eliiyi (2014) Trip allocation and stacking policies at a container terminal. Transportation Research Procedia 3 (2014) 565 – 573.

[17] Dong-liang Houa, Fang-rong Chen, Constraint Satisfaction Technology for Stacking Problem with Ordered Constraints3317 – 3321;

[18] Moussi et al. (2015) A hybrid ant colony and simulated annealing algorithm to solve the container stacking problem at seaport terminal. P.14-19.

[19] Moussi Ryadh, A. Yassine, T. Galinho (2011). Modern methods of combinatorial optimization for solving optimization problems in a containers terminal. Thesis published in 2012: P. 73-79.

Solving the earliest arrival flow problem using genetic algorithms

M. Hajjem[1], H. Bouziri[2], and E.G. Talbi[3]

[1] LARODEC, ISG, University of Tunis, Tunisia.
hajmanel@yahoo.fr
[2] LARODEC, ESSECT, University of Tunis, Tunisia.
hend.bouziri@gmail.com
[3] CRISTAL, Polytech'Lille, Université de Lille1, France.
Talbi@lifl.fr

The Earliest Arrival Flow Problem (EAFP) (Gale, 1959) is a variant of dynamic flow problem (Ford and Fulkerson, 1958). This problem tries not only to send as much flows as possible to destination node at every point in time, but also attempts to maximize the flow at the end of the horizon time. Many practical applications are modeled using earliest arrival flow and in many different fields, like in supply chains or in transportation (Ahuja et al., 1993). (Minieka, 1974) and (Hoppe and Tardos, 1994) have proposed polynomial algorithms for the EAFP with constant parameters. However, in this work, we consider the EAFP with flow-dependent transit time that has been presented by (Baumann, 2006).

Thus, given a dynamic network, we consider that we have a single source, single sink network with flow-dependent transit time and without storage in nodes. The objective of the earliest arrival flow problem is to send the maximal amount of flow from a source s to a sink d at each time unit of the horizon T. Mathematically this problem can be stated as follows:

$$Maximize \qquad x_{id}(t), \forall i \in N \setminus \{d\}, t \in [0, T[\tag{1}$$

subject to:

$$x_{ij}(t) \leq c_{ij}(t), \forall (i,j) \in A, t \leq T \tag{2}$$

$$x_i(t) = 0, \forall i \in N \setminus \{s,d\}, t \leq T \tag{3}$$

$$\sum_{(j,i) \in \varsigma_i^-} x_{ji}(t') = \sum_{(i,j) \in \varsigma_i^+} x_{ij}(t), \forall t = t' + \lambda_{ij}(t') \tag{4}$$

$$\lambda_{ij}(t) = \lambda_{ij}(0) \times (1 + \beta(x_{ij}(t)/c_{ij}(t))), \forall (i,j) \in A \tag{5}$$

The objective in equation (1) consists in maximizing the flow $x_{id}(t)$ starting from each node $i \in N \setminus \{d\}$ at time t and reaching destination node d at time $t + \lambda_{id}(t)$ for every $t \leq T - \lambda_{ij}(t)$. This objective should satisfy constraints of capacity in arcs (2) and in nodes (3). Flow over time is also subject to the conservation constraint which is formulated by (4) that does not allow the storage of flow in nodes.

Considering flow-dependent transit time which seems to be more realistic in the modeling of several real-life application, the transit time function in (5) is the *Bureau of Public Roads* function (BPR) defined by Sheffi (1985) and it was used by Baumann (2006). The BPR function depends on $\lambda_{ij}(0)$ which is the transit time of the arc (i,j) when it was empty. In this formula, $x_{ij}(t)$ is the flow departing the crossing of the arc (i,j) at time unit t and $c_{ij}(t)$ is the capacity of the arc (i,j) at time t. The parameter β reflects the level of correlation between the rate of flow and the transit time in each arc.

To solve this earliest arrival flow problem, we choose to use a genetic appraoch. This method begin by generating a population or a set of flow over time on a dynamic network. Then genetic operators are applied over generations to reach an efficient flow over time scheme on the given netwok.

In this paper, two chromosome representations are proposed and assessed to encode the flow over time property on a dynamic network. Moreover, initial population is constructed by the use of a greedy algorithm. Two crossover operator, that we call *periods-based crossover* and *Random-based crossover* are used. Further we propose two mutations methods, namely *random mutation* and *greedy mutation*.

Exprimental study based on the evaluation of several alternative scheme of the genetic algorithm for earliest arrival flow is conducted. Experiments are performed on the evacuation problem from building in emergency situation where we have to evacuate poeple as quickly as possible.

Bibliography

Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*.

Baumann, N. (2006). *Evacuation by Earliest Arrival Flows*. PhD thesis, University Dortmund.

Ford, L. and Fulkerson, D. (1958). Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419–433.

Gale, L. (1959). Transient flows in networks. *The Michigan Mathematical Journal*, 6:59–63.

Hoppe, B. and Tardos, E. (1994). Polynomial time algorithms for some evacuation problems. In *SODA'94 proceedings of the fifth annual ACM-SIAM symposium on discrete algorithms*, pages 433–441.

Minieka, E. (1974). Dynamic network flows with arc changes. *Networks*, 4:255–265.

Sheffi, Y. (1985). *Urban transportation networks: equilibrium analysis with mathematical programming methods*. Prentice-Hall.

Metaheuristics for smart grids and energy systems

Data Analytics Method Based on Feature Extraction for Voltage Sag Source Location

F. A. S. Borges, I. N. Silva
Department of Electrical and Computer Engineering
University of São Paulo – USP
São Carlos, Brazil
fabbioanderson@gmail.com

R. A. S. Fernandes
Department of Electrical Engineering
Federal University of São Carlos – UFSCar
São Carlos, Brazil
ricardo.asf@ufscar.br

***Abstract*—The correct location of the voltage sag sources corresponds to a very important task for the Power Quality area. However, this is not a trivial task due to the need of many monitoring devices. Therefore, with the data management from smart meters installed in primary distribution feeders, decision support tools that solve this problem become viable. Thus, this paper proposes an algorithm that determines the area where the voltage sag source is located. For this purpose, it was necessary to extract features from smart meters' voltage signals. In the sequence, we analyze the relevance of each feature in order to establish the most significant of them. In this way, the smart meter could extract this feature and sent it to the utility. At the utility side, the proposed algorithm will estimate the region where the voltage sag source is located. The location procedure is performed by cross checking the most relevant feature and the network topology.**

***Index Terms*—voltage sag, power quality, distribution feeders, disturbance location.**

I. INTRODUCTION

In accordance with standards and/or recommendations under the Power Quality (PQ) area, a voltage sag is characterized by the magnitude decrease of the nominal voltage, where the values are situated between 0.9 and 0.1 pu. Moreover, such disturbance corresponds to a great part of PQ events in primary distribution feeders. However, this kind of disturbance brings significant economic losses, mainly, for industrial consumers [1]. For this reason, it is extremely important for utilities that voltage sags are identified and located. This way, some actions should be taken in order to mitigate the disturbance.

In this sense, some methods have been published and presented good results to identify PQ disturbances [2]-[4]. On the other hand, methods designed to locate PQ disturbances are not common due to its complexity [5]-[7]. It is important to mention that methods related to the location of PQ disturbances are intended to indicate the position in the feeder where the disturbance source is connected. The feeder's position estimation also plays a key role in the decisions to be taken by the utility for preventive and/or corrective actions.

This work was supported by the FAPESP, CAPES and CNPq.

However, the process of location for voltage sags is not a trivial task, because this disturbance occurs in a short time and propagates throughout the feeder. Moreover, the deployment of smart meters, with capabilities of performing PQ analysis and provide data through communication channels, demands a high cost [5].

Considering the techniques found in the literature [6]-[13], it is observed that there is a difficulty in determining the location of the source of voltage sag. So, these methods typically use information acquired from just one monitoring point to determine if the source is located downstream or upstream of the meter. Therefore, for such methods have more accurate results, it is necessary to obtain information from other monitoring points. Thus, in a Smart Grid scenario that contains many smart meters installed, these methods would be viable [11].

In [6], the method estimates the voltage at each node using a multivariate regression model. The voltage sag source location is determined by means of the node that has the maximum voltage deviation and minimum standard deviation. This method was validated by using a system with 9 nodes (6 meters), and another system with 30 nodes (8 meters).

The Discrete Wavelet Transform was used in [12] to preprocessing signals generated after the switching of capacitor banks. The energy of each wavelet leaf was adopted as inputs to a hybrid system composed of Artificial Neural Network and Principal Component Analysis.

The method proposed in [13] is based on a current variation index calculated for each meter. According to the authors, the nodes (with meters) that present a voltage sag are near the source of the event. Following this premise, the proposed index was calculated before and after the PQ event. Thus, voltage sag source is found by comparing the current deviation rates.

It is noted that the above methods fail to achieve satisfactory coverage and development, as they require detailed system information but also rely on signal processing

calculations that must be provided by the monitoring system, which increases the implementation cost.

Thus, this paper presents a method that determines the location of voltage sags by analyzing the features of the disturbance that propagates in the network and is found in various parts of the system. Thus, the events detected near the origin present similarities, whereas there are dissimilarities in the disturbances detected in a position distant to the source. The features studied in this work could be easily provided and implemented by a monitoring system based on smart meters. Thus, instead of sending the waveform voltage to the control center, only information about the classified disturbance would be sent, reducing data traffic.

II. Simulation of Voltage Sags by Using ATP

In order to evaluate the proposed algorithm, it was necessary to model and simulate the IEEE 13-Bus Test Feeder [14] through the ATP (Alternative Transients Program) software. Having modeled this system (Figure 1), simulations were performed by changing the system's behavior to generate voltage sags with different durations and magnitudes. Thus, it was possible to obtain a database to validate the proposed algorithm.

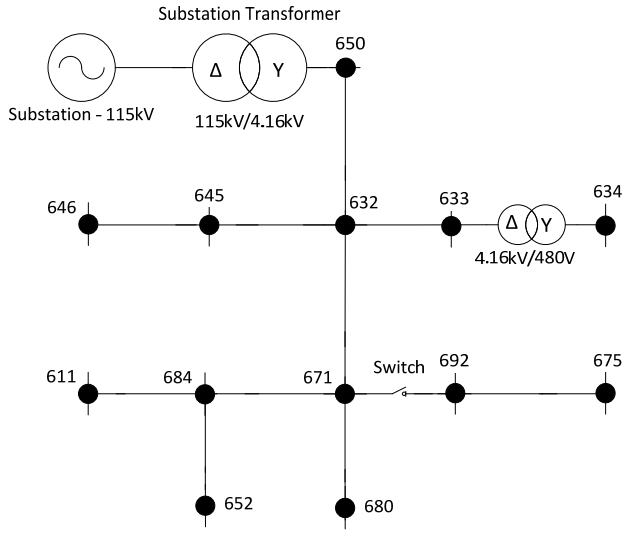


Figure 1. One-line diagram representing the IEEE 13-bus test feeder.

It is worth mentioning that all voltage sags were generated as a result of single-phase short-circuits, involving the phase C (with different fault impedances). The three-phase voltage was measured in the all nodes (with Phase C) and an example is illustrated in Figure 2.

Figure 2 shows an unbalanced steady-state condition at the beginning of the simulation. At time 0.04 s, the single-phase fault is applied to the system and, from this moment, it is noticed the occurrence of an oscillatory transient followed by a voltage sag with a magnitude of approximately 0.65 pu. Finally, as can be seen, the voltage sag occurs until the elimination of the short-circuit.

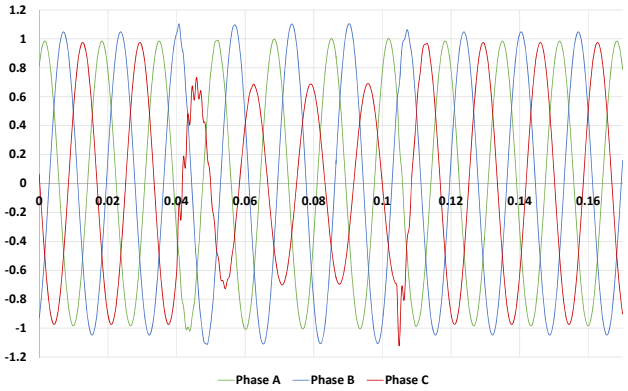


Figure 2. Three-phase voltage measured on the node 675 during a fault.

III. Data Analytics Based on Feature Extraction

A study about the event properties, as evidenced in each system branch, was conducted in order to better define heuristics about the system behavior in the presence of voltage sags. The knowledge obtained in this phase is used to implement the location method of the causes of sag.

A. *Feature Extraction Fundamentals*

A feature extractor should reduce the input pattern vector (i.e. the waveform of the detected disturbance) to a smaller dimension, preserving all useful information of the original vector. The extracted features should highlight the similarities between the disturbances detected around the location region, and present dissimilarities for the measured disturbances that are distant from the disturbance origin. In addition, the feature extraction will minimize the computational effort of the method.

The features analyzed were RMS, Peak Value, Shannon Entropy and Form Factor, respectively expressed by (1), (2), (3) and (4):

$$V_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} v_i^2}\ , \tag{1}$$

$$V_{peak} = V_{RMS} \cdot \sqrt{2}\ , \tag{2}$$

$$S = -\sum_{i=1}^{N} v_i^2 \cdot \log\left(v_i^2\right), \tag{3}$$

$$FF = \frac{\frac{1}{N}\sum_{i=1}^{N} v_i}{V_{RMS}}\ , \tag{4}$$

where v_i represents the i^{th} sample of instantaneous voltage and N is the quantity of samples for the segment.

B. *Disturbances Propagation Analysis*

After defining the metrics to be calculated, some fault scenarios were generated and the features were extracted from segmented data (voltage sag disturbance detected in each fault simulation). Three simulations are subsequently presented in order to illustrate how the disturbance propagation analysis

was performed according to the fault location. It is noteworthy that node 652 has been neglected due to the fact that it only has phase A and only single-phase faults in phase C were used in the tests.

Table I presents data of the features calculated in each node to a simulated fault applied at node 611, located at the end of the system. According to Table I, it is noted that the lowest values of RMS features, Peak Value and Form Factor were obtained at node 611 and node 684 that are close to the source. On the other hand, the Shannon Entropy feature does not show this similarity as the lowest value was found in node 650, which is located far from the point where the fault occurred.

TABLE I. FEATURES EXTRACTED FROM SMART METERS' SIGNALS FOR A FAULT APPLIED IN THE NODE 611 (PHASE C).

Nodes	RMS	Peak Value	Shannon Entropy	Form Factor
611	0.467	0.652	65.903	-0.0660
632	0.554	0.776	67.897	-0.0239
633	0.553	0.775	67.926	-0.0239
634	0.553	0.775	67.928	-0.0239
645	0.553	0.775	67.914	-0.0238
646	0.553	0.774	67.937	-0.0238
671	0.503	0.702	67.790	-0.0366
675	0.503	0.701	67.797	-0.0366
680	0.503	0.702	67.790	-0.0366
684	0.485	0.677	67.014	-0.0508
692	0.503	0.702	67.790	-0.0366
650	0.626	0.881	63.086	-0.0046

The next simulation analyzed corresponds to a fault situation in node 671. The data of the calculated features are presented in Table II. In this second scenario, it is noted that the calculated features presented very close values when compared among nodes upstream of the fault point. Moreover, downstream nodes have distinct values calculated at the fault point, thus emphasizing an area of similarity. The Form Factor feature was the one that best represented this similarity.

TABLE II. FEATURES EXTRACTED FROM SMART METERS' SIGNALS FOR A FAULT APPLIED IN THE NODE 671 (PHASE C).

Nodes	RMS	Peak Value	Shannon Entropy	Form Factor
611	0.4190	0.5766	67.71	-0.0728
632	0.4864	0.6774	68.38	-0.0451
633	0.4859	0.6767	68.40	-0.0450
634	0.4858	0.6766	68.40	-0.0450
645	0.4862	0.6772	68.39	-0.0449
646	0.4858	0.6766	68.41	-0.0448
671	0.4187	0.5776	67.79	-0.0728
675	0.4186	0.5776	67.79	-0.0726
680	0.4187	0.5776	67.79	-0.0728
684	0.4188	0.5771	67.75	-0.0728
692	0.4187	0.5776	67.79	-0.0728
650	0.5961	0.8364	63.70	-0.0080

Table III presents the feature values calculated for a fault applied at node 632. At this point, the disturbance has very close amplitude features, except for the voltage signal measured at the substation where the sag measured was less severe when compared to the others. Node 611 presented the lowest values of RMS and Peak Value. On the other hand, the lowest value of Form Factor was evidenced at nodes 671, 692 and 680, which are located closer to node 632, which originated the disturbance. For this scenario, it is concluded that the disturbance flows more strongly from the fault location to the ends, and the downstream nodes present a less severe disturbance.

TABLE III. FEATURES EXTRACTED FROM SMART METERS' SIGNALS FOR A FAULT APPLIED IN THE NODE 632 (PHASE C).

Nodes	RMS	Peak Value	Shannon Entropy	Form Factor
611	0.52025	0.74141	68.642	-0.045955
632	0.52840	0.75353	68.626	-0.045598
633	0.52781	0.75271	68.636	-0.045592
634	0.52775	0.75262	68.637	-0.045590
645	0.52814	0.75312	68.633	-0.045420
646	0.52762	0.75235	68.641	-0.045348
671	0.52123	0.74317	68.674	-0.046023
675	0.52077	0.74251	68.682	-0.045933
680	0.52123	0.74317	68.674	-0.046023
684	0.52073	0.74228	68.660	-0.045988
692	0.52123	0.74317	68.674	-0.046023
650	0.61403	0.86653	64.447	-0.008474

This analysis enables the understanding of the system behavior in the occurrence of a voltage sag disturbance originated in different parts of the system. The obtained heuristics are summarized below:

- The features RMS, Peak Value and Form Factor can define the degree of similarity between the disturbances, namely, nodes near the disturbance origin have similar feature values;
- The disturbance is "perceived" with greater intensity in the nodes located upstream of the location of the source that caused the event;
- When the cause of the disturbance is located in a node close to the substation, the disturbance is "perceived" with similar features in all system buses;
- When the cause of the disturbance is located in the system ends, the disturbance is "perceived" with greater intensity at the place of origin of the disturbance.

It is important to highlight that the most relevant feature is the form factor, because for a purely sinusoidal signal, it assumes a value of approximately zero. However, for waveforms containing oscillatory transients, the form factor presents abrupt changes in its value. Therefore, by using the form factor, it is possible to determine the oscillatory transient at the beginning of the signal and highlight the magnitude variation when the voltage sag is present.

IV. PROPOSED METHOD

Based on the data analytics previously presented, the proposed method uses the form factor as the information provided by smart meters installed at the feeder. Thus, it has been aimed at defining a region in which the voltage sag

source is located. Accordingly, the first step is to configure the topological structure of the feeder as a tree data structure. Thus, each node of the tree will represent a feeder node, as shown in Figure 3.

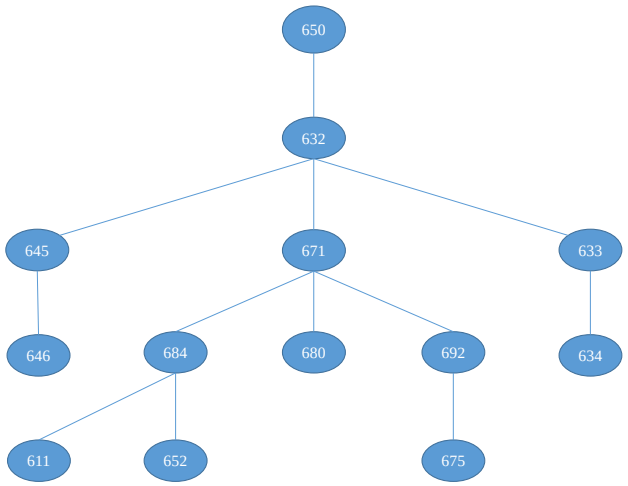


Figure 3. IEEE 13-bus test feeder represented as a tree data structure.

The second step defines the position of each smart meter, which is based on the power quality engineer's knowledge. In the sequence, the values of the form factor of each smart meter are received by the utility and normalized. Thus, it is possible to know how the disturbance sensitizes the meters. This normalization procedure is calculated based on (5):

$$Norm_i = \frac{FF_{\max} - FF_i}{FF_{\max} - FF_{\min}}, \quad (5)$$

where i represents the smart meter's ID. Highlighting that the form factors are calculated by the meters only when the voltage sag is detected, i.e., considering the segment.

Consequently, there is a need to calculate an adaptive threshold that corresponds to the summation of form factors of each smart meter divided by the number of meters (N_{meters}) installed on the feeder (6).

$$Threshold = \frac{\sum_{i=ID} FF_i}{N_{meters}}. \quad (6)$$

In this way, the smart meters that present form factors higher than the threshold are considered within the region where is located the voltage sag source. After obtained the list of smart meters, it is possible to eliminate the other meters and generate a newer tree. Thus, the root node will be defined by the smart meter with the highest form factor (equal to 1). Finally, the region where is located the voltage sag source is obtained by calculating the difference between the form factor of the root node and the form factors of its leaf nodes.

V. Results and Discussions

In order to conduct a well-defined study relating to the proposed method, it was simulated some cases of single-phase short-circuits, being one case for each node of the test feeder. In this sense, it is important to note that steps 1 and 2 of the algorithm are the same for all simulated cases. For this reason, the tree data structure and the location of smart meters have not changed. The meters are placed at the ends because of possible consumers, as it is usually where there is the greatest concentration of loads that would be affected by the disturbance. The substation meter, on the other hand, is commonly found (node 650). Finally, a last meter was allocated at node 671 due to this being a point of many system derivations with a radial feature. Such allocations may be seen in Figure 4.

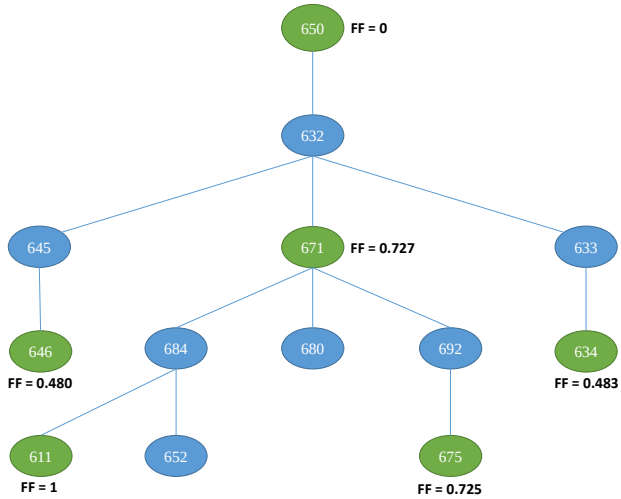


Figure 4. Tree data structure with smart meters allocated.

Table IV shows the results, where the first column shows the node where is applied the fault, i.e., the source of disturbance. The second and third columns of this table correspond to the calculated threshold and the region determined by the proposed algorithm, respectively. However, to explain how the algorithm works, three case studies are detailed.

TABLE IV. Localização da Fonte de Voltage Sags para Faltas Monofásicas Aplicadas em Distintos Nós do IEEE 13-Bus.

Faulty Node	Threshold	Location
611	0.416	611-684-671
632	0.827	671-632-645-646
633	0.582	634-633-632-645-646
634	0.547	634-633-632-645-646
645	0.452	645
646	0.357	646
671	0.689	671-692-675-684-611
675	0.547	671-692-675-684-611
680	0.452	671-680
684	0.541	611-684-671-692-675

The first case corresponds to a short-circuit at the node 684 and, for this case, the adaptive threshold calculated is 0.541. Following the previously determined steps, the meters that have values greater than the threshold were installed on the nodes 611, 675, and 671. Thus, the new tree data structure can be generated, where the node 611 is the root (highest form

factor) and the nodes 675 and 671 are the leaves, as shown in Figure 5.

From this new data structure, the path having the greatest difference, in terms of form factor, determines the location area. Therefore, it is noted that the method infers a voltage sag source between the node 611 and the node 675. However, among them are the nodes 684 (fault node), 671, and 692.

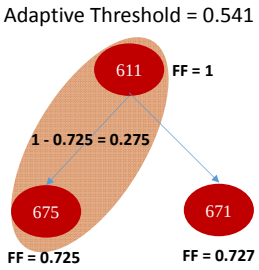


Figure 5. Tree data structure representing the results of case study 1 (fault at node 684).

Figures 6 and 7 present location areas for voltage sag sources when the short-circuit is applied at nodes 633 and 632, respectively.

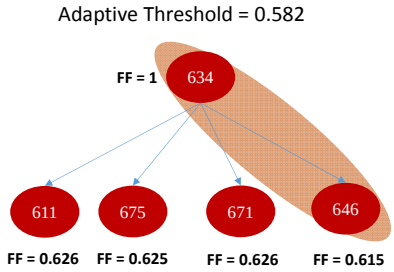


Figure 6. Tree data structure representing the results of case study 2 (fault at node 633).

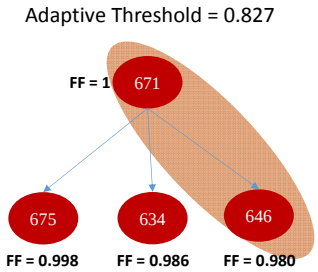


Figure 7. Tree data structure representing the results of case study 3 (fault at node 632).

It is observed in Table III and in Figures 5, 6, and 7 that the algorithm could correctly determine the location of the voltage sag sources. However, for some cases, this area involved a large number of nodes of the analyzed feeder. Nevertheless, the accuracy of the proposed algorithm is still good, whereas the distances between the nodes is small.

VI. Conclusions

This paper presented the development of an algorithm that represents an alternative to determine the location of voltage sag sources. For this purpose, smart meters connected to the primary feeder during short-circuit events calculate some features. Among these features, we verify the relevance of the form factor. In this way, it is clear that the proposed algorithm is quite simple and its application becomes feasible for Smart Grids. The results are quite satisfactory and demonstrated the need for a well-defined adaptive threshold. Thus, it would be important to idealize a method aimed at defining such a threshold for any feeder, i.e., considering its topology.

VII. Acknowledgements

The authors gratefully acknowledge the financial support for the development of this research provided by FAPESP (Process 2011/17610-0 and 2013/16778-0).

References

[1] R. C. Dugan, M. F. McGranaghan, and H. W. Beaty, Electrical Power Systems Quality, 3rd Edition. New York, 2002.

[2] Z. M. Chen, M. S. Li, T. Y. Ji, and Q. H. Wu, "Power Quality Disturbance Identification Using Morphological Pattern Spectrum and Probabilistic Neural Network," *IEEE General Meeting*, pp. 1-5, 2015.

[3] M. S. Manikandan, S. R. Samantaray, and I. Kamwa, "Detection and Classification of Power Quality Disturbances Using Sparse Signal Decomposition on Hybrid Dictionaries," IEEE Transactions on Instrumentation and Measurement, vol. 64, no. 1, pp. 27-38, 2015.

[4] O. P. Mahela, A. G. Shaik, and N. Gupta, "A Critical Review of Detection and Classification of Power Quality Events," *Renewable and Sustainable Energy Reviews*, vol. 41, pp. 495-505, 2015.

[5] G. W. Chang, J. P. Chao, H. M. Huang, C. I. Chen, and S. Y. Chu, "On Tracking the Source Location of Voltage Sags and Utility Shunt Capacitor Switching Transients," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 2124-2131, 2008.

[6] A. Kazemi, A. Mohamed, H. Shareef, and H. Raihi, "Accurate Voltage Sag-source Location Technique for Power Systems Using GACp and Multivariable Regression Methods," *International Journal of Electrical Power & Energy Systems*, vol. 56, pp. 97-109, 2014.

[7] N. Hamzah, A. Mohamed, and A. Hussain, "A New Approach to Locate the Voltage Sag Source Using Real Current Component," *Electric Power Systems Research*, vol. 72, no. 2, pp. 113-123, 2004.

[8] A. C. Parsons, W. M. Grady, E. J. Powers, and J. C. Soward, "A Direction Finder for Power Quality Disturbances Based upon Disturbance Power and Energy," *IEEE Transactions on Power Delivery*, vol. 15, no. 3, pp. 1081-1086, 2000.

[9] T. Tayjasanant, C. Li, and W. Xu, "A Resistance Sign-based Method for Voltage Sag Source Detection," *IEEE Transactions on Power Delivery*, vol. 20, no. 4, pp. 2544-2551, 2005.

[10] A. K. Pradhan and A. Routray, "Applying Distance Relay for Voltage Sag Source Detection," *IEEE Transactions on Power Delivery*, vol. 20, no. 1, pp. 529-531, 2005.

[11] M. Erol-Kantarci and H. T. Mouftah, "Energy-Efficient Information and Communication Infrastructures in the Smart Grid: A Survey on Interactions and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 179-197, 2015.

[12] Y. Y. Hong and B. Y. Chen, "Locating Switched Capacitor Using Wavelet Transform and Hybrid Principal Component Analysis Network," *IEEE Transactions on Power Delivery*, vol. 22, no. 2, pp. 1145-1152, 2007.

[13] G. W. Chang, J. P. Chao, S. Y. Chu, and C. Y. Chen, "A New Procedure for Tracking the Source Location," *IEEE PES General Meeting*, pp.1-4, 2007.

[14] IEEE Distribution Planning Working Group Report, "Radial Distribution Test Feeders," *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 975-985, 1991.

Memetic Algorithms and Local Search

Solving the 0-1 Multidimensional Knapsack Problem Using a Memetic Search Algorithm

Abdellah Rezoug · Dalila Boughaci · Mohamed Bader-El-Den

Abstract The 0-1 Multidimensional Knapsack Problem (MKP01) is one of the most well-known NP-hard problems in the combinatorial optimization field. This paper proposes an improved memetic search algorithm (MSA) for solving MKP01. The proposed MSA is a modified genetic algorithm (GA) combined with a stochastic local- simulated annealing algorithm (SLSA). First, we propose a new local search method SLSA that consists of a stochastic local search (SLS) and a simulated annealing (SA); Then, combine it with a genetic algorithm (GA) in the improved memetic search algorithm MSA. The proposed method MSA is implemented and evaluated on several well-known MKP01 benchmarks. MSA is compared to some approaches for MKP01 to measure its performance. The obtained results show that MSA algorithm is able to find high-quality solutions within a reasonable time.

Keywords Memetic algorithm · genetic algorithm · hybrid heuristic · combinatorial optimization · multidimensional knapsack problem.

A. Rezoug
Department of Computer Science
University Mhamed Bougara, Boumerdes, Algeria
E-mail: abdellah.rezoug@gmail.com

D. Boughaci
Department of Computer Science
University of Science and Technology Houari Boumediene, Algers, Algeria
E-mail: dalila_info@yahoo.fr

M. Bader-El-Den
School of Computing
University of Portsmouth, Portsmouth, United Kingdom
E-mail: mohamed.bader@port.ac.uk

1 Introduction

The 0-1 Multidimensional Knapsack Problem (MKP01) is a well-known optimization problem. The MKP01 is a strong NP-hard combinatorial optimization problem (Hartmanis 1982) which has been widely studied in the literature (Puchinger et al. 2010). The MKP01 has been discussed because of its theoretical importance and a wide range of applications. Many practical engineering design problems can be formulated as the MKP01, such as capital budgeting problem, project selection, cargo loading problems and in a variety of areas (Vasquez and Hao 2001).

Several methods have been proposed for solving the MKP01, where some are more effective than others. A non-exhaustive state of the art of the resolution methods is given in the following. The exact approaches are based on the implicit or explicit enumeration of all the feasible solutions in order to find the best one. The most common exact methods are branch and bound (Fukunaga 2011), linear programming, dynamic programming (Volgenant and Zoon 1990), integer programming, quadratic programming, etc. These methods obtain the optimal solution; however, the resolution time is relatively exponential with the size of data of the problem.

The approached methods are used when the size of data of the addressed problem is large or when the exact method is unable to find a solution in reasonable time. Local search, evolutionary algorithms and hybrid search methods are some examples of approaches methods. Among local search methods for MKP01, we mention the simulated annealing (Cho and Kim 1997), Tabu search (Vasquez and Vimont 2005) and Variable Neighbourhood Decomposition (Hanafi et al. 2009). Evolutionary methods such as Genetic Algorithm (Chu and Beasley 1998; Djannaty and Doostdar 2008; Khuri et al. 1994), Particle Swarm Optimization (Chih et al. 2014) have also been proposed for MKP01. Hybridization of several heuristics has been widely used for the MKP01 (Deane and Agarwa 2013; Yoon and Kim 2013).

In this paper, an improved memetic search algorithm MSA is proposed. MSA is based on the hybridization of a modified genetic algorithm (GA) and a stochastic local-simulated annealing (SLSA). SLSA is a local search algorithm that we propose to combine both the stochastic local search (SLS) (Boughaci et al. 2010) and the simulated annealing (SA) (Kirkpatrick and Vecchi 1983). The proposed algorithm MSA is compared with a pure GA, a hybrid GA with SA and a hybrid GA with SLS for solving MKP01 using the available datasets in the OR-Library [1].

The rest of the paper is organized as follows: a background of the used local search methods is presented in section 2. The proposed MSA approach for solving MKP01 is detailed in Section 3. Section 4 describes the experiments and gives some numerical results of the proposed algorithm on the considered datasets. Finally, Section 5 concludes and gives some future works.

[1] http://www.cs.nott.ac.uk/ jqd/mkp/

2 The 0-1 Multidimensional Knapsack Problem

The MKP01 is composed of N items and a knapsack with m different capacities b_i where $i \in \{1, \ldots, m\}$. Each item j where $j \in \{1, \ldots, n\}$ has a profit p_j and can take w_{ij} of the capacity i of the knapsack. The goal is to pack the items in the knapsack so as to maximize the profits of items without exceeding the capacities of the knapsack. The MKP01 can be represented as the following integer program:

$$\textbf{Maximize}: \quad \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\textbf{Subject to}: \sum_{j=1}^{n} w_{ij} x_j \leq b_i \qquad i \in \{1 \ldots m\} \tag{2}$$

$$x_j \in \{0, 1\} \qquad j \in \{1 \ldots n\} \tag{3}$$

3 The Memetic Search Algorithm (MSA) for the MKP01

MSA is a hybrid method composed of two algorithms: the Genetic Algorithm (GA) and the Stochastic Local Search-Simulated Annealing algorithm (SLSA). In MSA, the operators of the GA have been modified and adapted to the MKP01 and its efficiency has been improved by SLSA. In this section, we describe all the operators of GA and the SLSA algorithm.

3.1 The Stochastic Local-Simulated Annealing algorithm (SLSA)

The Stochastic Local-Simulated Annealing algorithm (SLSA) is a new approach based on two local search algorithms, the stochastic local search (SLS) (Boughaci et al. 2010) and the Simulated Annealing (SA) (Kirkpatrick and Vecchi 1983). SLSA performs the offspring X'_1, X'_2 a certain number of local steps that consists, first to create a neighbor solution by selecting an item to be added, second to remove all conflicting items.

With a probability $wp \in [0, 1]$, the item to be peeked is selected according to one of the two following criteria:

- An item I is chosen arbitrary. If I increases the objective function $f(X)$ then it will be packed in the knapsack, otherwise it will be accepted if the following expression is true $r_1 < \exp^{(-\Delta f/T)}$ where r is a random value, $\Delta f = f(X') - f(X)$ and T is a temperature value initially equal to T_0 relatively high.
- Choose the best item to be accepted.

The first step may cause a conflict. To eliminate all conflicts, the worst item is dropped from the knapsack. This process is repeated until eliminated all conflicts. After that, the temperature value is updated. In our case, the decreasing rule is found empirically.

3.2 The genetic algorithm step for MKP01

The MSA algorithm starts by launching the GA process. The GA steps and some modifications concerning selection, crossover, mutation, replacement operations are presented in following.

- MSA begins by the creation of the initial population P of individuals of a population size (PS). The Random Key method RK (Bean 1994) is used to create the individuals of P.
- The MSA selection operator selects items according to thier fitness values. The individuals with higher values will most likely be selected to reproduce, whereas, those with low values will be deserted. In MSA, the two individuals X_1, X_2 with the first and the second best fitness values are designated as parents for the crossing step.
- X_1, X_2, are used to perform a uniform crossover. The resulting offspring X'_1, X'_2 may be unfeasible solutions, so a repair process removes the worst item until they become feasible. Offspring X'_1 and X'_2 represent a feasible solutions, these lead to calculate their objective functions $F(X'_1)$ and $F(X'_2)$. A List Q is used to store the individuals that have already participated in a crossover. The purpose of this list is to prevent parents to be selected more than once during a number of iterations. The size of Q is defined according to the population size PS so that it turns the major part of its individuals $WT = F * PS$, with $F < 0.5$ is factor coefficient to calculate the size of Q.
- A mutation number of items (MNI) from offspring X'_1 and X'_2 are replaced by others items selected randomly from the best individual found so far X_{best}. This principle is inspired by the adjustment procedure of the improvised harmony in the Self-adaptive Global best Harmony Search (SGHS) (Pan et al. 2010). Substitute items must not be already included in the concerned offspring.
- the offspring X'_1, X'_2 after been corrected a local search improvement is performed using the SLSA (Algorithm of section 3.1).
- the offspring will be compared to the worst individuals in the population in terms of fitness, they replace them in the population if they are better.

The optimization process is repeated until the stopping criterion is checked. The criterion for stopping the optimization process is a limited number of iteration NI determined empirically according to the size of the studied problem.

4 Experiments

We coded MSA in C++ and compile it on a PC having 2 GHz Intel Core 2 Duo processor and 2 GB RAM. To evaluate the efficiency and performance of our MSA algorithm, it was initially tested on 54 standard test problems (divided into six different sets) which are available at the OR-Library [2] main-

[2] http://www.cs.nott.ac.uk/ jqd/mkp/

tained by Beasley. These problems are real-world problems widely used for the validation of the effectiveness of algorithms in the optimization community. These problems consisting of $m = 2$ to 30 and $n = 6$ to 105. After several experiments, we set the parameters for the MSA as in Table 1.

Table 1 The values of MSA parameters.

Parameter	Description	Value
NI	Number of iteration	50000
PS	Population size	200
WT	Wait time	70
N	Number of iteration of SLSA	100
wp	Probability of SLSA	0,93
T_0	Initial temperature	30
CT	Coefficient T update	0.0105
Nrun	Number RUN	30

4.1 Results on SAC-94 data

Table 2 shows the results of MSA application on the 54 instances. Here columns n and m represent the number of items, and the number of constraints (the number of dimensions) respectively, column *Optimum* is the value of Z optimal, column $Time$ is the average time of 30 runs. Column AVI is the AVerage number of Iteration of all the 30 runs. Column DFO is the Distance From the Optimum and finally NSR is the Number Successful Runs i.e. number of times MSA reaches the optimal solution.

From these results, we have identified various remarks. Firstly, MSA has succeeded to reach the optimal value once at least for all instances and 52 of the 54 in all 30 runs. The total average deviation of optimum is 0.0017%. Secondly, in 27 of 54 instances, the average execution time of MSA is less than one second, however, some instances required more time than the rest as WEISH 23-29, PET 5, WEING 7 and 8 and, in particular, the WEISH30, but in general, the time is relatively little with global average of 11,7 seconds. 7,4 seconds if the WEISH30 is ignored. Thirdly, the number of required average iteration varies from 1 to 600 thousand iterations (for PET1 and WEISH30 respectively) or an average of 35895 iterations and 23194 iterations without the WEISH30.

4.2 MSA vs. GA, GA-SA and GA-SLS

In this experiment the MSA approach was compared to three other approaches: the Genetic Algorithm (GA), its hybridization with the Simulated Annealing (GA-SA) (Rezoug et al. 2015) and by the Stochastic Local Search (GA-SLS)(Rezoug et al. 2015). In all tests, same parameters were used. We applied

Table 2 Results of MSA applied on the 54 OR-Library instances.

	n	m	$Optimum$	$Time$	AVI	DFO	NSR
HP	28	4	3418	5,020	24291	0	30/30
	35	4	3186	2,382	11237	0	30/30
PB	27	4	3090	1,133	5289	0	30/30
	34	4	3186	1,718	7717	0	30/30
	29	2	95168	0,335	1582	0	30/30
	20	10	2139	5,508	23963	0	30/30
	40	30	776	0,403	860	0	30/30
	37	30	1035	5,424	10746	0	30/30
PET	10	10	87061	0,013	1	0	30/30
	15	10	4015	0,010	2	0	30/30
	20	10	6120	0,013	13	0	30/30
	28	10	12400	0,905	3652	0	30/30
	39	5	10618	37,801	148	0,0922	9/30
	50	5	16537	13,059	47317	0,0026	29/30
SENTO	60	30	7772	0,773	1767	0	30/30
	60	30	8722	24,457	48509	0	30/30
WEING	28	2	141278	0,028	82	0	30/30
	28	2	130883	0,014	19	0	30/30
	28	2	95677	0,009	4	0	30/30
	28	2	119337	0,217	990	0	30/30
	28	2	98796	0,109	542	0	30/30
	28	2	130623	0,019	63	0	30/30
	105	2	1095445	58,112	121421	0	30/30
	105	2	624319	16,681	61761	0	30/30
WIESH	30	5	4554	0,010	10	0	30/30
	30	5	4536	0,057	216	0	30/30
	30	5	4115	0,063	248	0	30/30
	30	5	4561	0,013	28	0	30/30
	30	5	4514	0,013	35	0	30/30
	40	5	5557	0,647	2798	0	30/30
	40	5	5567	0,659	2863	0	30/30
	40	5	5605	0,893	3795	0	30/30
	40	5	5246	0,105	437	0	30/30
	50	5	6339	0,619	2491	0	30/30
	50	5	5643	0,774	3251	0	30/30
	50	5	6339	0,706	2810	0	30/30
	50	5	6159	0,782	3222	0	30/30
	60	5	6954	3,623	13939	0	30/30
	60	5	7486	0,613	2326	0	30/30
	60	5	7289	2,767	10593	0	30/30
	60	5	8633	9,375	31597	0	30/30
	70	5	9580	15,670	53919	0	30/30
	70	5	7698	4,035	15589	0	30/30
	70	5	9450	3,713	12805	0	30/30
	70	5	9074	3,445	11963	0	30/30
	80	5	8947	7,761	26499	0	30/30
	80	5	8344	32,683	112813	0	30/30
	80	5	10220	60,773	192505	0	30/30
	80	5	9939	29,397	96748	0	30/30
	90	5	9584	14,307	47672	0	30/30
	90	5	9819	20,250	67586	0	30/30
	90	5	9492	23,620	78641	0	30/30
	90	5	9410	18,233	61025	0	30/30
	90	5	11191	232,352	683614	0	30/30

the GA, GA-SA, GA-SLS and MSA algorithms on the WEISH27 instance (optimum = 9819). The value of fitness was noted every second for 15 seconds. With the average of fitness obtained in 10 runs, the curves 1 were drawn.

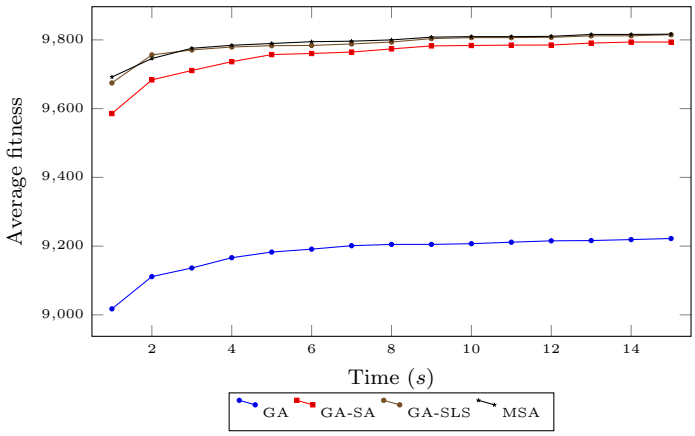


Fig. 1 Comparing MSA to GA, GA-SA, GA-SLS through the WEISH27

The curves of the Fig. 1 represent the evolution of the four algorithms for 15 seconds. Algorithms start using population generated according to RK. The algorithms MSA and GA-SLS curves are too close together except at the end where only MSA reaches the optimum.

These two algorithms exceeded slightly that of the GA-SA algorithm and largely that of the GA algorithm. We can deduce that MSA is faster and more efficient than the other three algorithms.

4.3 MSA vs. CRGA and SRGA.

Table 3 shows the comparison of MSA with CRGA and SRGA algorithms (Yang et al. 2013) in terms of effectiveness. Where CRGA and SRGA are two algorithms based on GA. The comparison was done according to the calculated means fitness. The values of results presented in Table 3 are the same published by the authors.

We can observe that MSA was able to reach the optimum in almost all instances, whereas CRGA and SRGA were not able to reach the optimum in any instance. MSA is more effective than CRGA and SRGA.

4.4 MSA vs. DPHEU.

Table 4 shows the comparison of MSA with DPHEU algorithm (Veni and Balachandar 2010) in terms of effectiveness. The comparison was done according

Table 3 Mean fitness obtained by MSA compared to CRGA and SRGA.

		CRGA	SRGA	MSA
instance	optimum	Mean	Mean	Mean
PET1	3800	3782	**3800**	**3800**
PET2	8706	8562	8662	**8706**
PET3	4015	3878	3941	**4015**
PET4	6120	5476	5630	**6120**
PET5	12400	11203	12240	**12400**
PET6	10618	10107	9953	**10608,2**
PET7	16537	15184	14915	**16536,5**
HP1	3418	3259	3214	**3418**
HP2	3186	2921	2864	**3186**
Weing1	141278	130885	131409	**141278**
Weing2	130883	113289	116883	**130883**
Weing4	119337	107535	106950	**119337**
Weing5	98796	79038	75109	**98796**
Weing6	130623	116773	115671	**130623**
Weing7	1095445	975269	783196	**1095445**
PB1	3090	2953	2936	**3090**
PB2	3186	2965	2907	**3186**
PB4	95168	83483	81412	**95168**
PB5	2139	1984	2016	**2139**
Weish1	4554	3774	3777	**4554**

to the calculated average percentage of deviation (A.P.O.D) and the number of problems for which the optimal solution was reached (N.O.P.T). The values of results presented in Table 4 are the same published by the authors.

Table 4 Average fitness and number of optimal solutions obtained by MSA compared to DPHEU.

	number	DPHEU		MSA	
data	of instances	A.P.O.D	N.O.P.T	A.P.O.D	N.O.P.T
HP	2	0.0	2	0.0	2
PB	6	0.04	5	**0.0**	**6**
PET	6	**0.0**	**6**	0,0158	4
SENTO	2	0.0	2	0.0	2
WEING	8	0.0	8	0.0	8
WEISH	30	0.03	28	**0.0**	**30**

Both MSA and DPHEU obtained similar results in three of the six sets (HP, SENTO and WEING). MSA is more effective than DPHEU in two sets PB and WEISH. DPHEU is more effective than MSA in the PET set. We can say that MSA and DPHEU are quite similar in terms of effectiveness.

5 Conclusion

This paper aims to propose an improved memetic algorithm (MSA) to the 0-1 multidimensional knapsack problem (MKP01). The proposed method is

the hybridization of a modified genetic algorithm (GA) and the Stochastic Local Search-Simulated Annealing (SLSA). In MSA, new technics for selection, crossover and mutation operators of the GA were proposed. Then the GA is combined with SLSA that fusions the Stochastic Local Search (SLS) and the simulated annealing (SA) algorithms. In order to show the effectiveness of MSA, several tests were carried out on a large range of benchmarks known by their complexity. Also, the algorithm was compared to other algorithms. In conclusion, the use of the best individual in mutation and the application of SLSA gave the MSA a powerful capacity of intensification. Similarly, the WT mechanism in the selection and the random crossover gave it a good ability of diversification. The obtained results of MSA are very competitive compared to other recent approaches. MSA is able to reach to the optimality almost all the instances.

References

Bean JC (1994) Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing, 6*(2), 154-160.

Boughaci D, Benhamou B, Drias H (2010) Local search methods for the optimal winner determination problem in combinatorial auctions. *Journal of Mathematical Modelling and Algorithms, 9*(2), 165-180.

Chih M, Lin CJ, Chern MS, Ou TY (2014) Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem. *Applied Mathematical Modelling, 38*(4), 1338-1350.

Cho JH, Kim YD (1997) A simulated annealing algorithm for resource constrained project scheduling problems. *Journal of the Operational Research Society*, 736-744.

Chu PC, Beasley JE (1998) A genetic algorithm for the multidimensional knapsack problem. *Journal of heuristics, 4*(1), 63-86.

Deane J, Agarwal A (2013) Neural, genetic, and neurogenetic approaches for solving the 0-1 multidimensional knapsack problem. *International Journal of Management & Information Systems (Online), 17*(1), 43.

Djannaty F, Doostdar S (2008) A hybrid genetic algorithm for the multidimensional knapsack problem. *International Journal of Contemporary Mathematical Sciences, 3*(9), 443-456.

Fukunaga AS (2011) A branch-and-bound algorithm for hard multiple knapsack problems. *Annals of Operations Research, 184*(1), 97-119.

Hanafi S, Lazic J, Mladenovic N, Wilbaut C (2009, June) Variable Neighbourhood Decomposition Search with Bounding for Multidimensional Knapsack Problem. In *Information Control Problems in Manufacturing* (Vol. 13, No. 1, pp. 2018-2022).

Hartmanis J (1982) Computers and Intractability: A Guide to the Theory of NP-Completeness (Michael R. Garey and David S. Johnson). *Siam Review, 24*(1), 90.

Khuri S, Bck T, Heitktter J (1994, April) The zero/one multiple knapsack problem and genetic algorithms. In *Proceedings of the 1994 ACM symposium on Applied computing* (pp. 188-193). ACM.

Kirkpatrick S, Vecchi MP (1983) Optimization by simmulated annealing. *science, 220*(4598), 671-680.

Pan QK, Suganthan PN, Tasgetiren MF, Liang JJ (2010) A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation, 216*(3), 830-848.

Puchinger J, Raidl GR, Pferschy U (2010) The multidimensional knapsack problem: Structure and algorithms. *INFORMS Journal on Computing, 22*(2), 250-265.

Rezoug A, Boughaci D, Badr-El-Den M (2015) Memetic Algorithm for Solving the 0-1 Multidimensional Knapsack Problem. In *Progress in Artificial Intelligence* (pp. 298-304). Springer International Publishing.

Vasquez M, Hao JK (2001) A logic-constrained knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. *Computational Optimization and Applications, 20*(2), 137-157.

Vasquez M, Vimont Y (2005) Improved results on the 01 multidimensional knapsack problem. *European Journal of Operational Research, 165*(1), 70-81.

Veni KK, Balachandar SR (2010) A new heuristic approach for large size zeroone multi knapsack problem using intercept matrix. *Int. J. Comput. Math. Sci, 4*(5), 259-263.

Volgenant A, Zoon JA (1990) An improved heuristic for multidimensional 0-1 knapsack problems. *Journal of the Operational Research Society*, 963-970.

Yang HH, Wang MT, Yang CH (2013) A hybrid of rough sets and genetic algorithms for solving the 0-1 multidimensional Knapsack problem. *Innovative Computing Information and Control, 9*(9), 3537-3548.

Yoon Y, Kim YH (2013) A memetic Lagrangian heuristic for the 0-1 multidimensional knapsack problem. *Discrete Dynamics in Nature and Society, 2013*.

Zou D, Gao L, Li S, Wu J (2011) Solving 01 knapsack problem by a novel global harmony search algorithm. *Applied Soft Computing, 11*(2), 1556-1564.

Subgradient method driven Lagrangian heuristic for Large Scale Generalized Assignment Problems

S. Haddadi, M. Cheraitia, and F. Guessoum

LabSTIC, Université 8 Mai 1945, BP 401, 24000 Guelma, Algeria
salim.haddadi@yahoo.com
meryem.cheraitia@hotmail.fr
fatima_guessoum@yahoo.fr

Abstract. This paper presents a heuristic approach for solving the minimization version of the generalized assignment problem, in which each task needs to be assigned to exactly one agent such that the total cost is minimized. The main ingredients of the algorithm are: (1) A general purpose solver is used to find an initial feasible solution to start with. (2) A standard subgradient method is used to solve the Lagrangian dual resulting from dualizing the knapsack constraints. (3) A local-improvement heuristic, based on solving a restricted generalized assignment problem where each task is only allowed to be assigned to one of two different agents, is used in each iteration of the subgradient method. (4) Another local-improvement heuristic, based on shifting or interchanging the assignments of tasks, is applied to every feasible assignment. Moreover, in a second version of our heuristic, we also use a heuristic variable fixing method to reduce the size of the problem before running the original heuristic on the reduced problem. Computational results are presented on both versions of our heuristic. The comparison with other known heuristics for solving the generalized assignment problem concludes that our heuristics are faster than the known competition and almost always produce solutions which are equally good or better. **Keywords** Combinatorial optimization; Heuristics; generalized assignment; Lagrangian heuristic.

1 Introduction

We have n tasks that should be assigned to m agents. Let b_i be the resource availability of agent i, let a_{ij} be the amount of resource required by agent i to perform task j and let c_{ij} be the cost so assigned. Each task must be assigned to one agent without exceeding his resource availability. The generalized assignment problem (GAP) is to find a minimum cost assignment. The mathematical model is

$$\begin{aligned} \min & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \\ & \sum_{j\in J} a_{ij}x_{ij} \leq b_i, \ i \in I \\ & \sum_{i\in I} x_{ij} = 1, \ j \in J \\ & x_{ij} \in \{0,1\}, \ i \in I, j \in J \end{aligned} \tag{1}$$

where x_{ij} is a binary variable that indicates whether task j is assigned to agent i, and where $I = \{i_1, i_2, ..., i_m\}$ and $J = \{j_1, j_2, ..., j_n\}$. We assume that the coefficients c_{ij}, a_{ij} and b_i are all non-negative integers.

The GAP is known to be strongly NP-hard, and even the problem of whether there exists a feasible assignment is NP-complete. Some theoretical results about the approximation guarantee (*i.e.* how can a heuristic solution be far from optimal in the worst case) may be found in [5,19,24].

The GAP has great practical significance [2,3,8,18,20]. For this reason, there have been many attempts to design exact algorithms [11,17,22,23]. However, because they need large computing times and memory requirements, the latter cannot address the large size instances arising in practice. Another alternative is to develop fast algorithms capable of quickly finding good assignments. The literature is rich of such methods. There are Lagrangian heuristics, such as [9,10,15], which

demonstrate average effectiveness, and a lot of meta-heuristics [1,4,6,7,14,21,25,26,27] which turn out to be more efficient.

Our goal in this paper is to show that a simple Lagrangian heuristic is still capable of competing with sophisticated meta-heuristics A general scheme of a Lagrangian heuristic for a combinatorial optimization problem P may be summarized as follows:

1. Choose a Lagrangian relaxation of problem P.
2. Use a subgradient method to solve the Lagrangian dual.
3. At each iteration of the subgradient method, try to derive a feasible solution from the relaxed solution.

The third is the crucial step. Usually, it is difficult to generate good feasible solutions for the GAP at this stage. This is why Lagrangian heuristics are abandoned in favor of the meta-heuristics. Our heuristic does not use the relaxed solution is step 3. Instead, the key contribution of this paper lies in introducing, in step 3, a new and powerful improvement heuristic based on solving a restricted generalized assignment problem (called FEASIBLE-2GAP) where each task is only allowed to be assigned to one of two different agents. Given the best current feasible assignment of cost c^*, problem FEASIBLE-2GAP is always feasible, and provides a feasible assignment of cost no less than c^*. Problem FEASIBLE-2GAP is proven to be NP-hard, and transformed to an equivalent monotone 0-1 IP with m constraints and n variables. Because of the special structure and sparsity of the latter (two nonzero entries per column), it is relatively easy to solve [13]. Usually special structure and sparsity contribute to make a problem easier [12]. The whole algorithm consists of iteratively defining and solving problem FEASIBLE-2GAP, in each iteration of the subgradient method. The definition of the successive FEASIBLE-2GAP problems is driven by the subgradient method. For this reason, our method can be considered as a Lagrangian heuristic. Next, in order to speed up the algorithm, we introduce a novel heuristic approach for reducing the size of the GAP by fixing the value of some variables. This idea turns out to be useful and promising as we shall see. The comparison of our heuristic with existing methods shows that it is the fastest, and one of the best from the accuracy point of view.

The paper is organized as follows. In Section 2, we introduce problem FEASIBLE-2GAP and prove its NP-hardness. The proposed heuristic is described in Section 3. In Section 4, we show how we can heuristically reduce the size of the GAP by fixing for ever the value of some of the variables. Section 5 is devoted to the computational experiments, and Section 6 ends the paper.

2 A special GAP

TWe consider the special problem of assigning n tasks to m agents where each task is only allowed to be assigned to one of two different agents. Let s_j and t_j be the two agents who can perform task j, with $s_j, t_j \in I$ and $s_j \neq t_j$. The problem's data consists of $2n$ costs $c_{s_jj}, c_{t_jj}, j \in J$, $2n$ coefficients $a_{s_jj}, a_{t_jj}, j \in J$ and m coefficients $b_i, i \in I$. Let $J_i^{(s)} = \{j \in J | s_j = i\}$ and $J_i^{(t)} = \{j \in J | t_j = i\}$. $J_i^{(s)} \cup J_i^{(t)}$ is the set of all the tasks that can be performed by agent $i, i \in I$. Since there are $2n$ variables, let them be $y_j, z_j, j \in J$. The mathematical model of this problem, called 2GAP, is

$$\min \sum_{j \in J} c_{s_jj} y_j + \sum_{j \in J} c_{t_jj} z_j \tag{2}$$

$$\sum_{j \in J_i^{(s)}} a_{s_jj} y_j + \sum_{j \in J_i^{(t)}} a_{t_jj} z_j \leq b_i, \ i \in I \tag{3}$$

$$y_j + z_j = 1, \ j \in J \tag{4}$$

$$y_j, z_j \in \{0, 1\}, \ j \in J \tag{5}$$

Problem 2GAP is NP-hard since even the decision problem, called 2GAP-FEASIBILITY, which asks whether there exist $y_j, z_j, j \in J$ satisfying (3-5) is NP-complete.

Theorem 1. *Problem 2GAP-FEASIBILITY is NP-complete.*

Proof. Consider the restriction of 2GAP-FEASIBILITY with $m = 2$ agents. Without loss of generality, let $s_j = 1$ and $t_j = 2$ for $j \in J$. Furthermore, assume that $a_{1j} = a_{2j} = a_j, j \in J, \sum_{j \in J} a_j = 2b$ and $b_1 = b_2 = b$. The restricted problem becomes that of seeking $y_j, z_j, j \in J$ satisfying

$$\begin{aligned} \sum_{j \in J} a_j y_j &\leq b \\ \sum_{j \in J} a_j z_j &\leq b \\ y_j + z_j &= 1, \quad j \in J \\ y_j, z_j &\in \{0,1\}, \quad j \in J \end{aligned}$$

which is nothing other than Partition, a well-known NP-complete problem.

In fact, the problem we are interested in, called FEASIBLE-2GAP, seems a little bit easier than 2GAP, since we know a feasible assignment (i.e. $y_j, z_j, j \in J$ satisfying (3-5)). More precisely, we know that if we set $y_j = 1, z_j = 0, j \in J$, we get a feasible assignment whose cost is $c^* = \sum_{j \in J} c_{s_j j} y_j$. Therefore the cost of an optimal solution of problem FEASIBLE-2GAP is no less than c^*. We show that problem FEASIBLE-2GAP is NP-hard since the problem FEASIBLE-2GAP-Decision which asks whether there exist a feasible assignment of cost greater than c^* is NP-complete.

Theorem 2. *Problem FEASIBLE-2GAP-Decision is NP-complete.*

Proof. There are trivial lower and upper bounds on the cost of an optimal solution of problem 2GAP which are

$$\begin{aligned} lb &= \sum_{j \in J} \min \left\{ c_{s_j j}, c_{t_j j} \right\} \\ ub &= \sum_{j \in J} \max \left\{ c_{s_j j}, c_{t_j j} \right\} \end{aligned}$$

Suppose there is a polynomial-time algorithm A for FEASIBLE-2GAP-Decision. Using binary search and then at most $\log(ub - lb)$ times algorithm A, we solve problem 2GAP. Since $\log(ub - lb)$ is polynomially bounded by the size of problem 2GAP, we have a polynomial-time algorithm for the latter. This contradicts theorem 1.

3 Lagrangian heuristic

First of all, we need a feasible assignment to begin with. We found that the greedy heuristic of [16], as well as the heuristic based on rounding the solution of the LP-relaxation of the GAP, are very weak in the sense that they provide feasible assignments that are far from lower bound. For this reason, we charge Cplex 10.1 to find an initial feasible solution. Once a first feasible assignment is found, Cplex is stopped, and ub is set to the value of this feasible solution. The three other basic components of the Lagrangian heuristic are: 1) a standard subgradient phase, 2) an improvement heuristic which is executed at each iteration of the subgradient method, and 3) a shift-interchange local-improvement heuristic.

3.1 Subgradient phase

A non-positive Lagrangian multiplier vector $\pi \in \mathbb{R}^m$ is associated to the relaxed knapsack constraints (1). The Lagrangian dual of the GAP is then the problem $\max_{\pi \in \mathbb{R}^m_-} L(\pi)$ where $L(\pi)$ is the optimal value of the Lagrangian relaxation of the GAP

$$LR(\pi) \begin{cases} \sum_{i \in I} \pi_i b_i + \min \sum_{i \in I} \sum_{j \in J} (c_{ij} - \pi_i a_{ij}) x_{ij} & \\ \sum_{i \in I} x_{ij} = 1 & j \in J \\ x_{ij} \in \{0,1\} & i \in I, j \in J \end{cases}$$

It is well known that $L(\pi)$ is a lower bound on the optimal value of the GAP, for all $\pi \in \mathbb{R}^m_-$. Consequently, the optimal solution of the Lagrangian dual provides the best bound. But finding the optimal vector π may be a challenging task. Fortunately, there exists a popular and fast method for finding a near optimal solution π^* of the Lagrangian dual. Usually beginning with $\pi^{(0)} = \mathbf{0}$, the subgradient method generates a sequence $\pi^{(k)}, k \geq 1$ and takes $\pi^* = \text{argmax}_{k \geq 0} L(\pi^{(k)})$. The Lagrangian vectors $\pi^{(k)}$ are iteratively updated

$$\pi_i^{(k+1)} = \min\{0, \pi_i^{(k)} + \rho \frac{ub - L\left(\pi^{(k)}\right)}{\left\|\sigma\left(\pi^{(k)}\right)\right\|^2} \sigma_i\left(\pi^{(k)}\right)\} \tag{6}$$

where ρ is the step-size parameter satisfying $0 < \rho \leq 2$, and ub is an overestimate of the optimal value of the GAP (usually ub is set to the cost of the best assignment at hand). Given $\pi^{(k)}$, $L\left(\pi^{(k)}\right)$ is the optimal value of problem $LR\left(\pi^{(k)}\right)$. It is easily computed: For every $j \in J$, compute $i_j = \text{argmin}_{i \in I}\left\{c_{ij} - \pi_i^{(k)} a_{ij}\right\}$ and set $x_{i_j j} = 1$. The value $L\left(\pi^{(k)}\right)$ is then computed accordingly. $\sigma\left(\pi^{(k)}\right)$ is the subgradient vector at $\pi^{(k)}$ whose components are $\sigma_i\left(\pi^{(k)}\right) = b_i - \sum_{j \in J} a_{ij} x_{ij}, i \in I$. The value $\rho\left(ub - L\left(\pi^{(k)}\right)\right) / \left\|\sigma\left(\pi^{(k)}\right)\right\|^2$ is the step-size in the subgradient direction.

It is well known that $L(\pi^*)$ is equal to the LP-relaxation value of the GAP. Fortunately, since our approach is purely heuristic, we need no lower bound, and the Lagrangian relaxation and the subgradient method are intended solely to drive the Lagrangian heuristic (by providing the reduced costs for the definition of the successive FEASIBLE-2GAP problems).

3.2 Improvement phase

Given an arbitrary iteration k, let $s_j, j \in J$ be the current best feasible assignment, with $s_j = i \iff x_{ij} = 1$. We compute the reduced costs

$$c_{ij} - \pi_i^{(k)} a_{ij} \tag{7}$$

and the coefficients

$$t_j = \text{argmin}_{\{i \in I, i \neq s_j\}}\{c_{ij} - \pi_i^{(k)} a_{ij}\}, j \in J \tag{8}$$

and define problem FEASIBLE-2GAP as in (2-5). By construction, problem FEASIBLE-2GAP is feasible. Therefore it must have a feasible solution whose cost is no less than the cost of the best feasible assignment at hand. In fact, by solving problem FEASIBLE-2GAP we expect an improvement. Indeed, from the results obtained during the computational experiments, this approach turns out to be powerful. It provides several good assignments of decreasing costs, as we shall see later.

The neighborhood of the current best feasible assignment has size 2^n since every task can be assigned to one of two agents. Clearly, it contains as special cases the shifting and the interchange improvements which will be presented in the next subsection. It can be seen that it also contains as special case the ejection chain of [26] and [27]. Solving exactly problem FEASIBLE-2GAP provides the best neighbor.

Before solving problem FEASIBLE-2GAP, we reduce it to a monotone 0-1 IP with n variables and m constraints, and two nonzero coefficients per column with opposite sign (see [13]). By letting $z_j = 1 - y_j, j \in J$, problem FEASIBLE-2GAP becomes

$$\sum_{j \in J} c_{t_j j} + \min \sum_{j \in J} \left(c_{s_j j} - c_{t_j j}\right) y_j$$

$$\sum_{j \in J_i^{(s)}} a_{s_j j} y_j - \sum_{j \in J_i^{(t)}} a_{t_j j} y_j \leq b_i - \sum_{j \in J_i^{(t)}} a_{t_j j}, \ i \in I$$

$$y_j \in \{0, 1\}, \ j \in J$$

Let us call this problem MONOTONE-01IP. Note that since the LP-relaxation of MONOTONE-01IP is a monotone LP, it can be solved with generalized network flow methods. Although problem MONOTONE-01IP is NP-hard, it is computationally relatively easy to solve because of its special structure and the sparsity of its constraint matrix (see [12,13]).

3.3 Shift-interchange heuristic (SIH)

Shift and interchange are well known simple local improvement procedures. They are detailed in [9]. Let $\{s_j, j \in J\}$ be the current feasible assignment and ub its cost. Shifting a task j means withdrawing it from agent s_j and assigning it to some agent $i \neq s_j$. A neighbor of the best current assignment is any assignment obtained by shifting a task to another agent. Its size is thus $O(mn)$. Let r_i be the remaining resource availability of agent $i, i \in I$. The task j is shifted to agent i if and only if it results in a cost improvement and maintains feasibility, i.e. $r_i \geq a_{ij},\ c_{ij} < c_{s_j j}$. After shifting task j to agent i, we proceed to the updates: $s_j \longleftarrow i,\ ub \longleftarrow ub + c_{ij} - c_{s_j j},\ r_{s_j} \longleftarrow r_{s_j} + a_{s_j j},\ r_i \longleftarrow r_i - a_{ij}$. The complexity of the shifting procedure is $O(mn)$ since the major computational effort is spent in exploring the neighborhood.

The interchange procedure consists of interchanging two agent/task selections. The size of the neighborhood is $O\left(n^2\right)$. Suppose that we want to interchange two tasks $j1$ and $j2$ previously assigned respectively to agents $i1$ and $i2$. This is the same as simultaneously shifting task $j1$ to agent $i2$ and task $j2$ to agent $i1$. The complexity of the interchange procedure is $O\left(n^2\right)$.

The shift-interchange procedure may be summarized as follows: a) Scan the shift (resp. interchange) neighborhood for the best admissible shifting (resp. interchange). b) Perform the best of the two. c) Repeat a) and b) until no improvement is possible.

The initial feasible assignment found by Cplex, and every assignment found at any iteration of the subgradient method, are submitted to the shift-interchange procedure.

3.4 The whole heuristic

We merely give a pseudo-code description of the heuristic called ILH (for Improved Lagrangian Heuristic) in Figure 1. Steps 02 and 3 of the algorithm cost $O(n \max\{m, n\})$ each. The complexity of step 8 is $O(1)$. Steps 03, 6 and 7 cost $O(m)$ each, and the complexity of step 1 (resp. 4 and 5) is $O(mn)$ (resp. $O(n)$ and $O(mn)$). Hence the complexity of algorithm ILH without steps 01 and 2 is $O(n \max\{m, n\})$. Therefore, if there is a heavy computational effort, it must happen in step 01 and/or step 2. Step 01 is executed once and does not cost very much, while step 2 is applied $NBIT$ times, where $NBIT$ is the number of iterations of the subgradient method. The implementation details will be given in the section devoted to the computational experiments.

4 Heuristic variable fixing procedure

Conventional variable fixing procedures exist in the literature (see for example [16,23]). These are mathematically correct logical rules, generally derived from the reduced costs. In contrast, the approach in this section is purely heuristic. We exploit a simple and useful idea for coping with the huge number of variables by discarding some of them. This idea results in speeding up the heuristic, still finding very good assignments. We first apply a basic subgradient method (algorithm ILH without steps 02, 2, 3 and 4, and where step 1 only computes the reduced costs. Note that the initial ub never changes). At the end of the subgradient method, we are interested in finding which of the variables are very often selected by the subgradient method, and which are never, or only occasionally, selected. For this purpose, we compute the "frequency" $f_{ij} = (1/NBIT) \sum_{k=0}^{NBIT} x_{ij}$ where $x_{ij}, i \in I, j \in J$, is the optimal solution of problem $LR\left(\pi^{(k)}\right)$. Note that, for task $j, j \in J$

$$\sum_{i \in I} f_{ij} = \frac{1}{NBIT} \sum_{k=0}^{NBIT} \sum_{i \in I} x_{ij} = \frac{1}{NBIT} \sum_{k=0}^{NBIT} 1 = 1$$

Intuitively, the more the value f_{ij} is close to 1 the greater is our wish to assign task j to agent i for ever. This observation brings us to assign every task j to agent i whenever the value f_{ij} is above a given threshold, thus completely discarding task j from the problem. Before going any further, let us consider an example taken from OR-library. The name of the instance is C515-1. The data

are (after conversion into minimization form)

$$c = \begin{pmatrix} 9 & 5 & 4 & 8 & \boxed{2} & 11 & \boxed{6} & 8 & 7 & 8 & 10 & 4 & 2 & \boxed{2} & 10 \\ 3 & 10 & 5 & 10 & 9 & 10 & 7 & \boxed{1} & 8 & 5 & 9 & 11 & 1 & 9 & 2 \\ 10 & 6 & 10 & \boxed{1} & 2 & 10 & 9 & 7 & 7 & 8 & 6 & 10 & 9 & 5 & \boxed{2} \\ 7 & 7 & 4 & 4 & 6 & 10 & 7 & 9 & 5 & \boxed{7} & \boxed{1} & \boxed{3} & 1 & 1 & 1 \\ 8 & 7 & 11 & 11 & 5 & 1 & 10 & 10 & 3 & 11 & 4 & 9 & 7 & 4 & 2 \end{pmatrix}$$

$$a = \begin{pmatrix} 8 & 15 & 14 & 23 & \boxed{8} & 16 & \boxed{8} & 25 & 9 & 17 & 25 & 15 & 10 & \boxed{8} & 24 \\ 15 & 7 & 23 & 22 & 11 & 11 & 12 & \boxed{10} & 17 & 16 & 7 & 16 & 10 & 18 & 22 \\ 21 & 20 & 6 & \boxed{22} & 24 & 10 & 24 & 9 & 21 & 14 & 11 & 14 & 11 & 19 & \boxed{16} \\ 20 & 11 & 8 & 14 & 9 & 5 & 6 & 19 & 19 & \boxed{7} & \boxed{6} & \boxed{6} & 13 & 9 & 18 \\ 8 & 13 & 13 & 13 & 10 & 20 & 25 & 16 & 16 & 17 & 10 & 10 & 5 & 12 & 23 \end{pmatrix}$$

$$b = (36, 34, 38, 27, 33)$$

When the subgradient method finishes, we compute

$$f = \begin{pmatrix} 0.03 & 0.43 & 0.06 & 0 & \boxed{0.97} & 0^* & \boxed{0.94} & 0^* & 0.49 & 0.01 & 0 & 0.01 & 0.08 & \boxed{0.95} & 0 \\ 0.58 & 0.48 & 0^* & 0 & 0 & 0.01^* & 0.01 & \boxed{0.98} & 0^* & 0.03 & 0 & 0 & 0.91 & 0 & 0.01 \\ 0^* & 0.01^* & 0.01^* & \boxed{0.96} & 0 & 0.02^* & 0 & 0 & 0.01 & 0^* & 0.02 & 0 & 0 & 0^* & 0 & \boxed{0.94} \\ 0^* & 0^* & 0.91 & 0.01 & 0 & 0.06 & 0.04 & 0 & 0^* & \boxed{0.92} & \boxed{0.97} & \boxed{0.98} & 0^* & 0.02 & 0 \\ 0.38 & 0.08 & 0.01^* & 0.03 & 0.03 & 0.91 & 0.01 & 0.01 & 0.50 & 0.01 & 0.03 & 0.01 & 0.01^* & 0.03 & 0.04 \end{pmatrix}$$

Choosing a threshold of 0.92, we see in the matrix f that the tasks $4, 5, 7, 8, 10, 11, 12, 14$ and 15 can be assigned to the appropriate agents (framed numbers). The cost of assigning the removed tasks (called "fixed cost"in the sequel) is $2+6+2+1+1+2+7+1+3 = 25$ and the data of the remaining smaller GAP are

$$c' = \begin{pmatrix} 9 & 5 & 4 & 11 & \boxed{7} & 2 \\ 3 & \boxed{10} & 5 & 10 & 8 & \boxed{1} \\ 10 & 6 & 10 & 10 & 7 & 9 \\ 7 & 7 & \boxed{4} & 10 & 5 & 1 \\ \boxed{8} & 7 & 11 & \boxed{1} & 3 & 7 \end{pmatrix}$$

$$a' = \begin{pmatrix} 8 & 15 & 14 & 16 & 9 & 10 \\ 15 & 7 & 23 & 11 & 17 & 10 \\ 21 & 20 & 6 & 10 & 21 & 11 \\ 20 & 11 & 8 & 5 & 19 & 13 \\ 8 & 13 & 13 & 20 & 16 & 5 \end{pmatrix}$$

$$b' = (12, 24, 0, 8, 33)$$

An optimal solution (framed numbers) of cost 31 is found, and the whole assignment is optimal for the original GAP with cost 56.

The new heuristic, called ILHRED, consists of applying twice the subgradient method. The first time, just for computing the values f_{ij} and removing every task such that f_{ij} is above a given threshold. The data of the reduced GAP and the initial cost are stored in a file, and the first subgradient method finishes. The second subgradient method is nothing other than algorithm ILH which reads the data of the smaller GAP from the file filled during the application of the first subgradient method.

Although we did not in our implementation, we can further reduce the size of the GAP by removing for ever every variable whose corresponding value f_{ij} is below a given threshold. For example, we can remove every variable (asterisked items in the matrix f) such that $f_{ij} \leq 0.02$. This leaves a smaller problem with the same optimal solution.

$$c'' = \begin{pmatrix} 9 & 5 & 4 & \boxed{7} & 2 \\ 3 & \boxed{10} & & & \boxed{1} \\ & & & & \\ & & \boxed{4} & 10 & \\ \boxed{8} & 7 & & \boxed{1} & 3 \end{pmatrix}$$

$$a'' = \begin{pmatrix} 8 & 15 & 14 & & \boxed{9} & 10 \\ 15 & \boxed{7} & & & & \boxed{10} \\ & & & & & \\ & & & \boxed{8} & 5 & \\ \boxed{8} & 13 & & \boxed{20} & 16 & \end{pmatrix}$$

$$b'' = (12, 24, 0, 8, 33)$$

We point out that this is a purely heuristic approach. If the goal is to solve the GAP, it should be avoided, since an optimal solution is not guaranteed. Nevertheless, the idea will undoubtedly attract the attention of the practitioners who want to quickly approximate the GAP.

5 Computational experiments

Our heuristics are coded in C and run on Intel Pentium Core Dual 2 GHz with 2 Gb memory. They are evaluated on twenty seven large instances from the web site http://www.al.cm.is.nagoya-u.ac.jp/~yagiura/gap. See Table 1 for their names and sizes. The reader is referred to [26] to see how these instances were generated. The smaller instances (from this site and from OR-library) are considered as satisfactorily solved by the BaB algorithm of [11].

5.1 The algorithm ILH

We begin by giving some details about our implementation. The number $NBIT$ of iterations of the subgradient method is fixed to 200 for the instances of type C, and 400 for the instances of type D and E. Beginning with the value 2, the coefficient ρ is multiplied by 0.98 at every iteration. We also found that it is better to use $1.2 \times ub$ in the update formulas (6) of the Lagrangian vector. These choices ensure a slow convergence. Note that we are not interested in the value of the lower bound. All computing times are reported in seconds.

Cplex is applied in step 01 just for finding a feasible assignment. It stops as soon as a first feasible assignment is found. Finding a first feasible assignment with Cplex can take an important amount of time, particularly for the largest instances with $n = 1600$ tasks as we shall see later. In step 3 of algorithm ILH, problem MONOTONE-01IP is submitted to Cplex (for the reader familiar with Cplex, the MIP emphasis is set on "hidden feasible solutions") during a fixed timelimit. This means that the solution obtained within this fixed timelimit is, generally, either suboptimal, or optimal but without any proof of optimality. From our experience, we have noticed that the hardest instances are those with a large number m of agents. From this observation, we fixed a timelimit to Cplex depending only on m. Because of this fixed timelimit, the whole execution time can be anticipated. For example, for an instance with $m = 80$, when the timelimit is $m/40 = 2$ seconds, the algorithm takes about $400 \times 2 = 800$ seconds. We imposed four different time-limits $m/80, m/40, m/20$ and $m/10$. The computational results of algorithm ILH are shown in Table 2 where Column 1 refers to the name of the instance, Column 2 gives the lower bound reported in [27] and Column 3 shows the cost of the feasible assignment provided by Cplex. The remaining columns are self-explanatory. The boldfaced values are the costs of the best assignments found. We observe that the larger the timelimit allowed to Cplex, the better the results, but also the larger the computing times.

Surprisingly, we faced a divergence problem of the subgradient method when running the instances D201600, D401600 and D801600. We found that the only way to enforce the convergence was to decrease the value of ρ by beginning with $\rho = 0.5$ or less instead of 2.

5.2 The heuristic ILHRED

The code of the heuristic ILHRED is cut into two pieces. The first, called preprocessing, performs a simple subgradient method, computes the values f_{ij}, removes tasks from the problem as previously explained, stores the data of the reduced GAP in a file and stops. In our implementation, the tolerance threshold is fixed to 0.92 for the instances of type C, and 0.78 for the instances of type D and E. The second part of algorithm ILHRED is algorithm ILH applied to the smaller GAP obtained during the preprocessing phase. The results of this heuristic on the benchmark instances are shown in Table 3. Columns 3 to 5 give respectively the number of remaining tasks, the cost of assigning the removed tasks, and the preprocessing time. Most of the preprocessing time is spent by Cplex in finding a feasible assignment. We can see that the preprocessing time can be important for the larger instances (for example, more than 38 seconds for the instance E801600). The sixth column refers to the cost of the assignment found by Cplex, the seventh column gives the cost of the best assignment found by ILHRED, the eighth column shows the computing time of the heuristic ILHRED, and the last column gives the total computing time. The timelimit for Cplex is set to $m/80$. The table shows that this heuristic finds very quickly very good assignments.

5.3 Comparison of algorithm ILH with Cplex

It would be interesting to compare algorithm ILH with Cplex by letting them run for the same amount of time. The comparison is presented in Table 4. It can be seen that both ILH($m/80$) and ILH($m/10$) are more efficient than Cplex, for all types of instances. Particularly, ILH($m/80$) is fast and twice more accurate in average than Cplex on the instances of type C and D, and about ten times more effective on the instances of type E.

5.4 Is algorithm ILH sensitive to the initial solution ?

[16] proposed the following greedy heuristic (MTH) for the GAP. Let $s(i, j)$ be a score function relative to the assignment of task j to agent i. The algorithm iteratively considers all the unassigned tasks and determines that task $j^\star$, which has the maximum difference between the largest and the second largest $f(i, j)$. The task $j^\star$ is then assigned to the agent for which $f(i, j^\star)$ is minimum. The process is repeated, after adjusting the remaining resource availability of the agent performing task $j^\star$, until all the tasks are assigned. Clearly, MTH may fail, and indeed it fails on all of the instances of type E. Also, as we previously noted, the solutions provided are far from LB by about 50% in average for the instances of type D (see the column labeled "MTH cost"in Table 5).

Usually, the performance of the local-improvement heuristics does depend heavily on the initial solution. It is not the case of algorithm ILH. Indeed, algorithm ILH($m/80$) is so efficient that it provides a good solution, already in the first iteration, that does not deviate from LB by more than 7% for the instances of type D (see the column labeled "Cost at iteration 1"in the same table). The comparison in Table 5 shows that algorithm ILH is unaffected by the initial feasible assignment since the average deviation from LB and the computation time are almost the same in the two cases, whether we begin with Cplex or with MTH.

5.5 How does the performance of algorithm ILH depend on the SIH ?

To answer this question, we run algorithm ILH($m/80$) with and without the SIH (see Table 6). We observe that the effect of the SIH is almost insignificant, since the average deviation from LB and the computing time are almost the same in the two cases, except on the three instances D201600, D401600 and D801600. In our opinion, this is due to the divergence problem mentioned earlier in the manuscript. Indeed, beginning the subgradient method with $\rho = 0.5$, instead of 2, results in a faster convergence and to a smaller lower bound.

5.6 Comparison of ILH and ILHRED with existing methods

There are numerous heuristic methods for the GAP, as we have remarked in the introduction, but only three of them are truly effective. These are the tabu search with ejection chain (TSEC) of [26], the path relinking approach with ejection chain (PREC) of [27], and the more recent tabu

search/Branch-and-Bound approach of [25] (TSBB(S) or TSBB(L) according to whether small or large timelimits are allowed). We first compare them with ILH and ILHRED in Table 7 from the accuracy point of view. The best values obtained by the heuristics are boldfaced and the next to best are labeled with superscript 2. We observe that ILH($m/10$) provides either the better assignment, or the next to best, while ILHRED($m/80$) is slightly worse. Furthermore, the average deviation from LB is computed for every type of instances. From this point of view, the two best heuristics are PREC and ILH($m/10$). They compete since the last dominates on C and E instances, while PREC dominates on D instances. The two methods are respectively followed by TSEC, ILHRED($m/80$), TSBB(L) and TSBB(S).

The comparison from the point of view of computing time is more problematic since the systems used are totally different. The heuristics TSEC and PREC are executed on a Sun Ultra II with a frequency of 300 MGz, and TSBB on a Viglen CX130 server (Xeon 3.0 GHz). The computing times need to be scaled before we can compare them. The scaling factor used, although not likely to be accurate, is based on the reference [25] which indicates that the Viglen CX130 is roughly ten times faster than the Sun Ultra II. Clearly, with 3 GHz of frequency, the Viglen is faster than our PC. So, our machine is at most ten times faster than the SUN ULTRA II. Therefore, in Table 8 giving the comparison, our computing times, and those of [25], are multiplied by 10 to make them expressed in equivalent Sun Ultra II times. The average values indicate that ILHRED($m/80$) is about 12 times faster than ILH($m/10$), which is about 2 to 3 times faster than PREC, TSEC and TSBB(S), which, in turn, are about ten times faster than TSBB(L).

If the successive 0-1 integer programs MONOTONE-01IP were solved exactly, our heuristic approaches would be perfectly deterministic. But since they are only approximated within a fixed timelimit, and since Cplex does not perform precisely the same amount of work in this fixed timelimit, our results are not always reproducible. Indeed, on the same instance and with exactly the same parameters, algorithm ILH may provide different solutions.

6 Conclusion

We proposed two key ideas for dealing with the large sized instances of the GAP. The first idea is a ILH based on solving, at each iteration of the subgradient method, a restricted GAP where each task is only allowed to be assigned to one of two different agents. This special GAP is transformed into a monotone feasible 0-1 IP with two nonzero entries per column. Algorithm ILH turns out to be powerful and the resulting heuristic is faster than the known competition and almost always produces solutions which are equally good or better. The second is a useful idea for coping with the large instances arising in practice. Using the information provided by the subgradient method, it consists of heuristically reducing the size of the problem before running the original heuristic on the reduced problem. A benefit of this size reduction is that it is potentially applicable to every combinatorial optimization problem. At the state of our knowledge, this is the first time the two ideas are proposed. They would be undoubtedly interesting for practitioners, and for improving the quality and speed of the existing metaheuristics.

References

1. Alfandari, L., Plateau, A., & Tolla P. (2004). A path relinking algorithm for the generalized assignment problem. In: Metaheuristics: Computer Decision-Making, Volume 86 of Applied Optimization, Springer US, pp.1-17. Doi:10.1007/978-1-4757-4137-7_1.
2. Barbas, J., & Marín, A. (2004). Maximal covering code multiplexing access telecommunication networks. European journal of operational research, 159, 219–238. Doi:10.1016/S0377-2217(03)00381-3.
3. Campbell, G M., & Diaby, M. (2002). Development and evaluation of an assignment heuristic for allocating cross-trained workers. European Journal of Operational Research, 138, 9–20. Doi:10.1016/S0377-2217(01)00107-2.
4. Chu, P. C., & Beasley J. E. (1997). A genetic algorithm for the generalized assignment problem. Computers and Operations Research, 24, 17–23. Doi:10.1016/S0305-0548(96)00032-9.
5. Cohen, R., Katzir, L., & Raz, D. (2006). An efficient approximation for the generalized assignment problem. Information Processing Letters, 100, 162–166. Doi:10.1016/j.ipl.2006.06.003.
6. Díaz, J. A., & Fernández, E. (2001). A tabu search heuristic for the generalized assignment problem. European Journal of Operational Research, 132, 22-38. Doi:10.1016/S0377-2217(00)00108-9.

7. Feltl, H., & Raidl, G. R. (2004). An improved hybrid genetic algorithm for the generalized assignment problem. ACM Symposium on Applied Computing, Nicosia, Cyprus, pp. 990–995. Doi:10.1145/967900.968102.
8. Foulds, L. R., & Wilson, J. M. (1997). A variation of the generalized assignment problem arising in the New Zealand dairy industry. Annals of Operations Research, 69, 105–114. Doi:10.1023/A:1018968625626.
9. Haddadi, S. (1999). Lagrangian decomposition based heuristic for the generalized assignment problem. INFOR: Information Systems and Operational Research, 37, 392–402.
10. Haddadi, S., & Ouzia, H. (2001). Effective Lagrangian heuristic for the generalized assignment problem. INFOR: Information Systems and Operational Research, 39, 351–356.
11. Haddadi, S., & Ouzia, H. (2004). Effective algorithm and heuristic for the generalized assignment problem. European Journal of Operational Research, 153, 184–190. Doi:10.1016/S0377-2217(02)00710-5.
12. Hochbaum, D.S., & Naor, J. (1994). Simple and fast algorithm linear and integer programs with two variables per inequality. SIAM Journal on Computing, 23, 1179-1192. Doi: 10.1137/S0097539793251876.
13. Hauchbaum, D. S. (2004). Monotonizing Linear Programs with Up to Two Nonzeroes Per Column. Operations Research Letters, 32, 49–58. Doi:10.1016/S0167-6377(03)00074-9.
14. Higgins, A. J. (2001). A dynamic tabu search for large-scale generalized assignment problems. Computers and Operations Research, 28, 1039–1048. Doi:10.1016/S0305-0548(00)00024-1.
15. Jeet, V., & Kutanoglu, E. (2007). Lagrangian relaxation guided problem space search heuristics for generalized assignment problems. European journal of operational research, 182, 1039–1056. Doi:10.1016/j.ejor.2006.09.060.
16. Martello, S., & Toth, P. (1881). An algorithm for the generalized assignment problem. In: J. P. Brans (ed.), Operational Research '81, North-Holland, Amsterdam, pp. 589-603.
17. Nauss, R. M. (2003). Solving the generalized assignment problem: An optimizing and heuristic approach. Informs Journal on Computing, 15, 249–266. Doi:10.1287/ijoc.15.3.249.16075.
18. Nowakovski, J., Schwärzler, W., & Triesch, E. (1999). Using the generalized assignment problem in Scheduling the ROSAT space telescope. European journal of operational research, 112, 531–541. Doi:10.1016/S0377-2217(97)00408-6.
19. Nutov, Z., Beniaminy, I., & Yuster, R. (2006). A $(1-1/e)$-approximation algorithm for the generalized assignment problem. Operations Research Letters, 34, 283–288. Doi:10.1016/j.orl.2005.05.006.
20. Öncan, T. (2007). A survey of the generalized assignment problem and its applications. INFOR: Information Systems and Operational Research, 45, 123–141. Doi:10.3138/infor.45.3.123.
21. Özbakir, L., Baykasoglu, A., & Tapkan, P. (2010). Bees algorithm for the generalized assignment problem. Applied Mathematics and Computation, 215, 3782–3795. Doi:10.1016/j.amc.2009.11.018.
22. Pigatti, A., De Aragão, M. P., & Uchoa, E. (2005). Stabilized branch-and-cut-and-price for the generalized assignment problem. Electronic Notes in Discrete Mathematics, 19, 389–395. Doi:10.1016/j.endm.2005.05.052.
23. Posta, M., Ferland, J. M., & Michelon, P. (2012). An exact method with variable fixing for solving the generalized assignment problem. Computational Optimization and Applications, 52, 629–644. Doi:10.1007/s10589-011-9432-0.
24. Schmoys, D. B., & Tardos, E. (1993). An approximation algorithm for the generalized assignment problem. Mathematical Programming, 62, 461–474. Doi:10.1007/BF01585178.
25. Woodcock, A. J., & Wilson, J. M. (2010). A hybrid tabu search/branch and bound approach to solving the generalized assignment problem. European journal of operational research, 207, 566–578. Doi:10.1016/j.ejor.2010.05.007.
26. Yagiura, M., Ibaraki, T., & Glover, F. (2004). An Ejection chain approach for the generalized assignment problem. Informs Journal on Computing, 16, 133–151. Doi:10.1287/ijoc.1030.0036.
27. Yagiura, M., Ibaraki, T., & Glover, F. (2006). A path relinking approach with ejection chains for the generalized assignment problem. European Journal of Operational Research, 169, 548–569. Doi:10.1016/j.ejor.2004.08.015.

Algorithm ILH

Input m, n, a, b, c
Output Best feasible assignment $\{s_j, j \in J\}$ and its cost ub

Initialization

01. Apply Cplex to find an initial feasible assignment $\{s_j, j \in J\}$, and let ub be its cost
02. Submit this feasible assignment to the shift-interchange improvement procedure (update $\{s_j, j \in J\}$ and ub whenever an improvement is achieved)
03. $\rho \longleftarrow 2, \pi^{(0)} \longleftarrow \mathbf{0}$

Loop $(k = 0, \cdots, NBIT)$

1. Compute the reduced costs as in (7), the coefficients $t_j, j \in J$ as in (8) and define problem MONOTONE-01IP as in (2-5)
2. Solve problem MONOTONE-01IP
3. Submit the feasible assignment found to the shift-interchange improvement procedure, and update $\{s_j, j \in J\}$ and ub if there is an improvement
4. Solve the relaxed problem and compute the value $L\left(\pi^{(k)}\right)$
5. $\sigma_i\left(\pi^{(k)}\right) \longleftarrow b_i - \sum_{j \in J} a_{s_j j}$ for $i \in I$
6. $\pi_i^{(k+1)} \longleftarrow \pi_i^{(k)} + \rho \frac{1.2 \times ub - L\left(\pi^{(k)}\right)}{\left\|\sigma\left(\pi^{(k)}\right)\right\|^2} \sigma_i\left(\pi^{(k)}\right)$ for $i \in I$
7. For $i \in I$, if $\pi_i^{(k+1)} > 0$ then $\pi_i^{(k+1)} \longleftarrow 0$
8. $\rho \longleftarrow \rho \times 0.98$

Algorithm 1: Pseudo-code of algorithm ILH

Instances	m	n
C10400, D10400, E10400	10	400
C20400, D20400, E20400	20	
C40400, D40400, E40400	40	
C15900, D159400, E15900	15	900
C30900, D30900, E30900	30	
C60900, D60900, E60900	60	
C201600, D201600, E201600	20	1600
C401500, D401600, E401600	40	
C801600, D801600, E801600	80	

Table 1. Sizes of the instances

		Cplex		ILH($m/80$)		ILH($m/40$)		ILH($m/20$)		ILH($m/10$)	
	LB	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time
C10400	5,596	5,645	0.28	**5,597**	24.39	**5,597**	42.01	**5,597**	73.08	**5,597**	130.53
C20400	4,781	4,839	0.57	**4,782**	43.94	4,783	79.49	**4,782**	151.82	**4,782**	297.07
C40400	4,244	4,289	1.99	**4,244**	44.37	**4,244**	53.01	4,245	115.47	**4,244**	143.82
C15900	11,339	11,368	0.76	11,342	40.73	11,343	77.82	11,342	149.98	**11,341**	290.62
C30900	9,982	10,027	2.50	9,985	82.16	**9,984**	153.51	9,985	286.87	**9,984**	559.62
C60900	9,325	9,364	9.54	9,330	143.11	9,329	261.62	9,329	486.72	**9,328**	885.84
C201600	18,802	18,880	1.97	18,805	61.11	**18,804**	118.15	18,805	208.04	**18,804**	404.78
C401600	17,144	17,204	6.00	17,148	113.61	**17,147**	209.92	**17,147**	396.88	**17,147**	771.38
C801600	16,284	16,346	23.50	16,290	220.34	**16,288**	397.06	**16,288**	725.42	**16,288**	1,376.10
Av. dev. from LB		0.603%		0.026%		0.022%		0.024%		0.018%	
D10400	24,959	25,139	0.16	24,976	58.85	24,974	101.33	24,972	199.10	**24,970**	389.89
D20400	24,561	24,855	0.49	24,603	113.93	24,605	215.37	24,600	393.76	**24,597**	773.32
D40400	24,350	24,639	1.06	24,464	219.21	24,465	396.37	24,448	775.16	**24,443**	1,522.54
D15900	55,403	55,588	0.76	55,432	93.22	55,430	162.45	55,429	310.29	**55,427**	605.53
D30900	54,833	55,011	2.22	54,897	178.81	54,895	325.14	**54,888**	608.69	**54,888**	1,193.53
D60900	54,551	54,921	4.90	54,696	368.61	54,670	645.22	54,681	1,223.26	**54,659**	2,400.01
D201600	97,823	97,981	2.59	97,856	151.44	97,868	246.77	97,858	449.42	**97,855**	850.99
D401600	97,105	97,437	6.87	97,173	287.34	97,173	478.11	97,170	891.38	**97,166**	1,686.06
D801600	97,034	97,523	15.78	97,196	688.81	97,186	1,078.81	97,171	1,826.82	**97,158**	3,393.68
Av. dev. from LB		0.604%		0.157%		0.152%		0.139%		0.126%	
E10400	45,745	46,629	0.16	45,747	54.03	45,748	99.08	**45,746**	181.07	**45,746**	361.07
E20400	44,876	45,656	0.70	44,879	85.49	**44,877**	148.98	**44,877**	279.02	**44,877**	485.83
E40400	44,557	45,871	2.23	44,586	177.23	44,592	332.68	44,576	556.67	**44,575**	1,068.29
E15900	102,420	102,862	1.08	102,433	81.85	102,426	156.07	102,426	293.63	**102,421**	573.12
E30900	100,426	101,878	3.26	100,436	147.43	**100,430**	284.25	100,437	490.99	100,432	890.18
E60900	100,144	101,719	14.34	100,188	302.08	100,178	573.76	100,184	1,061.75	**100,164**	1,886.33
E201600	180,642	181,949	3.51	180,667	122.18	180,654	221.92	180,648	408.81	**180,646**	774.14
E401600	178,293	179,775	10.63	178,318	226.59	178,301	418.10	178,302	782.48	**178,300**	1,444.80
E801600	176,816	178,280	36.29	176,851	438.20	176,834	790.82	176,833	1,446.39	**176,829**	2,783.19
Av. dev. from LB		1.382%		0.020%		0.017%		0.013%		0.009%	

Table 2. Computational results of algorithm ILH

		Preprocessing			ILH($m/80$)			
	LB	Remain. tasks	Fixed cost	Prep. time	Cplex cost	Best cost	ILH time	Total time
C10400	5,596	158	3,318	0.36	5,612	5,598	9.93	10.29
C20400	4,781	198	2,387	0.68	4,817	4,783	18.68	19.36
C40400	4,244	215	1,942	2.20	4,275	4,245	12.51	14.71
C15900	11,339	401	6,123	0.95	11,382	11,342	18.32	19.27
C30900	9,982	478	4,613	2.87	10,011	9,986	36.46	39.33
C60900	9,325	544	3,661	10.22	9,356	9,329	59.22	69.44
C201600	18,802	761	9,728	2.39	18,856	18,805	25.59	27.98
C401600	17,144	888	7,553	6.86	17,230	17,149	50.85	57.71
C801600	16,284	1,116	4,922	25.22	16,386	16,289	94.00	119.22
Av. dev. from LB					0.465%	0.032%		
D10400	24,959	113	18,955	0.20	25,079	24,979	24.99	25.19
D20400	24,561	173	15,793	0.56	24,835	24,608	51.11	51.67
D40400	24,350	257	9,763	1.25	24,708	24,470	98.00	99.25
D15900	55,403	408	34,369	0.95	55,599	55,427	40.76	41.71
D30900	54,833	431	33,484	2.59	56,010	54,923	84.85	87.44
D60900	54,551	734	11,198	5.56	54,986	54,693	163.96	169.52
D201600	97,823	751	59,388	2.98	98,405	97,864	62.05	65.03
D401600	97,105	1,074	34,237	7.81	97,359	97,180	122.90	130.71
D801600	97,034	1,438	10,899	17.58	97,481	97,185	316.66	334.24
Av. dev. from LB					0.847%	0.167%		
E10400	45,745	91	35,559	0.23	46,603	45,746	19.16	19.39
E20400	44,876	132	30,267	0.78	46,770	44,896	30.84	31.62
E40400	44,557	181	25,059	2.46	45,758	44,635	32.53	34.99
E15900	102,420	354	63,268	1.25	103,621	102,421	33.82	35.07
E30900	100,426	372	60,301	3.57	102,061	100,437	65.13	68.70
E60900	100,144	485	47,541	15.01	102,092	100,229	127.86	142.87
E201600	180,642	710	102,443	3.89	181,435	180,654	51.66	55.55
E401600	178,293	858	84,996	11.36	179,271	178,307	99.75	111.11
E801600	176,816	999	68,245	38.27	178,248	176,869	201.33	239.60
Av. dev. from LB					1.705%	0.039%		

Table 3. Computational results of algorithm ILHRED($m/80$)

	LB	Allowed time	ILH($m/80$) Cost	CPLEX Cost	Allowed time	ILH($m/10$) Cost	CPLEX Cost
C10400	5,596	24.39	**5,597**	**5,597**	130.53	**5,597**	**5,597**
C20400	4,781	43.94	**4,782**	4,783	297.07	**4,782**	**4,782**
C40400	4,244	44.37	**4,244**	4,245	143.82	**4,244**	**4,244**
C15900	11,339	40.73	**11,342**	**11,342**	290.62	**11,341**	**11,341**
C30900	9,982	82.16	**9,985**	9,989	559.62	**9,984**	9,986
C60900	9,325	143.11	**9,330**	9,329	885.84	**9,328**	9,329
C201600	18,802	61.11	**18,805**	18,808	404.78	**18,804**	18,805
C401600	17,144	113.61	**17,148**	17,150	771.38	**17,147**	17,148
C801600	16,284	220.34	**16,290**	16,320	1,376.10	**16,288**	16,292
Av. dev. from LB			0.026%	0.056%		0.018%	0.026%
D10400	24,959	58.85	**24,976**	24,983	389.89	**24,970**	24,976
D20400	24,561	113.93	**24,603**	24,617	773.32	**24,597**	24,606
D40400	24,350	219.21	**24,464**	24,505	1,522.54	24,443	**24,442**
D15900	55,403	93.22	**55,432**	55,466	605.53	**55,427**	55,435
D30900	54,833	178.81	**54,897**	54,937	1,193.53	**54,888**	54,922
D60900	54,551	368.61	**54,696**	54,888	2,400.01	**54,659**	54,862
D201600	97,823	151.44	**97,856**	97,894	850.99	**97,855**	97,862
D401600	97,105	287.34	**97,173**	97,295	1,686.06	**97,166**	97,295
D801600	97,034	688.81	**97,196**	97,484	3,393.68	**97,158**	97,484
Av. dev. from LB			0.157%	0.291%		0.126%	0.233%
E10400	45,745	54.03	**45,747**	**45,747**	361.07	**45,746**	45,747
E20400	44,876	85.49	**44,879**	44,882	485.83	**44,877**	44,879
E40400	44,557	177.23	**44,586**	44,635	1,068.29	44,575	**44,574**
E15900	102,420	81.85	**102,433**	102,450	573.12	**102,421**	102,427
E30900	100,426	147.43	**100,436**	100,634	890.18	100,432	**100,429**
E60900	100,144	302.08	**100,188**	100,707	1,886.33	**100,164**	100,243
E201600	180,642	122.18	**180,667**	180,782	774.14	**180,646**	180,654
E401600	178,293	226.59	**178,318**	178,716	1,444.80	**178,300**	178,312
E801600	176,816	438.20	**176,851**	177,545	2,783.19	**176,829**	177,205
Av. dev. from LB			0.020%	0.191%		0.009%	0.056%

Table 4. Comparison of the ILH and Cplex by running them during the same amount of time

		ILH($m/80$) beginning with CPLEX				ILH($m/80$) beginning with MTH				
	LB	Cplex cost	Cplex time	Cost	Total time	MTH cost	MTH time	Cost at iter. 1	Cost	Total time
C10400	5,596	5,645	0.28	**5,597**	24.39	7,222	0.05	5,647	**5,597**	24.19
C20400	4,781	4,839	0.57	**4,782**	43.94	6,189	0.10	4,871	**4,782**	47.43
C40400	4,244	4,289	1.99	**4,244**	44.37	5,372	0.16	4,317	**4,244**	47.15
C15900	11,339	11,368	0.76	**11,342**	40.73	14,897	0.28	11,478	**11,342**	41.19
C30900	9,982	10,027	2.50	**9,985**	82.16	13,127	0.45	10,141	9,986	82.75
C60900	9,325	9,364	9.54	**9,330**	143.11	11,788	0.92	9,525	**9,330**	144.45
C201600	18,802	18,880	1.97	**18,805**	61.11	24,431	1.01	18,962	**18,805**	61.69
C401600	17,144	17,204	6.00	**17,148**	113.61	22,146	2.01	17,460	**17,148**	115.10
C801600	16,284	16,346	23.50	16,290	220.34	20,041	5.13	16,541	**16,289**	209.29
Av. dev. from LB		0.603%		0.026%		28.508%		1.515%	0.026%	
D10400	24,959	25,139	0.16	24,976	58.85	33,080	0.05	26,622	**24,974**	58.83
D20400	24,561	24,855	0.49	24,603	113.93	35,083	0.09	26,397	**24,600**	115.25
D40400	24,350	24,639	1.06	**24,464**	219.21	39,151	0.14	26,299	24,471	222.79
D15900	55,403	55,588	0.76	55,432	93.22	75,475	0.27	59,234	**55,426**	93.88
D30900	54,833	55,011	2.22	**54,897**	178.81	82,620	0.46	58,726	54,908	180.47
D60900	54,551	54,921	4.90	54,696	368.61	86,082	0.89	57,808	**54,688**	371.18
D201600	97,823	97,981	2.59	**97,856**	151.44	144,320	1.02	104,157	97,863	151.77
D401600	97,105	97,437	6.87	**97,173**	287.34	150,603	1.96	103,192	97,176	284.06
D801600	97,034	97,523	15.78	97,196	688.81	154,982	5.14	102,702	**97,187**	693.57
Av. dev. from LB		0.604%		0.157%		49.246%		6.744%	0.147%	

Table 5. Performance of the ILH beginning with "good " and "bad" initial solutions

		ILH(m/80)		ILH(m/80) without SIH	
	LB	Cost	Time	Cost	Time
C10400	5,596	**5,597**	24.39	**5,597**	23.61
C20400	4,781	**4,782**	43.94	**4,782**	43.54
C40400	4,244	**4,244**	44.37	4,245	53.18
C15900	11,339	**11,342**	40.73	**11,342**	39.16
C30900	9,982	**9,985**	82.16	9,986	82.58
C60900	9,325	**9,330**	143.11	**9,330**	140.22
C201600	18,802	**18,805**	61.11	**18,805**	53.98
C401600	17,144	**17,148**	113.61	**17,148**	108.61
C801600	16,284	16,290	220.34	**16,289**	211.22
Av. dev. from LB		0.026%		0.028%	
D10400	24,959	24,976	58.85	**24,975**	56.17
D20400	24,561	**24,603**	113.93	24,608	109.07
D40400	24,350	**24,464**	219.21	24,466	212.64
D15900	55,403	55,432	93.22	**55,431**	82.19
D30900	54,833	**54,897**	178.81	54,905	164.69
D60900	54,551	**54,696**	368.61	54,709	324.14
D201600	97,823	97,856	151.44	**97,855**	112.11
D401600	97,105	**97,173**	287.34	97,194	218.13
D801600	97,034	**97,196**	688.81	97,257	454.78
Av. dev. from LB		0.157%		0.220%	
E10400	45,745	45,747	54.03	**45,746**	53.03
E20400	44,876	44,879	85.49	**44,878**	91.47
E40400	44,557	44,586	177.23	**44,577**	180.73
E15900	102,420	**102,433**	81.85	102,435	77.25
E30900	100,426	100,436	147.43	**100,433**	149.24
E60900	100,144	**100,188**	302.08	100,208	302.33
E201600	180,642	180,667	122.18	**180,663**	107.04
E401600	178,293	178,318	226.59	**178,304**	212.79
E801600	176,816	176,851	438.20	**176,836**	418.27
Av. dev. from LB		0.020%		0.017%	

Table 6. Performance of the ILH with and without SIH

	LB	ILHRED($m/80$)	ILH($m/10$)	TSEC	PREC	TSBB(S)	TSBB(L)
C10400	5,596	**5,597**	**5,597**	**5,597**	**5,597**	**5,597**	**5,597**
C20400	4,781	**4,782**	**4,782**	**4,782**	**4,782**	4,783[2]	**4,782**
C40400	4,244	**4,244**	**4,244**	**4,244**	4,245[2]	4,246	4,245[2]
C15900	11,339	11,342[2]	**11,341**	**11,341**	**11,341**	11,345	11,342[2]
C30900	9,982	9,985[2]	**9,984**	9,985[2]	**9,984**	9,993	9,993
C60900	9,325	9,330[2]	**9,328**	**9,328**	**9,328**	9,339	9,337
C201600	18,802	18,805	18,804[2]	**18,803**	**18,803**	18,805	10,805
C401600	17,144	17,148	17,147[2]	17,147[2]	**17,145**	17,152	17,152
C801600	16,284	16,290	**16,288**	16,291	16,289[2]	16,297	16,290
Av. dev. from LB		0.032%	0.018%	0.021%	0.020%	0.063%	0.049%
D10400	24,959	24,976	24,970[2]	24,974	**24,969**	24,978	24,976
D20400	24,561	24,603	24,597[2]	24,614	**24,587**	24,665	24,631
D40400	24,350	24,464	24,443[2]	24,463	**24,417**	24,589	24,572
D15900	55,403	55,432	55,427[2]	55,435	**55,414**	55,465	55,462
D30900	54,833	54,897	54,888[2]	54,910	**54,868**	55,012	55,012
D60900	54,551	54,696	54,659[2]	54,666	**54,606**	54,986	54,785
D201600	97,823	97,856	97,855[2]	97,870	**97,837**	97,938	97,921
D401600	97,105	97,173	97,166[2]	97,177	**97,113**	97,467	97,328
D801600	97,034	97,196	97,158	97,109[2]	**97,052**	97,523	97,449
Av. dev. from LB		0.167%	0.126%	0.150%	0.072%	0.412%	0.322%
E10400	45,745	45,747	**45,746**	**45,746**	**45,746**	**45,746**	**45,746**
E20400	44,876	44,879	**44,877**	44,882	44,879[2]	44,899	**44,877**
E40400	44,557	44,586	44,575[2]	44,589	**44,574**	44,640	44,640
E15900	102,420	102,433	**102,421**	102,423	102,422[2]	102,428	**102,421**
E30900	100,426	100,436	**100,432**	100,442	100,434[2]	100,591	100,500
E60900	100,144	100,188	**100,164**	100,185	100,169[2]	100,441	100,363
E201600	180,642	180,667	**180,646**	180,647[2]	**180,646**	180,650	180,650
E401600	178,293	178,318	**178,300**	178,311	178,302[2]	178,500	178,394
E801600	176,816	176,851[2]	**176,829**	176,866	176,857	177,075	177,075
Av. dev. from LB		0.039%	0.009%	0.020%	0.012%	0.109%	0.077%

Table 7. Comparison of the heuristics from the solution quality point of view

	ILHRED($m/80$)	ILH($m/10$)	TSEC	PREC	TSBB(S)	TSBB(L)
C10400	103	1,305	3,000	3,000	10,000	100,000
C20400	194	2,971				
C40400	147	1,438				
C15900	193	2,906	10,000	10,000	10,000	100,000
C30900	393	5,596				
C60900	694	8,858				
C201600	280	4,048	50,000	50,000	50,000	500,000
C401600	577	7,714				
C801600	1,192	13,761				
Average	419	5,400	21,000	21,000	23,333	233,333
D10400	252	3,899	3,000	3,000	10,000	100,000
D20400	517	7,733				
D40400	993	15,225				
D15900	417	6,055	10,000	10,000	10,000	100,000
D30900	874	11,935				
D60900	1,695	24,000				
D201600	650	8,510	50,000	50,000	50,000	500,000
D401600	1,307	16,861				
D801600	3,342	33,937				
Average	1,116	14,239	21,000	21,000	23,333	233,333
E10400	194	3,611	3,000	3,000	10,000	100,000
E20400	316	4,858				
E40400	350	10,683				
E15900	351	5,731	10,000	10,000	10,000	100,000
E30900	687	8,902				
E60900	1,429	18,863				
E201600	556	7,741	50,000	50,000	50,000	500,000
E401600	1,111	14,448				
E801600	2,396	27,832				
Average	801	11,408	21,000	21,000	23,333	233,333

Table 8. Comparison of the heuristics from the computing time point of view. All times are SUN ULTRA II equivalent

Hybridation of PSO and SFLA for QoS aware web service composition

Sadouki Samia[1] , Tari Abdelkamel[1] , Boudries Fouzia[1]

1 : Le Laboratoire d'Informatique MEDicale; Faculté des Sciences exactes
Université de Bejaia, 06000 Bejaia - Algeria

Abstract

The goal of QoS aware web service composition is to find the best combination of services such that their aggregated QoS values should be optimized and QoS constraints set by consumers should also be satisfied. QoS aware web service composition is a global optimization problem belonging to NP-hard class given the number of available services. Thus, an optimal solution cannot be found by exact algorithm within a reasonable time. In this paper, a new meta-heuristic bio-inspired is presented to address the QoS aware service composition; it is based on hybrid Particle Swarm Optimization Shuffled frog leaping algorithm. Compared with original SFLA and PSO, the results of experimental evaluation show that our improvements significantly outperform the existing algorithm in execution time with better QoS performance.

Keywords: QoS, Optimization, Bio-inspired algorithms, SFLA, PSO, Web Service composition.

1. Introduction

With the emergence of cloud and SaaS, more and more web services will be available, thus, a large number of web services with the same functionalities and different Quality of Service (QoS such as price, response time, availability, reliability, reputation, security, throughput, …) can be found. Hence the appearance of a new emerging challenge: QoS aware web service composition 'QoS -WSC' [31] [22].

The goal of QoS aware service composition is to find the best combination of services such that their aggregated QoS values should be optimized. QoS-WSC is a global optimization problem belonging to NP-hard class given the number of available [13]. Thus, an optimal solution can't be found by exact algorithm within a reasonable time. Various approaches to address this problem have been proposed in literature [3], the majority of them don't address functional and non functional aspects at the same time.

A novel hybrid optimization method called P-SFLA is proposed, which introduces PSO to SFLA by combining the fast convergence speed of PSO, and global search strategy of SFLA, frog leaping in SFLA doesn't constraint to the limit length, which helps the frog getting out of the local optimum. We proposed to improve the leaping rules in SFLA algorithm by applying a PSO particle moving, which helps the population evolving more efficiently.

A description and mathematical formulation of QoS aware composition problem and the two algorithms PSO and SFLA is discussed in Section 3. In section 4 the hybrid PSO SFLA is presented, while the implementation and the evaluation are illustrated in Section 5.

2. Related work

Web service composition combines available services to provide new functionality. Many services with equivalent functionalities can be published by different service providers with different quality of service. Since the QoS of the service selected has a direct impact on the composition of the service obtained, the best set of selected services is the set that optimizes the QoS of the composition.

In literature review, many approaches are proposed to address the problem of QoS -aware web services selection for composition which was modelized as a combinatorial optimization problem with an optimal solution (exact algorithm) or near-optimal (approximate, heuristics).

Zeng et al. [21] proposed a QoS aware middleware for dynamic and quality-driven service composition. They used an integer programming based approach to solve the problem of global optimization in web service selection. Alrifai et al. [3] initially used mixed integer programming (MIP) to find the optimal decomposition of global QoS constraints into local constraints. Then, they use a distributed local selection method to find the best web services that could satisfy these local constraints. Lo of recent works use decomposition of global QoS constraints into local constraints [22] [17]. Exact optimization methods are very effective when the size of the problem is small. However, these methods suffer from poor scalability due to the exponential time complexity.

Although, as stated earlier, some heuristic algorithms help these algorithms to converge sooner, different kinds of meta-heuristic algorithms are proposed to find a near-to-optimal solution more efficiently than exact solutions; these algorithms are inspired by the nature [19][16].

A genetic algorithm is one of this meta-heuristic, where the composite service is encoded as a chromosome, and the fitness function is the aggregated QoS of the composite service. The fitness will increase from generation to generation. We recall that a fittest chromosome represents the optimal solution of the problem [10]. Despite that Genetic Algorithm is a powerful approach to solve combinatorial optimizing problems; it suffers from premature convergence and fixing key parameters for QoS -based web service composition.

Ant Colony Optimization (ACO) algorithm is also used for solving the QoS-WSC problem [7] [15]. ACO is a population based Meta-heuristic which is used to find approximate solutions to difficult optimization problems. In ACO, a set of software agents called artificial ants search for good solutions to a given optimization problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a graph. The artificial ants incrementally build solutions by moving on the graph. There is only one kind of pheromone in ACO, which can't satisfy question of multiple attributes in web services composition [9].

Recently, particle swarm optimization PSO showed its strong searching ability, and is also applied for finding optimized service composition [11] [20]. However, similar with genetic algorithms, by applying PSO when searching for the optimal composition solution, the problem of early stagnancy in a local optimum cannot be avoided.

Other authors modelled the QoS -WSC as multi-objective problem both in exact and near to optimal algorithm [2].

For the purpose of solving QoS-WSC, we provide a new meta-heuristic algorithm (Shuffled frog leaping algorithm) SFLA, which proved its effectiveness for the resolution of different optimization problem [8] [14], in [1] authors use the original SFLA algorithm in cloud service composition, but the SFLA is characterized by low convergence speed [3], in order to improve the ability of SFLA, we proposed to introduces PSO to SFLA by combining the fast convergence speed of PSO and global search strategy of SFLA.

3. Background

3.1. QoS aware web service composition

Quality of service « QoS »

Quality of service « QoS » is an indicator to measure and describe some performance characteristics of a service such as response time, availability, price, reputation etc [18]. Lets VQS a set of quality attributes of the service S, $\mathrm{VQS} = \{\mathrm{VQS(q)}, \mathrm{q} = 1 : \mathrm{nbq}\}$ where $\mathrm{VQS(q)}$ determines the value of the qth quality attribute of S and nbq is the number of quality attributes.

Web service composition

Web service composition comes from software reuse. Its basic idea is combining the existing web services according to a certain relation to construct a new or better web service to satisfy a complex user's requirements [4].

A composite service CS is a triple $CS = < S, R, VQC >$ where:

- $S = \{S_c, c = 1: nbc\}$ Is a subset of registered abstract services, which can be composed together and satisfy the user's needs. S_c may be an atomic service or a composite service. Each web service S_c has a set of concrete web services offering the same functionalities with different QoS designed by a class Cc.
- $R = \{\bullet, !, \oplus, \otimes\}$ is a set of composite operators where $(\bullet)$ is a sequential operator, $(!)$ is a loop operator, $(\otimes)$ is a parallel operator and $(\oplus)$ is a conditional operator.

  Web service composition model is specified as a workflow consisting of a set of abstract services. At run time, concrete Web service is selected, and invoked for each abstract service.
- QoS of a composite service is defined as : $VQC = \{VQC(q), q = 1 : nbq\}$ where $VQC(q)$ determines the value of the qth quality attribute of CS and nbq is the number of quality attributes.

Table 1 illustrates the computation of QoS of the composition in different cases.

Attributes	Sequential	Parallel	Conditional	Loop
Additive	$\sum_{i=1}^{n} VQS_i(q)$	$\sum_{i=1}^{n} VQS_i(q)$	$\sum_{i=1}^{n} VQS_i(q) * pi$	$n * VQS_i(q)$
Multiplicative	$\prod_{i=1}^{n} VQS_i(q)$	$\prod_{i=1}^{n} VQS_i(q)$	$\sum_{i=1}^{n} VQS_i(q) * pi$	$VQS_i(q)^n$
Max-operator	$\sum_{i=1}^{n} VQS_i(q)$	$\max VQS_i(q)$	$\sum_{i=1}^{n} VQS_i(q) * pi$	$n * VQS_i(q)$

Table1: The aggregation of attributes for different composition structure

Objective function

The problem of finding the best service composition without enumerating all possible combinations is considered as an optimization problem, in which the objective function value has to be optimized. The objective function $F(CS)$ is a weighted Somme of the differences QoS values of the composition. $F(CS)$ is computed as follow:

$$F(CS) = \sum_{q=1}^{nbq} w_q * VQC(q) \qquad (1)$$

Where $\sum_{q=1}^{nbq} w_q = 1$, w_q presents the user's preference weight .

We implement QoS of all concrete services as a multidimensional matrix Figure 1.

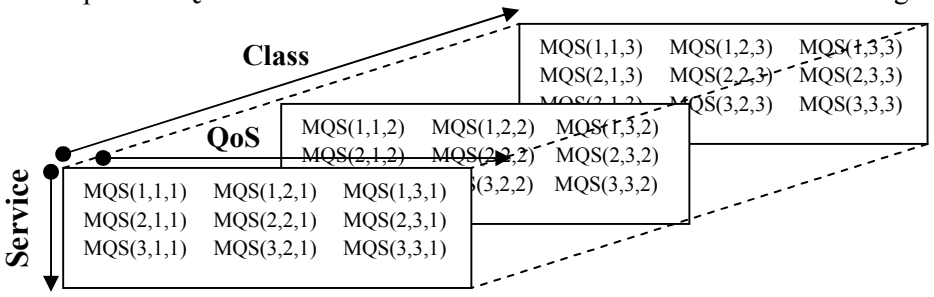


Figure 1: Multidimensional matrix for the QoS

Where: $MQS(s,q,c)$ is the qth propriety of QoS value and the index of the concrete service in a service class c.

In order to evaluate a given composition CS, an objective function is used to map all the vectors of QoS into a single value (mono-objectif problem).

A normalization of the values of the QoS attributes to the same scale is realised in order to avoid inaccurate evaluation due to different measurement metrics used for different QoS attributes. The QoS attributes can be classified into two groups: Positive and Negative attributes.

The values of positive attributes need to be maximized, while the negative values attributes must to be minimized.

In the normalization phase, positive and negative QoS attributes are scaled as follow:

Positive attributes:

$$MQS'(s,q,c) = \frac{MQS(s,q,c) - Qmin(q)}{Qmax(q) - Qmin(q)} \qquad (2)$$

Positive attributes:

$$MQS'(s,q,c) = \frac{MQS(s,q,c) - Qmin(q)}{Qmax(q) - Qmin(q)} \qquad (3)$$

Where $Qmax(q)$, $Qmin(q)$ are the maximum and minimum values of the qth attribute and $MQS(s,q,c)$ is the value of the qth attribute for a selected candidate service s in the class c.

3.2. Shuffled Frog Leaping Algorithm (SFLA)

SFLA is a meta-heuristic for solving discrete optimization problems. it is inspired by the interactive behaviour and global exchange of information of frogs searching for food laid on discrete stones randomly located in a pond. It combines the advantages of Evolutionary Algorithms and Swarm based Algorithms which are inspired by the natural evolution and collective behaviour in animals respectively [8] [5].

The algorithm uses mimetic evolution in the form of infection of ideas from one individual to another in a local search. A shuffling strategy allows for the exchange of information between local searches to move toward a global optimum [8].

SFLA involves a population of frogs (every frog represents a feasible solution) with the same structure but different adaptation. The population is partitioned into a number of groups referred as a memeplex. The algorithm performs an independent local search within each memeplex; and after certain number of local iterations, the whole population of frogs are shuffled and reorganized into new memeplexes to ensure global optimization [14].

The different steps of SFL algorithm are described as follows:

Step 1: generate randomly a population of P frogs.

Step 2: calculate the objective function value of all frogs

Step 3: Sort the frogs in ascending order according to their objective function value.

Step 4: Partition P into M memeplex, according to the formula: $(k + m(j-1))$thfrog goes to the kth memeplex in the jth position with $j = 1,,N \quad ,k = 1,,M$

Step 5: For each memeplex;

- Determine Fb and Fw, Fb (Fw)is the best frog (worst frog respectively) according to the objective function value.

- Improve the worst frog position using Eqs. (4) and (5);

  Change in frog position:

  $$Di = r\,and() (Fb - Fw), \quad (-Dmax \leq Di \leq Dmax) \quad (4)$$

  Position update:

  $$Fw = Fw + Di \quad (5)$$

  rand() : a random number between 0 and 1;

  Dmax : the maximum allowed change in a frog's position.

- Repeat for a specific number of local iterations Liter;

Step 6: if Liter go to step 7 else go to Step 5

Step 7: shuffle all the memeplexes and Sort the frogs in ascending order.

Step 8: if Giter (number of global iterations) return Fg the best solution, else go to step 4.

3.3. Particle Swarm Optimization

PSO is a population-based stochastic optimization technique developed by Kennedy and Eberhart in 1995. It is inspired by the social behavior of insect colonies, bird flocks, fish schools and other animal societies [12].

PSO has good performance and requires low computational cost. It is effective and easy to implement as it uses numerical encoding. A particle in PSO is analogous to a fish or bird moving in the D-dimensional search space.

A particle status on the search space is characterized by its position and velocity. Each particle position at any given time is influenced by both its best position called pBest and the best position in the swarm referred as gBest.

The position and velocity are updated in every generation as follows:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (pBest_i - X_i^k) + c_2 r_2 (gBest - X_i^k) \quad (6)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (7)$$

where V_i^{k+1} is the velocity of particle i at iteration $k+1$, V_i^k is the velocity of particle i at iteration k, ω is the inertia weight, and c_1, c_2 are the acceleration coefficients (cognitive and social coefficients), r_1 and r_2 are the random numbers between 0 and 1, X_i^k is the current position of particle i at the kth iteration, $pBest_i$ is the best previous position of the ith particle, $gBest$ is the position of the best particle in the swarm, and X_i^{k+1} is the position of ith particle at $k+1$ iteration.

The procedure for original PSO is as follows:

Step 1: Initialize a population of particles with random positions and velocities.

Step 2: Evaluate the objective values of all particles, set $pBest$ of each particle equal to its current position, and set $gBest$ equal to the position of the best initial particle.

Step 3: Update the velocity and the position of each particle according to (6) and (7).

Step 4: Map the position of each particle in the solution space and evaluate its fitness value according to the desired optimization fitness function.

Step 5: For each particle, compare its current fitness value with its $pBest$ value. If the current value is better, then update $pBest$ with the current position.

Step 6: Determine the best particle of the swarm. If the fitness value is better than $gBest$, updating $gBest$ with this fitness value.

Step 7: If the stopping criterion is met, then output $gBest$ and its objective value; otherwise, go Step 2.

4. Approach description (P- SFLA)

The shuffling process of SFLA helps the full exchange of information and benefits the algorithm's global search ability. SFLA is characterized by low convergence speed. In this paper, to improve the ability of SFLA, we proposed an hybrid algorithm PSO with SFLA. This hybridation integrates the fast convergence speed of PSO and the global search ability of SFLA, which can not only avoid trapping into local optimum and be close to the global optimum solution with higher precision, but also speeds up the convergence.

Mainly, we have integrated the particle moving formula (6) and (7) given in section 3.3 of PSO in the step of leaping rule in SFLA process.

The proposed algorithm is given as follows:

---

1. *Begin;*
2. *Generate random population of P solutions (frogs);*
3. *For each frog $F \in P$: calculate the objective function value;*
4. *Sort the population P in ascending order of their objective function value;*
5. *Divide P into M memeplexes;*
6. *Determine the best global frog Fg.*
7. *For each memeplex;*
   - *7.1 Determine the best and worst frogs Fb, Fw;*
   - *7.2 Improve the worst frog position using the PSO moving:*
   - *7.3 Repeat for a specific number of iterations Liter;*
8. *End;*
9. *Combine the evolved memeplexes;*
10. *Sort the population P in ascending order of their objective function value;*
11. *If Giter then the best solution= P(1),*
    else goto 5;
12. *End;*

First, an initial population of solutions is generated randomly, P frogs. Frogs are sorted in ascending order of their objective function values. Then, the entire population is partitioned into M sub-populations (memeplexes) such that each memeplex containing N frogs ($M * N = p$). Second, within each memeplex, frogs with the best and the worst objective function value are identified as Fb and Fw, respectively. Also, the frog with the global best objective function value in the population is identified as Fg. Then, an evolution process is applied to improve only the frog Fw in each cycle using PSO particle moving formula.

The improvements of Fw continue for a specific number of evolutionary iterations Liter within each memeplex. After the local searching for all m memeplexes, all frogs in all memeplexes are shuffling (mixed) and reordered into M memeplexes. If convergence criterion is satisfied then Fg is returned as the best solution. Otherwise the algorithm reiterates until Giter.

5. Experimentation

In this section, we report the results of simulation experiments used to study the performance of our algorithm (P-SFLA) in comparison with the original SFLA and original PSO.

The algorithms are implemented with Matlab R2012 on a HP dc7900 machine with 2 Intel Duo 2.4 GHZ processors and 2 GB RAM.

Dataset and test case generation

The test cases are generated by varying the number of service classes in the workflow from 10 to 100 with a step size of 20. The number of service candidates per class varies from 100 to 1000 with steps of 200 and QoS constraints number from 5 to 30. The value of each QoS attributes for each candidate service and the weight for each attribute is randomly generated, satisfying the requirement that summation of all the weights equal to 1. For each test case with a different number of service classes or service candidates per class, each algorithm is executed 50 times, and the average value is then used to evaluate the algorithm's performance.

Metrics and result analysis

In the literature, the metrics used to evaluate these kinds of algorithms are:

Running time (RT): Represents the CPU time required for an algorithm to obtain a solution.

Optimality (O): Represents the global objective function value obtained by the algorithm.

a) Running time evaluation

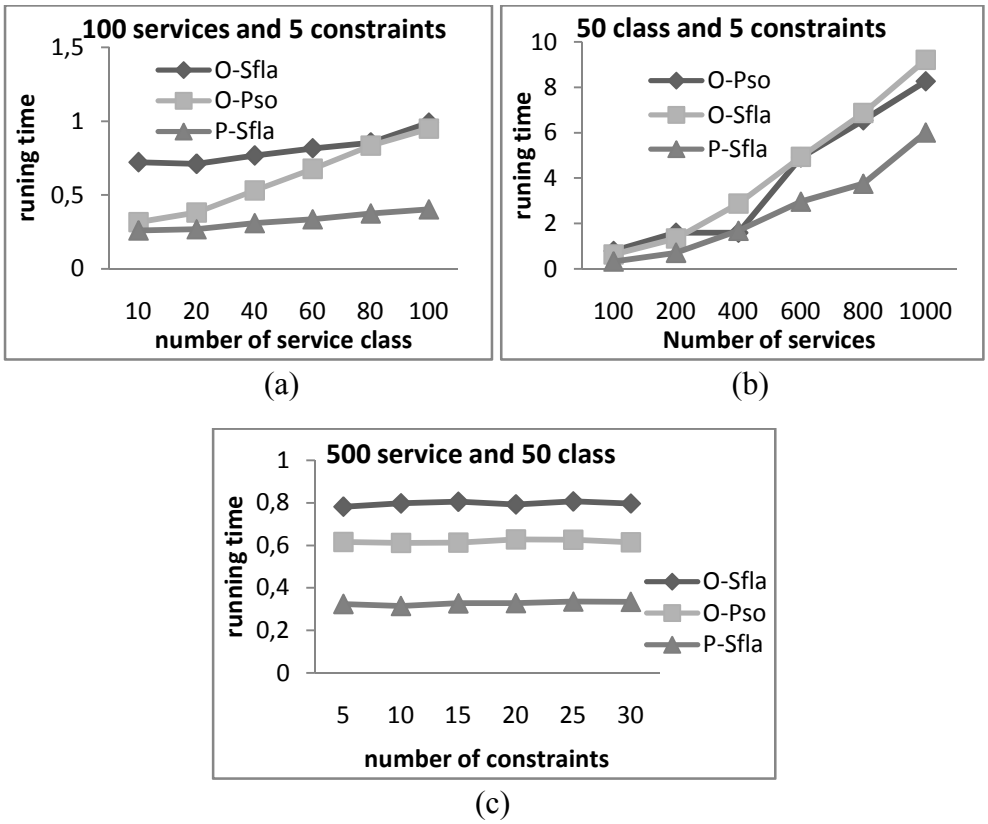


Figure 2: Running time evaluation (a,b,c)

The Figure 2 shows the running time comparison of the three algorithms. In Figure 2(a) the RT is measured with respect to an increasing number of service class with 100 services candidates per class and 5 constraints. It is noted that with increase in the number of service class, RT increases for both algorithms O-SFLA and O-PSO. While P-SFLA RT generally remains constant. Consequently, P- SFLA gives better results.

In Figure 2(b) RT is measured with respect to an increasing number of service candidates per class with 50 service classes and 5 constraints. This Figure shows an increasing of RT for the three algorithms with a low value for the P-SFLA. In Figure2(c), RT stays almost constant with increasing

number of constraints for all algorithms with a gap of about 0.3s favorably to P-SFLA. As demonstrated by these figures, the RT required by our algorithm is significantly shorter than the original PSO and SFLA.

b) Optimality evaluation:

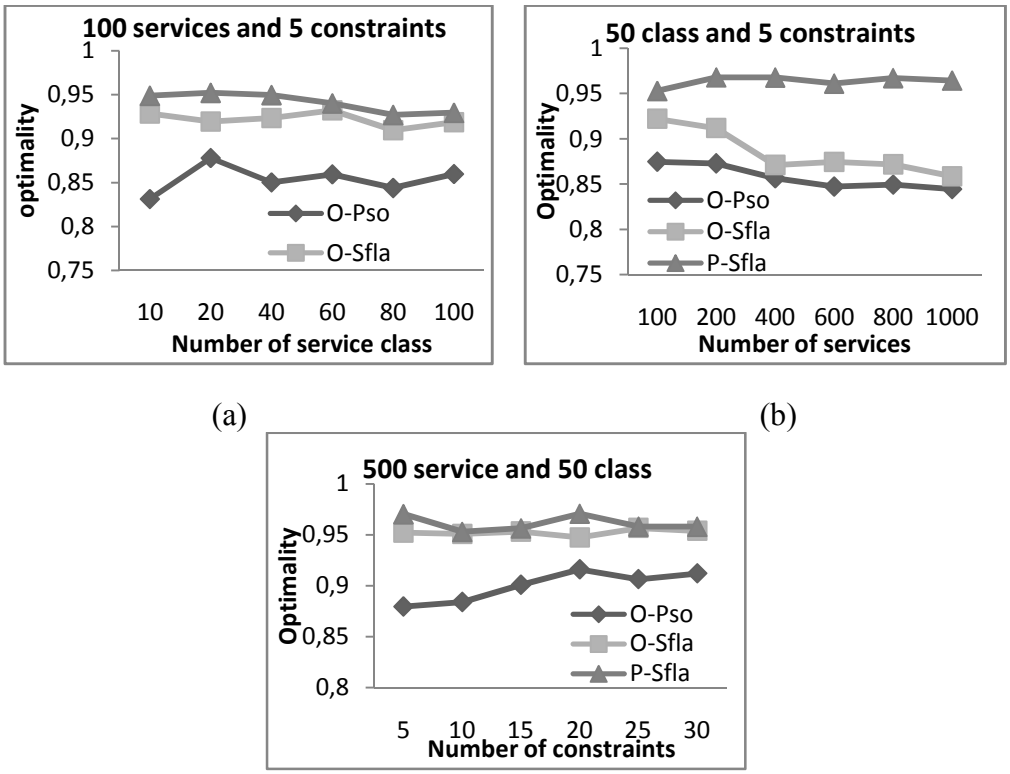


Figure 3: Optimality evaluation (a,b,c)

The optimality is given by the objective function value. In our case, we tent to maximize this value. From Figure 3, we can see the optimality comparison when the number of service classes ranges from 10 to 100 in (a), the number of service candidates varies from 100 to 1000 in (b) and the number of constraints from 5 to 30 in (c).The three figures show that the values of the objective function achieved by our algorithm P-SFLA are best comparing with the results obtained by O-PSO and O-SFLA.

The above experiments show that our solution can identify a composite service with an optimal value of objective function faster than PSO and SFLA one. Thus, our approach offers a more efficient and scalable solution for the service selection problem and is more suitable for a selection problem with a high complexity.

6. Conclusion

This paper presents an hybrid Particle Swarm Optimization Shuffled frog leaping algorithm for addressing the QoS aware web service composition. The original SFLA is modified with introducing the particle moving of PSO to SFLA leaping rule which combine the fast search strategy of PSO and global search strategy of SFLA.

Compared with SFLA and PSO, P-SFLA, for a large dataset, is shown to be an efficient algorithm for solving the QoS aware service web selection for composition problems. It offers important improvements and excellent performances in terms of running time, scalability and optimality evaluations, which measures the complexity of the algorithm.

References

[1] Ali Younes, Mohamed Essaaidi and Ahmed El Moussaoui. SFL Algorithm for QoS -based Cloud Service Composition. International Journal of Computer Applications, 97(17), pp. 42-49, 2014.

[2] Alrifai M, D. Skoutas, and T. Risse, Selecting skyline services for QoS -based web service composition, Proceedings of the 19th international conference on World Wide Web , pp. 11-20, 2010.

[3] Alrifai M, Risse T, Combining global optimization with local selection for efficient QoS -aware service composition, Proceedings of the 18th International World Wide Web Conference(WWW), pp. 881-890, 2009.

[4] Angel Lagares Lemos, Florian Daniel, and Boualem Benatallah. 2015. Web service composition: A survey of techniques and tools, ACM Computing Surveys, 48(3), Article 33, 41 pages. December 2015.

[5] Binitha S, S Siva Sathya, A Survey of Bio inspired Optimization Algorithms, International Journal of Soft Computing and Engineering (IJSCE),2(2), pp. 137-151, May 2012.

[6] Chellammal Surianarayanan, Gopinath Ganapathy, Manikandan Sethunarayanan Ramasamy, An approach for selecting best available services through a new method of decomposing QoS constraints, Service Oriented Computing and Applications 9(2), pp.107-138, 2015.

[7] Dhore S.R., M. Kharat, QoS based web services composition using ant colony optimization: mobile agent approach, International Journal of Advanced Research in Computer and Communication Engineering 1 (7), pp. 519-527, 2012.

[8] Eusuff M, Lansey K ,Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm, Journal of Water Resources Planning and Management, Volume 129, Issue 3, pp. 210-225,May 2003.

[9] Gatha Jayjit J. , Gohel Piyush V. ,A Novel Web Service Composition Using Ant Colony Optimization with Agent Based Approach, International Journal of Emerging Technologies and Innovative Research, Volume 2, Issue 5,pp. 1686-1688 May 2015.

[10] Gerardo Canfora, MassimiliaNo Di Penta, Raffaele Espositio, Maria Luisa Villani, An Approach for QoS -aware Service Composition based on Genetic Algorithms, GECCO '05 Proceedings of the 2005 conference on Genetic and eVolutionary computation, pp. 1069-1075, 2005.

[11] Guosheng Kang, Jianxun Liu, Mingdong Tang, Yu Xu,An Effective Dynamic Web Service Selection Strategy with Global Optimal QoS Based on Particle Swarm Optimization Algorithm, IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, 2012.

[12] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia. pp. 1942–1948. December 1995.

[13] Li J. et al., An efficient and reliable approach for quality-of-service-aware service composition, International Journal of Information Sciences Volume 269, pp. 238-254, 10 June 2014.

[14] Priyanka Roy , Pritam Roy, Abhijit Chakrabarti, Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect, Applied Soft Computing , Applied Soft Computing 13, pp. 4244-4252, 2013.

[15] Quanwang Wu, Qingsheng Zhu, Transactional and QoS -aware dynamic service composition based on ant colony optimization. Future Generation Comp. Syst. 29(5), pp.1112-1119, 2013.

[16] Rajeswary C., A survey on Efficient Evolutionary algorithms for Web Service Selection, International Journal of Management, IT and Engineering, 2(9), pp. 177-191, 2012.

[17] Sherry X. Sun , Jing Zhao, A decomposition-based approach for service composition with global QoS guarantees, Journal of Information Sciences, 199, pp. 138-153,September, 2012

[18] Sravanthi Kalepu, Shonali Krishnaswamy, Seng Wai Loke, Verity: A QoS Metric for Selecting Web Services and Providers, Proceedings of the Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03), 2004

[19] Wang, L., Shen, J. & Yong, J. (2012). A survey on bio-inspired algorithms for web service composition.Proceeding of the 16th International Conference on Computer Supported Cooperative Work in Design: IEEE, pp. 569-574,2012 .

[20] Wenbin Wang, Qibo Sun, Xinchao Zhao , Fangchun Yang, An improved Particle Swarm Optimization Algorithm for QoS -aware Web Service Selection in Service Oriented Communication, International Journal of Computational Intelligence Systems, Volume 3, Supplement 01, 2010.

[21] Zhenwu WANG, Benting Wan,Web Services Selection Based on Discrete Artificial Bee Colony Algorithm, International Journal of Digital Content Technology and its Applications(JDCTA) ,6(21),November 2012

[22] Zhi-Zhong Liu & Xiao Xue & Ji-quan Shen & Wen-Rui Li, Web service dynamic composition based on decomposition of global QoS constraints, The International Journal of Advanced Manufacturing Technology, 69(9), pp. 2247-2260, December 2013.

Systematic Search for Local-Search SAT Heuristics

Andrew W. Burnett and Andrew J. Parkes

ASAP Research Group
School of Computer Science, University of Nottingham, United Kingdom
psxab4 @nottingham.ac.uk, ajp @cs.nott.ac.uk

Abstract. Heuristics for local-search are a commonly used method of improving the performance of algorithms that solve hard computational problems. Generally these are written by human experts, however a long-standing research goal has been to automate the construction of these heuristics. In this paper, we investigate the applicability of a systematic search on the space of heuristics to be used in a local-search SAT solver.

1 Introduction

Algorithms to solve hard computational problems generally employ carefully-designed heuristics to improve their performance. For example, within local-search procedures for propositional satisfiability (SAT)[1]. A well-established class of algorithms are those inspired by "WalkSAT" [5, 4, 2], which begins with an assignment of variables, and in each iteration, a selected variable is chosen to have its assignment changed. It is the role of the "Variable Selection Heuristic" (VSH) to choose this variable on each iteration. Usually, the VSH is designed by a human expert; however, a long standing desire has been to automate the generation of heuristics to reduce the human element, and one well-established technique for this is using genetic programming (GP). Specifically, in the context of SAT, Fukunaga [1] used GP to build VSHs for use in a local-search SAT solver; the GP was able to build VSHs that performed competitively with two of the best heuristics at the time.

Separately, Katayama [3] observed that, within the context of functional programming, when given an initial function set, desired type signature, input and output, a brute-force search of the space of procedures was a viable method for finding programs that satisfy these constraints. In this paper, we observe that the space of potential VSHs for SAT can be small enough that it can be amenable to systematic search methods. We used a tree-based systematic search (to a certain depth) to find candidate VSHs. We show that this method is able to find well-performing VSHs, and so argue for a role for systematic search to complement sampling based methods such as GP.

2 Language used, experiments and results

Fukunaga [1] outlined several quantities associated with variables and clauses present in current heuristics as follows:

- *Broken Clause* - A clause that is broken (evaluates to $False$ under the current assignment)
- *Positive Gain* - Number of unsatisfied clauses that will become satisfied if a variable v is flipped
- *Negative Gain* - Number of satisfied clauses that will become unsatisfied if a variable v is flipped
- *Net Gain* - The net change in number of unsatisfied clauses if a variable v is flipped
- *Age* - The number of iterations since a variable v was last flipped

Figure 1 outlines the grammar used to generate VSHs. Space precludes giving full details, but we work with a similar language to Fukunaga [1]. Examples of VSHs defined using this grammar are given in Figure 3. Using this grammar we enumerated all possible VSHs that contain 15 or fewer symbols (we regard the number of symbols as the "depth" of the VSH). We chose 15 as it is just beyond depth 14, where an example of an already known well-performing WalkSAT VSH is

[1] SAT is the decision problem of whether a given propositional logic formula P, has a truth assignment to the variables in P such that P is satisfied. Generally, P is given in Clausal Normal Form (CNF), that is a conjunction of clauses, each clause is a disjunction of literals, and a literal is variable or its negation.

⟨*VSH*⟩ ::= IfRandLt ⟨*Probability*⟩ ⟨*VSH*⟩ ⟨*VSH*⟩
| GetBestVar ⟨*GainType*⟩ ⟨*VarSet*⟩
| GetSecondBestVar ⟨*GainType*⟩ ⟨*VarSet*⟩
| GetOldestVar ⟨*VSH*⟩ ⟨*VSH*⟩
| IfTabu ⟨*Age*⟩ ⟨*VSH*⟩ ⟨*VSH*⟩
| IfVarCompare ⟨*Comparator*⟩ ⟨*GainType*⟩ ⟨*VSH*⟩ ⟨*VSH*⟩
| IfVarCond ⟨*Comparator*⟩ ⟨*GainType*⟩ ⟨*Integer*⟩ ⟨*VSH*⟩ ⟨*VSH*⟩
| VarRandom ⟨ *VarSet* ⟩
| IfNotMinAge ⟨*VarSet*⟩ ⟨*VSH*⟩ ⟨*VSH*⟩
⟨*VarSet*⟩ ::= BrokenClause0
⟨*Comparator*⟩ ::= LessThan | LessThanEqual | Equal
⟨*GainType*⟩ ::= PosGain | NegGain | NetGain
⟨*Integer*⟩ ::= -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5
⟨*Age*⟩ ::= 5 | 10 | 20 | 30 | 40 | 50
⟨*Probability*⟩ ::= 0.1 | 0.3 | 0.5 | 0.7 | 0.9

Fig. 1. The grammar used to generate candidate VSHs. Compared to Fukunaga's work [1] we reduce the search space by only considering a single broken clause, *BrokenClause0*.

Table 1. A numerical analysis of the VSH at various depths, D. The ratio of numbers at depth n versus $n-1$; number with fitness ≥ 20, and the % with fitness ≥ 20

D	# VSP	ratio	≥ 20	% ≥ 20
5	1	-	0	0
6	24	24	0	0
7	189	7.9	7	3.7
8	614	3.2	30	4.9
9	1,272	2.1	0	0
10	3,996	3.1	6	0.2
11	12,173	3.0	380	3.1
12	62,238	5.1	3,742	6.0
13	223,155	3.6	11,708	5.2
14	714,542	3.2	23,384	3.3
15	2,264,475	3.2	58,161	2.6

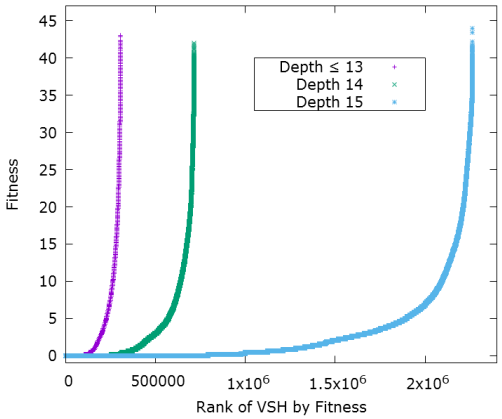


Fig. 2. Shows all heuristics at each level and their fitness sorted in ascending order

found. In Table 1 we can see that the number of VSHs grows rapidly with the number of symbols; tending towards increasing by around a factor of 3.2 per extra symbol. Note that the number of VSHs is still very small when compared to the size of the search space of most combinatorial optimisation problems. Hence, if the fitness of a VSH could be determined quickly, then the search would be easy; however, evaluating the fitness of a VSH requires testing it against several problem instances. In our case, this is 100 satisfiable SAT instances from SATLIB [2]. In an attempt to reduce computation time, not all VSHs were ran on all instances; rather, if a VSH performed poorly on an initial selection, it was not ran on the remaining instances. We repeated this experiment five times for each heuristic and, taking the number of times a solution was successfully found and the number of flips required to obtain these solutions, we computed a fitness value for a VSH. (details not given due to lack of space.) It should be noted that this fitness value should be considered "noisy" due, in part, to the random assignment initially generated for each VSH on each training instance.

In Table 1 we can see the number and % of VSHs at each depth with a score greater than 20; a value chosen because it is close to the fitness value of the WalkSAT VSH. We find these values to be surprisingly large. Figure 2 shows each heuristic, ordered by fitness, plotted against its fitness. We can see from this graph that a large proportion of these VSHs generated are unable to solve a single problem instance in our training set. Figure 3 shows two examples of VSHs written using the grammar in Figure 1; WalkSAT and VSH-1. VSH-1 is the VSH that scored highest according to our fitness function. This Figure also shows the results of running these two VSHs on satisfiable

[2] `http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html`

```
WalkSAT(0.5):
 IfVarCond NegGain == 0
   (GetBestVar NegGain BC0)
   (IfRandLt 0.5
     (GetBestVar NegGain BC0)
     (VarRandom BC0))
```

```
VSH-1:
 GetOldestVar
   (GetOldestVar
     (GetOldestVar
       (GetBestVar NetGain BC0)
       (GetBestVar NegGain BC0))
     (GetBestVar NetGain BC0))
   (GetBestVar PosGain BC0)
```

	uf50		uf100		uf150		uf200		uf250	
Name	SR	AFS	SR	AFS	SR	AFS	SR	AFS	SR	AFS
WalkSAT	**1.0**	**790**	**0.998**	**4036**	0.95	11833	0.853	11550	0.858	22514
VSH-1	0.998	319	0.994	1649	**0.975**	**4357**	**0.933**	**7500**	**0.957**	**13706**

Fig. 3. Top Left: WalkSAT (at noise 0.5). Due to redundancies in the language, for each WalkSAT VSH with noise p, there are two semantically identical VSH produced by changing *Equal* to *LessThanEqual*. Top Right: VSH-1. Bottom: Results of running the heuristics VSH-1 and WalkSAT on the testing sets. Showing the median Success Rate (SR) of instances solved and Average Flips to Solution(AFS)

instances from SATLIB not included in our training set. We can see that, although WalkSAT has a higher success rate on easier problem sets, conversely, on harder problem sets VSP-1 outperforms WalkSAT in both success rate and number of flips. Suggesting that, while our fitness function is noisy, it is a good indicator of whether a VSH performs well for problems of larger size.

3 Conclusions and Future Work

We have given a method to automatically build effective heuristics, with the aim to save time of human experts and potentially allowing partial specialisation to particular sets of problem instances. The crucial observation is that, with reasonable restriction of the language, the space of "reasonable" VSHs is surprisingly small and, as a proportion, the number of "good" VSHs is surprisingly large. This allowed us to search the space using a systematic search and produce a number of VSHs that outperform the original WalkSAT on some training sets (we do not claim here that VSH-1 is more effective across a wider set of instances, only that it worked well on the instances used.) Besides avoiding repetition, a systematic search has the potential to discover 'isolated heuristics' that might not be easy to find using GP; some good heuristics could potentially have substructures that are not useful in general and so could be less likely to emerge in GP.

Future work should reduce the search space by taking account of semantic equivalence - VSHs that are syntactically different but semantically equivalent need to be recognised as such. Another potential future direction is in identifying poorly performing VSHs and not having to evaluate these at all using a surrogate fitness functions. In such systematic search a natural issue is what search ordering to use; in this preliminary study we have simply evaluated all within the depth limit; but a heuristic ordered "best first" should be considered using current knowledge about good VSHs leading to a form of large-scale neighbourhood search, possibly hybridising or complementing GP methods.

Acknowledgements: Andrew W. Burnett thanks the EPSRC for financial support.

References

1. Alex S Fukunaga. Automated discovery of local search heuristics for satisfiability testing. *Evolutionary Computation*, 16(1):31–61, 2008.
2. Holger H Hoos and Thomas Stützle. *Stochastic local search: Foundations & applications*. Elsevier, 2004.
3. Susumu Katayama. Power of brute-force search in strongly-typed inductive functional programming automation. In *PRICAI 2004: Trends in Artificial Intelligence*, pages 75–84. Springer, 2004.
4. David McAllester, Bart Selman, and Henry Kautz. Evidence for invariants in local search. 1997.
5. Bart Selman, Henry A Kautz, and Bram Cohen. Noise strategies for improving local search. In *Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 337–343. American Association for Artificial Intelligence, 1994.

Metaheuristics for signal and image processing

Classification of Facial Image Expression using Binary Encoded Messy GA in Hybrid Metaheuristics Framework

Bhupesh Kumar Singh

***Abstract*: Classification is in the core of all the pattern recognition activities pursued so far. Hence improvement in classification performance has been a research endeavor. Ensemble of classifiers having different learning approaches is introduced in present work, combining evolutionary approach (Messy GA) along with SVM and ANN. Approaches, differing in knowledge representation, learning bias and exploration of search space reduce the probability of error correlation. Thus chances of increment in classification accuracy are high. The proposed approach is applied on facial expression classification problem for recognizing emotion using two well known databases Jaffe and Cohn-Kanade and a self prepared database named as Real. The significant amendment in feature subset selection approach with appropriate evolutionary algorithm, proper use of feature learning method and optimization technique, 85.4% classification rate is achieved with binary encoding.**

Introduction

Classification is the process of assigning individual input data to predefined classes on the basis of similarities defined by labeled dataset. There are various applications of pattern classification systems. The emerging approach is ensemble of classifiers as it provide more accurate results than with individual classifiers (Ledezma, 2010), (Hansen and Salamon, 1990). The methods of ensemble classifiers include manipulation based on training parameter, error function, feature space, output labels, clustering and training pattern (Rahman and Verma, 2013). Here a methodology has been used to classify facial expression.

The discipline of the facial expression classification has many applied manifestation social science, behavioral science, entertainment and computer science (Gunes, 2008; Li and Phung, 2010 and Filko 2013). Recently it was recognized as an effective mean of human computer interaction. Even further if captured exactly it can be used for surveillance camera capturing devious behavior, suspect of the crime scene, and other behavior symbol.

Basic Concept and Terminologies

1.1. Messy GA based approach

Standard Genetic algorithm (GA) is useful for solving the combinatorial optimization problem using fixed length coding and genetic operators (Goldberg, 1989). Messy GA is better for solving larger feature space problem as it selects features as subset while standard GA selects

features individually. The execution time taken by Messy GA is much less as compared to the one taken by standard GA **(**Goldberg, Korb and Deb, 1989, 1990). It processes a variable length string which can be underspecified or over specified. Application of Messy GA in feature selection has been widely reported (Whitley et al., 1997; Stringer and Annie, 2004; Hsiao, 2012). GA needs an encoding scheme for representing the individual messy gene which is collection of: Gene numbers and Allele values. Messy Chromosomes is collection of Messy Genes **.**The proposed Framework uses two representation schemes: Binary and Prufer for coding messy genes for showing the variation in results due to change the representation scheme. Binary Coding is bit representation scheme. In this type of encoding schemes chromosome are represented as string of 0 and 1 (Shivnandan, 2008). Flipping the bits is also easy in binary coding and based on the mutation rate. It is scheme in which fitness is depends on value and order of the strings. The formulation of optimization problem is given as: In matrix form, row is represented the expression as$E_1 E_2 \ldots\ldots\ldots\ldots E_a$. The column represents features $C_1 C_2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots C_b$.Where a, is expression of all images and b is no features.

Messy operators

Messy GA uses two main operator cut and splice rather traditional GAs crossover operator. Cut operator cuts chromosomes into two parts at random position while slice combines the chromosomes produced by cut operator to structure an individual. Cut probability is defined as

$$p_c = p_k(\lambda - 1) \tag{1}$$

Where $p_k$specified cut probability, λ represents length of string. Splice probability p_s is prescribed probability. These two operation works together for two different strings. These operations are similar to crossover operation of standard GA. Besides cut and splice operators Messy GA have another operator which is called allelic mutation operator which changes the mutation probability 0 to 1 and vice versa. Messy GA proceeds in three phases: Initialization, Primordial and Juxtaposition phase. In Initialization phase population is created randomly by the combination of individuals (combination of genotype i.e chromosomes and phenotype i.e. solution set). The created population is much large to fit all the genes. These genes work as candidate solution primordial phase. In primordial phase filtering and selection process takes place. Tournament selection scheme is used for selecting the best individuals which further

processes in next phase named as juxtzpositional phase. This phase uses cut and splice operator. This phase allow to survive highly fit individuals for reproduction.

1.2. Support vector Machine

SVM (Vapnic, 1995) classifies feature input by transforming it into the high dimension space to obtain the best hyperplane that differentiate one class to other classes. A hybrid approach of Messy GA-SVM will be a good approach for feature selection and for improving classification accuracy (Cheng and Wu, 2008). Here it is being used for fitness evaluation based on maximum with winner takes all the heuristic.

1.3. Artificial Neural Network

Neural networks are able to estimate the posterior probability, which provides the basis for establishing classification rule and performing statistical analysis (Zhang, 2000; Jha 2010). In proposed framework Radial basis function neural network is used for optimization. The basic idea of using Radial basis function neural network over the other is that it provides high approximation **(**Feilat, & Ma'aita, 2012).

Facial Expression Databases description

The well known Jaffe database consisting of Japanese Female Expression used for various studies related to emotion recognition (Lyons, M. J. et al., 1998). Similarly Cohn-Kanade database also commands wide acceptance (Lucey et al., 2010). We have taken only 189 images of 7 seven female subject of CK+ to check the accuracy of our work. Besides one more data set containing 147 images of 7 facial expressions was also prepared.

Experimental setup

This section considers the experiment results comparing the (Messy GA, SVM and ANN) approach with validation approach (GA, multi-SVM). Both algorithms are coded with two encoding scheme Prufer and Binary to show the dependency of these algorithms on representation scheme. Tournament selection probability in binary coding is taken 2 and in

prufer coding is 4. Cut and splice probability is set for Messy GA is set to 0.0667 and 1 while mutation probability is set to 0 to observe the effect of recombination alone. Crossover and mutation probability for GA is set to .8 and .01.These parameter selection strategies provide best results presented approach.

Messy GA proceeded over 12 generation for both phase primordial and juxtaposition phase while GA for 10 generation. We considered the population size by default as entire features. Fitness evaluation process uses four option for terminate approaches: maximum number of generations achieved, maximum fitness value desired, threshold for fitness variance below which the population is assumed to be converged and threshold for the unique number of individuals in the converged population.

Radial basis function is set for 500 epochs for both coding schemes. Proper parameters are used to design new RB function to reduce the MSE (mean square error).2/3 and 1/3 ratio of images has been taken for testing and training.

Results

The results are explains in two section. First and second section explain Messy GA results and classification of each database on proposed approach (ensemble of classifiers) with both Binary coding and third section explain the validation results using individual classifier(multi SVM).

Table 1: Messy GA results on all used database

Database	Generation	String length	Chromosome
Jaffe	1	7	(7,0)(8,1)(5,0)(6,0)(3,0)(4,1)(1,1)
	2	5	(1,1)(2,1)(3,1)(4,1)(7,1)
	3	5	(7,1)(8,1)(5,1)(6,1)(1,1)
	4	3	(5,1)(6,1)(7,1)
	5	7	(5,1)(6,1)(7,1)
	6	7	(5,1)(6,1)(7,1)(8,1)(1,1)(2,1)(3,1)
	7	6	(1,1)(2,1)(3,1)(4,1)(5,1)(6,1)
	8	1	(1,1)
	9	3	(1,1)(2,1)(3,1)
	10	3	(1,1)(2,1)(3,1)
	11	2	(1,1)(2,1)
	12	8	(5,1)(6,1)(7,1)(8,1)(1,1)(2,1)(3,1)(4,1)

Kanade	1	4	(1,0)(2,1)(3,0)(4,0)
	2	8	(5,1)(6,1)(7,1)(4,1)(1,1)(2,1)(3,1)(4,1)
	3	7	(5,1)(6,1)(7,1)(8,1)(1,1)(2,1)(3,1)
	4	4	(1,1)(2,1)(3,1)(4,1)
	5	2	(1,1)(2,1)
	6	2	(1,1)(2,1)
	7	6	(5,1)(6,1)(7,1)(8,1)(1,1)(2,1)
	8	7	(1,1)(2,1)(3,1)(4,1)(5,1)(6,1)(7,1)
	9	8	(5,1)(6,1)(7,1)(8,1)(1,1)(2,1)(3,1)(4,1)
	10	7	(5,1)(6,1)(7,1)(8,1)(1,1)(2,1)(3,1)
	11	6	(1,1)(2,1)(3,1)(4,1)(5,1)(6,1)
	12	3	(1,1)(2,1)(3,1)
Real	1	8	(1,0)(2,1)(3,0)(4,0)(5,0)(6,1)(7,1)(8,1)
	2	7	(1,1)(2,1)(3,1)(4,1)(5,1)(6,1)(7,1)
	3	5	(5,1)(6,1)(7,1)(8,1)(1,1)
	4	3	(1,1)(2,1)(3,1)
	5	4	(1.1)(2,1)(3,1)(4,1)
	6	1	(1,1)
	7	8	(1,1)(2,1)(3,1)(4,1)(5,1)(6,1)(7,1)(8,1)
	8	1	(1,1)
	9	2	(1,1)(2,1)
	10	5	(1,1)(1,2)(1,3)(1,4)(1,5)
	11	4	(1,1)(1,2)(1,3)(1,4)
	12	2	(1,1)(2,1)

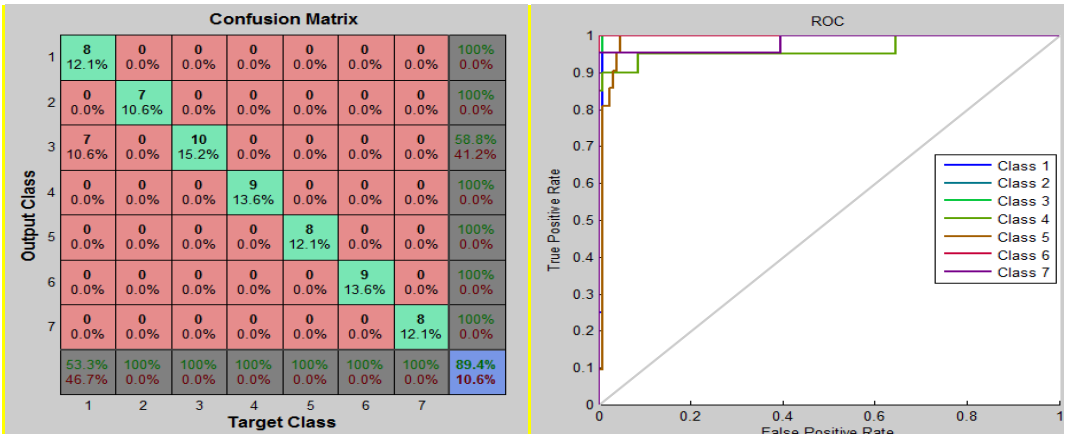


Fig 1: Confusion matrix and ROC curve for Jaffe dataset using Messy GA-SVM-ANN approach for binary coding

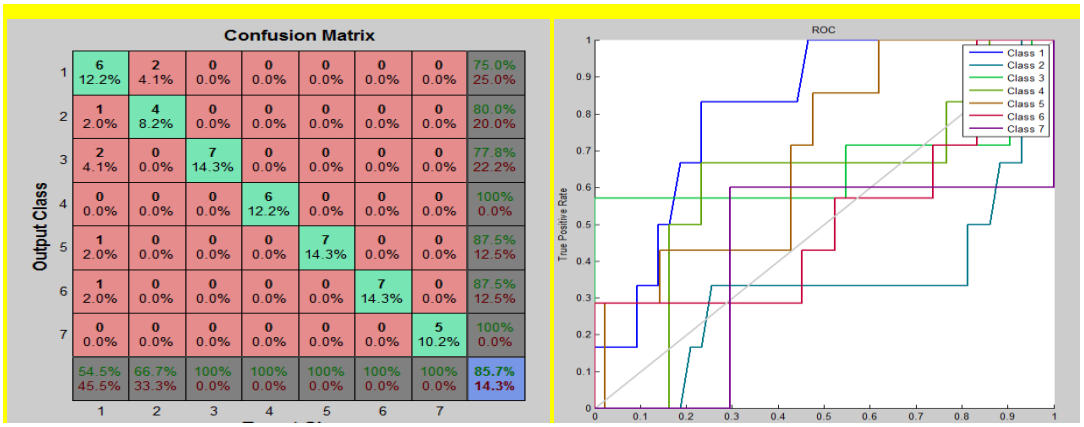


Fig 2 : Confusion matrix and ROC curve for Cohn-Kanade database using Messy GA-SVM-ANN Approach for binary coding

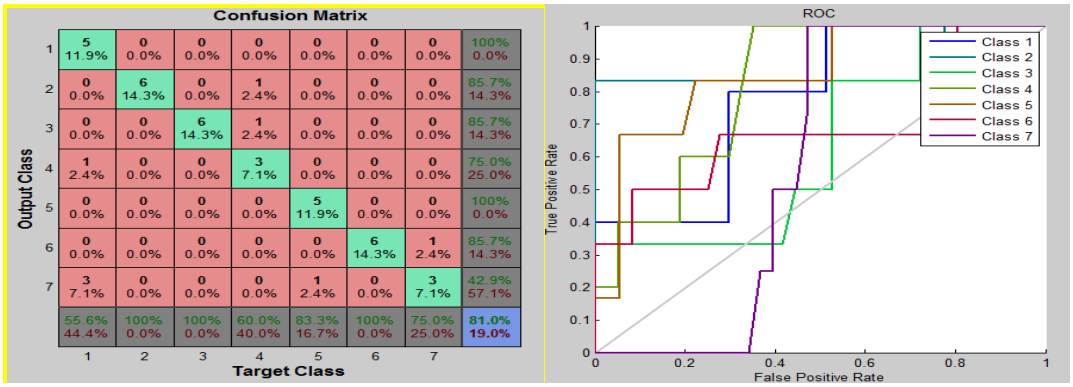


Fig 3: Confusion matrix and ROC curve for Real database using Messy GA-SVM-ANN Approach for binary coding

Comparison with other approach on Jaffe and Cohn-Kanade database

In this research paper, proposed work is also compared with other approaches. The comparison results of all approaches with our proposed approach are depicted in **Table 2.** The reference of all the approaches is mentioned in literature citied.

Table 2: Comparison with other approaches

Databases	Approach	Reference	Accuracy in %
Jaffe	Geometry and Gabor	Zhang *et al.* ,1998	90.1
	Gabor and LVQ	Bashan and Venayagamoorthy, 2008	90.2

	Gabor filters	Koutlas and Fotiadis, 2008	92.3
	Gabor filters	Liu and Wang, 2006	92.5
	2DPCA with feature selection and SVM	Oliveira *et al*., 2011	94.0
	LPB	Liao *et al*., 2006	94.5
	Gaussian process	Cheng *et al*., 2010	95.2
	2D locality preserving projections	Zhi and Ruan, 2008	95.9
Cohn-Kanade	LBP + template matching	Shan *et al*., 2005	79.1
	Gabor filter + SVM	Cohen *et al*. ,2003	73.2
	Generalized Hough transform	Fanelli, *et al*., 2012	86.7
	Gabor filter + SVM	Bartlett *et al*., 2005	89.1
	Messy GA,SVM,ANN	**Proposed approach**	**85.4**

The results explained in **Table 2** depict the comparative analysis of proposed approach (Messy GA-SVM-ANN) with the validation approach (GA-multi SVM) on Binary and Prufer encoding schemes. It can be visualized that accuracy 85.4% has been achieved for the labeling images of Jaffe database. However, overall performance of proposed approach compare well with validation approach. The poverty of the performance in our own data base termed real is associated with the poor expression of the emotion by the volunteers mostly the students of technical stream

Reference

Bartlett, M., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I. and Movellan, J. (2005). Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 568).

Bashan, S. and Venayagamoorthy, G. K. (2008). Recognition of Facial Expressions using Gabor Wavelets and Learning Vector Quantization. Engineering Applications of Artificial Intelligence, 28: 1056–1064. 573.

Cheng, M.Y. and Wu, Y.W. (2008). Dynamic Prediction of Project Success using Evolutionary Support Vector Machine Inference Model. The 25th Symposium on Automata and Robotics in Construction.

Cheng, F., Yu, J. and Xiong, H. (2010). Facial Expression Recognition in Jaffe Dataset Based on Gaussian Process Classification. IEEE Transactions on Neural Networks, 572 21(10), 1685–1690.

Chou, J.S.; Cheng, M.Y; Wu, Y.W. and Pham, A.D. (2014). Optimizing Parameters of Support Vector Machine using Fast Messy Genetic Algorithm for Dispute Classification. Expert Systems with Applications 41: 3955–3964.

Fanelli, G., Yao, A., Noel, P. L., Gall, J. and Van Gool, L. (2012). Hough Forest-based Facial Expression Recognition from Video Sequences. In Trends and Topics in Computer Vision (pp. 195-206). Springer Berlin Heidelberg.

Feilat, E.A. and Ma'aita, E.M. (2012). RBF Neural Network Approach for Identification and Control of DC Motors. TJER, 9(2): 80-89.

Filko, D. and Martinovi, G. (2013). Emotion Recognition System by a Neural Network based Facial Expression Analysis. Online ISSN 1848-3380, Print ISSN 0005-114 ATKAFF 54(2): 263–272.

Fogel, D. (1995). Evolutionary Computation, IEEEPress, NewYork, pp.631-637.

Goldberg, D.E.; Korb, B and Deb, K. (1989). "Messy Genetic Algorithm: Motivation, Analysis and First Result", complex system, 3: 493-530.

Goldberg, D.E.; Korb, B and Deb, K. (1990). "Messy Genetic Algorithm Revisited: Studies in Mixed Size and Scale. Complex System Publication 4: 415-444.

Gunes, T. and Polat, E. (2008). Feature Selection for multi-SVM Classifiers in Facial Expression Classification. Computer and Information Sciences, ISCIS '08. International Symposium.

Hsiao, F. (2012). Neuro-fuzzy Cost Estimation Model Enhanced by Fast Messy Genetic Algorithms for Semiconductor Hookup Construction. Computer-Aided Civil and Infrastructure Engineering. Page 1–18.

Ledezma, A., Aler, R., Sanchis, A., & Borrajo, D. (2010). GA-stacking: Evolutionary stacked generalization. Intelligent Data Analysis, 14(1), 89-119.

Hansen, L.K., and Salamon, P (1990). Neural Network Ensembles.IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(10),993–1001.

Jha, G.K. (2010).Artificial Neural Network and its Application. International Conference on Power Electronics, IICPE.

Koutlas, A. and Fotiadis, D. I. (2008). An Automatic Region based Methodology for Facial Expression Recognition. In IEEE International Conference on Systems, Man and Cybernetics (pp. 662–666).

Li, P.; Phung, S.; Bouzendam, A. and Tivive, F. (2010). Feature Selection for Facial Expression Recognition.2nd Europe workshop on visual information processing (pp.35-39).USA.IEEE.

Liu, W. and Wang, Z. (2006). Facial expression recognition based on fusion of multiple Gabor features. In 18th International Conference on Pattern Recognition 2006 (pp.607 536–539).

Lucey, P; Cohn, J.F; Kanade, T; Saragih, J; Ambadar, Z and Matthews, I. (2010). "The Extended Cohn-Kanade Dataset(CK+):A Complete Database for Action Unit and Emotion-Specified Expression."In computer Vision and Pattern Recognition Workshops (CVPRW). IEEE Computer Society Conf on, IEEE, 94-101.

Lyons, M. J. et al., (1998). Coding Facial Expressions with Gabor Wavelets, 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200-205.

Rahman A and Verma,B.(2013). Ensemble Classifier Generation using Non–uniform Layered Clustering and Genetic Algorithm, Elsevier Knowledge Based Systems, vol. 43, pp. 30–42.

Shan, C., Gong, S. and McOwan, P. (2005). Robust Facial Expression Recognition using Local Binary Patterns. In IEEE International Conference on Image Processing (pp. 370–373).

Sivanandam S.N.& Deep, S.N.(2008).Introduction to Genetic Algorithm. ISBN 978-3-540-73189-4 Springer Berlin Heidelberg New York.

Stringer,H and Annie,S.(2004). Bloat is Unnatural: An Analysis of Changes in Variable Chromosome Length Absent Selection Pressure. UCF Technical Report CS-TR-04-0.

Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. New York: Springer.

Whitley, D; Beveridge, J.R.; Guerra, C. and Graves, C. (1997). "Messy Genetic Algorithm for Subset Feature Selection. Proceedings of the Seventh International Conference on Genetic Algorithms ICGA-97, East Lansing, MI, page 568--575. San Francisco, CA: Morgan Kaufmann.

Zhang, Z., Lyons, M. J., Schuster, M. and Akamatsu, S. (1998). Comparison between Geometry-based and Gabor-wavelets-based Facial Expression Recognition using Multi-layer Perceptron. In Third IEEE international Conference on Automatic Face and Gesture Recognition (pp. 454–459).

Zhang, G.P.(2000).Neural Networks for Classification: A Survey. IEEE Transaction on System, Man and Cybernetics—Part C: Application and Reviews, 30(4).

Zhi, R. and Ruan, Q.(2008). Facial expression recognition based on two-dimensional discriminate locality preserving projections. Neurocomputing, 71: 1730–1734.

Bhupesh Kumar Singh Received his M.Tech Computer Engineering from Indian Institute of Technology, Delhi (India). Pursuing Ph.D. from IIIT Allahabad. His research interest includes Digital Image Processing and Analysis, Machine Learning and Mathematics.

Robust solutions to combinatorial optimization problems under data scenarios

A new transformation technique for arc routing problem into node routing problem on sparse feasible graph

Sara Tfaili[1], Abdelkader Sbihi[2] and Adnan Yassine[1,3]

[1] Université du Havre, Laboratoire de Mathématiques Appliquées du Havre (LMAH), 25 Rue Philippe Lebon, 76600 Le Havre, France
sara.tfaili@etu.univ-lehavre.fr
[2] École de Management de Normandie, Axe Logistique-Terre-Mer-Risque (LTMR), 30 rue de Richelieu, 76600 Le Havre, France
a.sbihi@em-normandie.fr
[3] Institut Supérieur d'Études Logistiques (ISEL), 11 Quai Frissard, 76600 Le Havre, France
adnan.yassine@univ-lehavre.fr

1 Introduction

Arc Routing Problems ([2], [3]) known by their large real applications were not well developed through the research until the last decades. In this paper, we spotlight on this type of routing problems defined over special types of graphs specified by "sparsity", and we introduce a new transformation of the capacitated arc routing problem (CARP) into a capacitated vehicle routing problem (CVRP) that conserves the structure of the underlying graph as well as the structure of the problem itself. Sparse graph feasibility for routing problems have been studied for the first time by Beasley and Christofides [1] in which the underlying graph has some "sparsity". In this paper, we study a special class of arc routing problems that are defined on special particular graphs. We are interested particularly in what is called sparse graphs [6]. Typically, a sparse graph is a graph whose number of arcs is relatively small by comparing it to its number of vertices. One does know that most arc routing problems are defined on sparse graphs. As there is no strict distinction between sparse and dense graphs, we may present here an attempt to define what a sparse graph looks like: Let $G = (V, E)$ be a graph where V is the set of vertices and E is the set of edges. G is said to be a sparse graph if $|E| = O(|V|^{\alpha})$ and $1 < \alpha < 2$. In this study, we intend to transform the arc routing problem into a vehicle routing problem. In fact, there exist different methods of such a transformation, but it never conserves the sparsity of graphs ([4], [5]). However, we introduce a new transformation technique that conserves the sparsity of some particular graphs as well as the structure of the problem provided with some numerical results.

2 Transformations of ARP into VRP through literature

Let $G = (V, E)$ be an undirected graph where V is the set of vertices and E is the set of edges. Each edge is associated with a travel cost $c_e > 0$. Let $R \subset E$ such that $|R| = r$ be the set of the required edges that must be served, and for each $e \in R$, let d_e be the demand of e. Note that each required edges must be served exactly one time by exactly one walk in G. Throughout the following, let $G' = (V', E')$ be the undirected graph that is obtained by any of the following transformations. The following part exposes briefly some previous transformations that were derived before.

2.1 A first transformation attempt

In this transformation, proposed by Pearn et al. in [8], each required edge $e \in E$ is represented in the new graph G' by three nodes in V' in which two are lateral and one is central where d_e is distributed among them such that if $e = \{i, j\}$ is replaced by the three nodes (s_{ij}, m_{ij}, s_{ji}), then

$$d_e = d(s_{ij}) + d(m_{ij}) + d(s_{ji})$$

thus, the complete graph G' in which the VRP is going to be considered has $3r + 1$ nodes. Considering the distribution of the costs in G', it is given as shown below:
$w(0, s_{ij}) = \frac{1}{4}c(i, j) + dist(i, 0)$

$w(s_{ij}, s_{kl}) = \frac{1}{4}(c(i,j) + c(k,l)) + dist(i,k)$ if $\{i,j\} \neq \{k,l\}$
$w(s_{ij}, s_{kl}) = 0$ if $\{i,j\} = \{k,l\}$
$w(m_{ij}, v) = \frac{1}{4}c(i,j)$ if $v = s_{ij}$ or $v = s_{ji}$
$w(m_{ij}, v) = \infty$ otherwise
Note that $dist(i,j)$ denotes the cost of the shortest path between the nodes i and j in G. It is easy to see that for a sparse graph, the structure is not conserved.

2.2 A second transformation attempt

Instead of replacing each required edge in G by three nodes as in the previous transformation by [8], Longo et al. in [5] replaced it by two nodes, and thus G has now $2r+1$ nodes. Let $e = \{i,j\}$ be replaced by the two nodes (s_{ij}, s_{ji}), then
$d(s_{ij}) + d(s_{ji}) = d_e$
For the cost distribution, it is defined in the following:
$w(0, s_{ij}) = dist(0,i)$;
$w(s_{ij}, s_{kl}) = 0$ if $\{i,j\} = \{k,l\}$
$w(s_{ij}, s_{kl}) = c(i,j)$ if $\{i,j\} = \{l,k\}$,
$w(s_{ij}, s_{kl}) = dist(i,k)$ if $\{i,j\} \neq \{k,l\}$ and $\{i,j\} \neq \{l,k\}$.

2.3 A compact transformation

In their paper, Foulds et al. [4], have introduced a new compact transformation of arc routing problem into node routing problem in which the resulting node routing problem instance contains only $r+1$ nodes (r is always the number of the required edges in G). Moreover, for each required edge in G, $e = \{i,j\} \in R$, there exists one single node m_{ij} such that $d(m_{ij}) = d_e$. Considering the problem equivalence and the cost distribution in G and in G', the concept of searching for a shortest path between each two required edges is still present. For the costs in G', a new pair of binary variables was introduced in order to specify the direction of the travel in the graph G', where different cases where studied according to the direction of the service. For more details, one can refer to the paper by [4]. Although this transformation eliminates two out of every three nodes generated by the transformation proposed by [8], but it still considers the obtained VRP on the complete graph formed by the new obtained nodes just like all other previously discussed transformations.

3 A new proposed transformation technique

As we aim at studying the ARP over sparse graphs, and due to the fact that all the VRP are studied over the complete graphs, this means that there is no transformation out of those presented before can be adequate one as their concept of considering the complete graph runs against our concept of sparsity. Thanks to the recent studies that deals with the sparse VRP, this motivates us to look for a new transformation that ensures the sparsity conservation of the graph without affecting the structure of the problem. Our transformation is as follows: an ARP in graph $G = (V,E)$ is transformed into an equivalent VRP in a graph $G' = (V', E')$ where G' is just the line graph of G; $G' = L(G)$. Every edge $e = \{i,j\} \in E$ is represented by one single $m_{ij} \in G'$ in G' (not only the required edges are transformed into nodes, but also the non-required ones i.e. all the nodes). Hence, the resulting problem is defined now on $L(G) = (V', E')$ in which $V' = \bigcup_{(i,j)\in E} \{m_{ij}\}$.

In the follwing, we give an example and give some initial results for the proposed transformation.

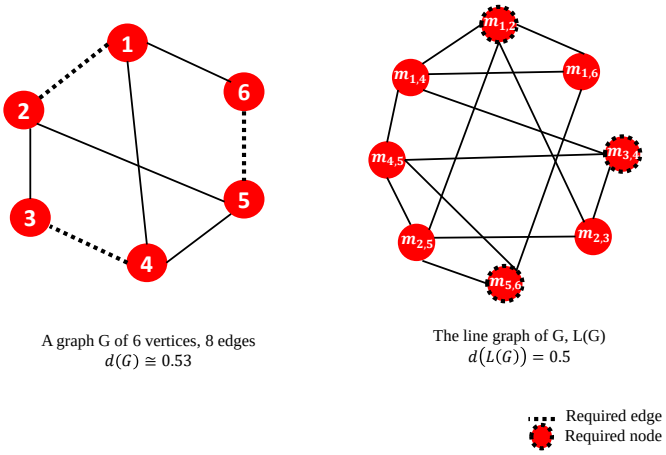


Fig. 1. A graph G and its line graph $L(G)$

# of vertices in G	# of edges in G	Density of G, d(G)	# of vertices in L(G)	# of edges in L(G)	Density of L(G), d(L(G))
13	13	≅ 0.17	13	14	≅ 0.18
14	14	≅ 0.15	14	15	≅ 0.16
15	15	≅ 0.14	15	16	≅ 0.15
16	16	≅ 0.13	16	17	≅ 0.14
17	17	0.125	17	18	≅ 0.132

Number of vertices in G	Number of edges in G	Density of G	Density of L(G)
20	30	0,158	0,137
30	45	0,1	0,09
50	75	0,06	0,054
100	150	0,03	0,0268
200	300	0,015	0,013

Number of vertices in G	Number of edges in G	Density of G	Density of L(G)
21	30	0,143	0,140
31	45	0,1	0,092
51	75	0,058	0,054
101	150	0,029	0,027
201	300	0,0149	0,0134

Fig. 2. Conservation results of the transformation

Motivations of choosing the line graph

- The structure of the connected graph can be recovered completely from its line graph.
- The incident edges in G are equivalent to adjacent vertices in $L(G)$.
- The edges of G are equivalent to the vertices in L(G).
- Line graphs can be recognized in linear time.

These results allow us to re-formulate the original problem as well as to develop new metaheursitics over our assumptions concerning the nature of the graphs and their special sparse structure with a clear study about the effects of different parameters included in the problem. our main contribution is that we consider, and for the first time, the ARP on its original sparse graph. This consideration is assured by our new transformation that conserves the sparsity of the original graph and the structure of the problem. Theoretical results with some numerical tests will be also presented. On the other hand, an adequate robust optimization will be applied after introducing a set of scenarios on the costs. Thanks to Oukil [7] who has considered the VRP on its original sparse graph allowing us to take an advantage to apply to our problem.

References

1. J. Beasley and N. Christofides. Vehicle routing with a sparse feasibility graph. European Journal of Operational Research, 98(3):499-511, May 1997.
2. H. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems, part 1: The chinese postman problem. Operations Research, 43(2):231-242, 1995.
3. H. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems, part 2: The rural postman problem. Operations Research, 43(3):399-414, 1995.
4. L. Foulds, L. Humberto, and J. Martins. A compact transformation of arc routing problems into node routing routing problems. Annals of Operations Research, 226(1):177-200, March 2015.
5. H. Longo, M. P. de Aragao, and E. Uchoa. Solving capacitated arc routing problems using a transformation to the cvrp. Computers and Operations Research, 33(6):1823-1837, June 2006.
6. J. Nešetřil and P. O. de Mendez. Sparsity: Graphs, Structures, and Algorithms. Springer Berlin Heidelberg, 2012.
7. Oukil, A. Exploiting Sparsity in Vehicle Routing Algorithms, PhD Thesis, Lancaster University, UK, 2008.
8. W. Pearn, A. Assad, and B. Golden. Transforming arc routing into node routing problems. Computers and Operations Research, 14(4):285-288, October 1987.

A cooperative multi-agent framework for designing DisFCSPs and their resolution through metaheuristics

Naira LOUIZI
COSMOS Laboratory
National School of Computer Sciences, university of Manouba
Tunisia

Sadok BOUAMAMA
COSMOS Laboratory
National School of Computer Sciences, university of Manouba
Tunisia
University of Jeddah
Saudi Arabia

***Abstract*— In this paper, we introduce a new multi-agent framework conceived for designing Distributed Fuzzy Constraint Satisfaction Problems (DisFCSPs), and their resolution via population based metaheuristics. Different types of agents (interface agents, mediator agent, proposer agents and negotiator agents) are interacting in a cooperative architecture to receive, to read and to deliver problem information, to coordinate activities, to build promising solutions and to generate improved ones, performing a metaheuristic process as a collaborative behaviour. Genetic Algorithm (GA) is able to exploit the architecture and can be easily described in our framework.**

Keywords*— combinatorial optimization; Distributed Fuzzy Constraint Satisfaction Problems; Multi Agent Systems; metaheuristics; Genetic Algorithm.

I. INTRODUCTION

Constraint Satisfaction Problem (CSP) [15] is a formalism proposed by the Artificial Intelligence to provide a generic representation framework to simply formulate many decision problems. A variety of optimization techniques for these problems are available. The CSP formalism is defined informally by a set of variables, subject to a number of constraints that limit the set of values to be allowed to the variables. Constraints in CSPs are called "crisp", "boolean", "yes or no" , or "hard" constraints that can be either satisfied or not, without intermediate state. A solution of a CSP must satisfy all the constraints of the problem.

The approach of rigid constraints, modeled by the CSP formalism, has usage limitations in designing and solving some practical optimization problems. Indeed, some problems are characterized by the flexibility of knowledge they incorporate. Fuzzy Constraint Satisfaction Problem (FCSP) formalism is then appeared in 1993 [3], as a formal framework of processing flexibility for constraint satisfaction problems, and we mean to say that in the same manner as the Fuzzy logic is an extension of Classical Logic, FCSP is an extension of CSP.

On an other hand, there exist problems with information that cannot be centralized, because the knowledge about the problem, that is, variables and constraints, may be logically or geographically distributed among physical distributed agents. Since CSP is the formalism designed essentially to solve problems arising from Artificial Intelligence in a centralized way, it can be extended to operate with several domains of the Distributed Artificial Intelligence. A Distributed Constraint Satisfaction Problem (DisCSP) [18] is a CSP which variables and constraints are distributed among automated agents. Agents are cooperating to find a consistent consensus that satisfies inter-agent constraints.

DisCSPs consider constraints that can be precisely defined and fully satisfied. But real life problems, like scheduling problems [17], [11], timetable scheduling problems [2], planning problems [17], E-Business problems [10], E-Commerce problems [7], opponent's belief problems [12], are inherently both fuzzy and distributed, where preferences and constraints are imprecise. When applied to real life problems, DisCSPs have a limited applicability. Numerous scientific studies [9], [7], [16] have proved that the negotiation approach and the fuzzy logic area, which have been for a long time exploited individually, have to be combined if we want to achieve better results in terms of designing and solving real life problems.

Fuzzy constraints allow to deal with situations when constraints are soft, prioritized and/or imprecisely defined. This can be explored by the MAS (Multi Agent Systems) community by incorporating fuzzy constraints in DisCSPs. Hence, the Distributed Fuzzy Constraint Satisfaction Problem (DisFCSP) formalism [6] is emerging. DisFCSP extends both DisCSP and FCSP. It can be considered as either a version of DisCSP where every constraint is fuzzy, or an extension of FCSP in a distributed environment.

In this paper, we provide a cooperative multi-agent framework, where different types of agents (interface agents, mediator agent, proposer agents and negotiator agents) are interacting in a cooperative architecture to design and solve DisFCSPs. The rest of this paper is structured as follows: In section 2, we provide related work. Section 3 summarizes the proposed architecture, the underlying communication protocol, and studies their properties. Section 4 shows experimental results. In the last section, we provide concluding comments.

II. LITERATURE REVIEW

DisFCSPs form real life problems better by allowing the incorporation of elements of fuzzy set theory to the DisCSPs formalism. It exists several tools that permit to perform the design and the resolution of DisFCSPs. Some of them are highlighted below.

The FeNAs (Fuzzy e-Negotiation Agents) approach [8] employs fuzzy constraints to express preferences, constraints and each party's objectives over problems' issues. Fuzzy constraint-based reasoning and utility theory permit to find solutions that maximizes the satisfaction of all constraints of the parties.

In the DisPFCSPs (Distributed Prioritized Fuzzy Constraint Satisfaction Problems) approach [14], the notion of Prioritized Fuzzy Constraint Satisfaction Problems (PFCSPs) is chosen as the basis of the negotiation model, and it enables negotiation to be carried out over fuzzy constraints of multiple issues of a product. The model, also, incorporates the concept of a reward, from argumentation / persuasion-based models.

Both of the FeNAs and the DisPFCSPs approaches deal with multi-issue bilateral negotiation in competitive or semi-competitive environments.

The FCNs (Fuzzy Constraint Networks) approach [13] advances the state of the art in the way that it deals with multi-issue multilateral negotiation, using fuzzy constraints on a combination of multiple attributes. Agents can reach the deal that is close to the overall satisfaction degree of the constraints.

We observe that aforementioned works focus only on negotiation in trading environments, thus, agents may be seen as software entities which encapsulate distributed functional entities (system's users, consumers, clients in trading environment). Agents in our framework are problem independent, in such a way that they represent abstract interacting modeling and solving entities of DisFCSPs. Interactions between those entities form a metaheuristic as a cooperative behaviour.

By surveying the literature, we found many CSPs solving approaches that have been extended to solve DisCSPs or FCSPs separately. But to our knowledge, it do not exist such an initiative which investigate the power of metaheuristics to solve DisFCSPs. One of our objectives is to cover this lack in the literature, by presenting a generic and uniform framework adapted to DisFCSPs, and to develop a family of metaheuristics that can be embedded in this framework.

To resume, our contribution differentiate from the other ones in the literature in three points:

- generic framework for designing and resolution of DisFCSPs
- in this framework, classical metaheuristics can be accommodated and extended
- collaborative environment (not competitive).

III. A MULTI-AGENT COOPERATIVE FRAMEWORK FOR DISFCSPS

The main intent of our framework is to develop a generic and uniform methodology that allows to formally design DisFCSPs and use metaheuristics as their solving techniques.

The framework follows the principle of separation between the different activities in metaheuristics' process, namely the preprocessing activity, the initialization activity, the processing activity, and the coordination activity, which respectively reflect the corresponding main implementation steps for any metaheuristic. Thus, the framework is based on a cooperative architecture which involves four cooperative agents' types: interface agents, proposer agents, negotiator agents and mediator agent, and interactions between them. We note that the architecture allows exchanging informations between agents belonging to the same type or to different types in order to achieve problems designing and metaheuristic activities.

A. Preprocessing

Depending on the treated problem and related to its real context, the preprocessing phase may include:

- decision of the problem input data representation by an appropriate data structure;
- decision of solutions coding type;
- definition of the objective function;

- parameters' adjustment and strategies' choices.

Interface agent: An interface agent assists the framework user to actively operate on an interactive system interface. Its roles consist of:

- reading inputs that the user gives;
- displaying final solutions to the user;
- delivering informations relevant to the problem issues (preferences, objectives, parameters, strategies) to the mediator agent.

B. Initialization

The second phase consists of creating the initial population by performing a metaheuristic initialization strategy (i.e. like random generation, sequential diversification, parallel diversification or heuristic initialization). The high quality of the initial population accelerates the optimization process and has an influence on the quality of the optimal solutions it produces. To carry out the instantiation activity, agents in this phase deal with original problem issues, represented by variables, domains' variables, and fuzzy constraints.

On a priority basis, the construction of the initial population begins with variables' instantiation of the proposer agent having the higher priority. Afterwards, each agent constructs an instantiation of its variables, taking into account instantiations of preceding agents. In the case that an agent is unable to make assignment to its variables, it sends a message to the mediator agent which informs the last agent in the priority ranking, which must search for a new assignment to its variables. A complete instantiation is made by accumulating values' assignments of all agents. This process is repeated until a number *p*, indicating the required number of initial solutions (population size), is reached.

Proposer agent: The purpose of a proposer agent is to build candidate solutions by generating a set of promising sub-assignments, and to coordinate with the mediator agent to receive problem's input data and transmit the initial population.

Formalism: The input of these agents is the original problem issues. A network of agents support the modeling of the original global problem as a DisFCSP. Proposer agents, so, formulate a distributed network of FCSPs. The original problem is distributed among a number of agents representing the participants, and formulated as a network of FCSPs (each of which handled by a single agent) connected by inter-agents' fuzzy constraints. Each agent handle a sub-problem which is defined as a FCSP. The output of these agents is a population of initial candidate solutions.

The initialization phase is formally described as a set of N A_{prop} agents (proposer agents), representing the participants of the global DisFCSP problem:

$$A_{prop} = \{V, D, C, P, O, A\}$$

with: V: variables , D: domains' variables , C: fuzzy constraints, P: agent' priority, O: objective function, A: action

- $V = V_{intra} \cup V_{inter}$: a finite set of X variables representing the unknowns of the agent n:
  - $V_{intra} = \{\{V^n_x\}/ (n=1,..,N), (x=1,..,X)\}$: the set of x intra-variables of agent n $(n=1,..,N)$ representing its private variables
  - $V_{inter} = \{\{V^n_x\} \cap \{_{m=1,..,(N\setminus\{n\})} V^m_y\}\}$: a finite set of $(x \cap y)$ inter-agents' variables of agent n $(n=1,..,N)$ representing its shared or negotiation variables
- $D = \{\{D^n_x\}/ (n=1,..,N), (x=1,..,X)\}$: a finite and discrete set of private domains for the variables of agent n, where each D^n_x is a set of possible values for the variable x
- $C = C_{intra} \cup C_{inter}$: a finite set of K fuzzy (soft, hard, or prioritized) constraints representing the constraints handled by agent n:
  - $C_{intra} = \{\{C^n_k\}/ (n=1,..,N), (k=1,..,K)\}$: a finite set of k intra-constraints of agent n $(n=1,..,N)$ representing its private constraints
  - $C_{inter} = \{C^n_k\} \cap \{_{m=1,..,(N\setminus\{n\})} C^m_h\}$: a finite set of $(k \cap h)$ inter-constraints of agent n $(n=1,..,N)$ representing its shared or negotiation constraints
- P : the ratio of hard constraints of agent n representing its priority
- $O = max\ [min\ _{i=1,\ ...,\ m}\ \mu_k(V^n_x)]$: objective function of agent n
- A = the action performed by agent n to construct initial assignments

C. Processing

The third phase consists of the optimization algorithm itself that will work on improving the initial population. It supports the modeling of a distributed network of α-cut CSPs, and the improvement of the initial population. Initial solutions have, each of which, a global satisfaction degree, or a fitness value (satisfaction degree of the objective function). For each level α_j / $1 > \alpha_j > 0$, an α_j-CSP is formed by the set of hard constraints $C_i^{\alpha j}$ containing the tuples that satisfy C_i to a degree greater than or equal to α_j. Then, negotiator agents perform a set of steps of their metaheuristic process. Each α_j-CSP agent performs its own improvement

dynamic. Each agent makes a local search in a local space of solutions having the same fitness value, trying to improve the solutions it has received from the proposer agents.

Agents belonging to this phase deal with initial solutions, and are capable of executing computation of improved solutions. A negotiator agent performs its local improvement process on an α-cut CSP. Each agent is responsible of an α-cut CSP, a set of solutions violating constraints in the same level α.

It must be noted that solutions handled by a negotiator agent differentiate by their collective properties, essentially the satisfaction degree of the global objective function.

Negotiator agent: Negotiator agents are improving solutions' agents. The negotiator agent is capable of alternating intensification and diversification to improve initial solutions given by the proposer agents.

Formalism: The input of these agents is the initial population generated by the proposer agents. A network of agents constructs a distributed network of α-cut CSPs. Each agent handle a sub-problem which is defined as an α-cut CSP. The output of these agents is a set of improved solutions.

The processing phase is formally described as a set of N A_{neg} agents, representing the set of α-cut CSPs.

$$A_{neg} = \{V, D, C, O, A\}$$

with: V: variables , D: domains' variables , C: crisp constraints, O: objective function, A: action

- $X = X_{intra} \cup X_{inter}$:a finite set of variables representing the unknowns of the problem:
  - $X_{intra} = \{X^j_n\}$: the finite set of n decision variables of agent j $(j=1,..,p)$
  - $X_{inter} = \cap_{p=1,..,j} X^p_m$: the set of m inter-agents' variables
- $D = \{D^j_i\}/\ i=1,..,n:$ a finite and discrete set of private domains for the variables of each agent j $(j=1,..,p)$, where each D^j_i is a set of possible values for the variable x^j_i
- $C = C_{intra} \cup C_{inter}$: a finite set of crisp constraints representing the constraints of the distributed problem:
  - $C_{intra} = \{C^j_k\}$: the finite set of k crisp constraints of agent j $(j=1,..,p)$
  - $C_{inter} = \cap_{p=1,..,j} C^p_m$: the set of m inter-agents' crisp constraints
- $O = max\ [min_{\ i=1,\ ...,\ m}\ \mu_k (V^n_x)]$: objective function of agent n
- A = the action performed by agent n to improve initial solutions

D. Coordination

The purpose of the coordination phase is to retrieve and filter information received from interface agents, and to coordinate the activities of the other agents (proposer and negotiator agents).

Mediator agent: Actions performed by the mediator agent can be resumed in:

- attributing priority values to proposer agents;
- calculating the global satisfaction degree (fitness value) of each complete instantiation and sends it to negotiator agents;
- assigning each initial solution to an α-cut agent according to the global satisfaction degree;
- dynamically creating new negotiator agents only if necessary.

The details related to the framework architecture and the negotiation protocol are presented in Fig. 1.

IV. EXPERIMENTS

In this section, we present experimental results of the application of the genetic algorithm, which is adapted to our multi-agent cooperative framework, and implemented in Java using the multi-agent platform Jade. The experiments are performed on an Intel core i7-4500U (2.4 Ghz).

Genetic algorithm is a randomized population-based metaheuristic which is acknowledged as good solver for hard problems. Genetic algorithms have been developed by J. Holland in the 1970s to understand the adaptive processes of natural systems. Then they have been applied to optimization and machine learning in the 1980s [4]. The goal of our experimentation is to compare an implementation of standard genetic algorithm for DisFCSPs with a an another one for FCSPs. The genetic process for DisFCSPs is detailed in Fig. 2.

The objective function is to maximize f(x), such that:

$$max\ f(x) = max\ (min_{i=1,...,\ m}\ \mu_{ci}(x)) \qquad (1)$$

$_{i=1, \ldots, m}$ $\mu_{ci}(x)$ is the global satisfaction degree of the fuzzy constraints with the instantiations. It is influenced by the degree of satisfaction of each fuzzy constraint and is obtained as aggregation by the min operator [5].

Concerning the GA parameters, all the experimentations employ the integer chromosome representation, a size of initial population equal to 100, a cross-over probability equal to 0,5, and a mutation probability equal to 0,015. Due to the stochastic aspect of genetic algorithms, we have performed 10 experimentations per example of random generated problems, and then we take the average without considering outliers. For each combination density-tightness, we also take the average of the 30 generated examples.

To validate our approach, we choose to test it on randomly generated binary problems with fuzzy constraints and various sets of parameters settings.

A class of random binary FCSPs [1] can be specified by mean of four parameters <*n*, *d*, *c*, *t*> where:

- *n* is the number of variables,
- *d* is the common size of the domains $D_1, \ldots, D_n$,
- *c* is the density of the constraint network,
- *t* is the average tightness of constraints.

A class of random binary DisFCSPs can be specified by mean of five parameters <*m*, *n*, *d*, *c*, *t*> where:

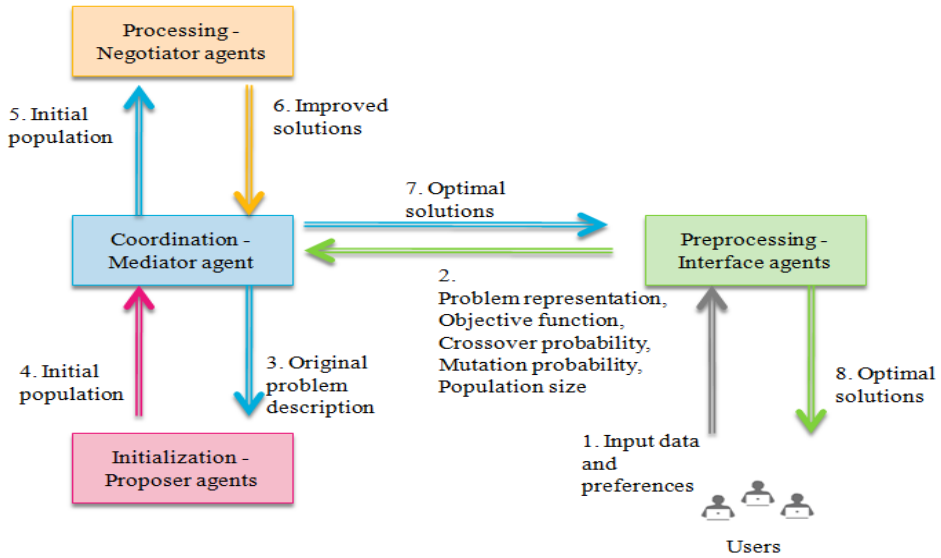
- *m* is the number of agents,
- *n* is the number of variables,
- *d* is the common size of the domains $D_1, \ldots, D_n$,
- *c* is the density of the constraint network,
- *t* is the average tightness of constraints.

We use the following numerical values for all upcoming experiments: agents number = 10, variables number = 50, maximal domain size of each variable = 50. For the connectivity and the tightness we choose the following values: 0.1, 0.3, 0.5, 0.7 and 0.9. So we obtain 25 density-tightness combinations. For each combination, 30 random examples are generated, which gives the total of 750 examples. For each combination density-tightness, we take the average of the 30 generated examples without considering outliers.

In order to make a quick and concise comparison between the executions of the two approaches, we performed a set of experiments with the same GA on random problems and we compute satisfaction ratios of A1 approach (GA for FCSPs) and A2 approach (GA for DisFCSPs) as follows:

Satisfaction-ratio = A2-Satisfaction/A1-Satisfaction.

Fig. 1. The dynamic of the cooperative multi-agent framework



We are going to evaluate the performances according to the global degree of satisfaction. Each time, the GA was able to find a solution of the considered instance after around 8 to 12 generations. The results are presented in TABLE 1, where any number greater than 1 indicates superior performance by *A2*.

V. CONCLUSION

In this article, we have discussed a multi-agent cooperative approach to metaheuristics. We have described the multi-level architecture, the global dynamic of the framework and its properties, and we have tested the genetic algorithm by specializing the meta-architecture.

Further works could be considered by using our proposed approach, to describe other metaheuristics and to solve other DisFCSPs. No doubt further refinements of our approach would allow their performances to be improved.

Fig. 2. The genetic process for DisFCSPs

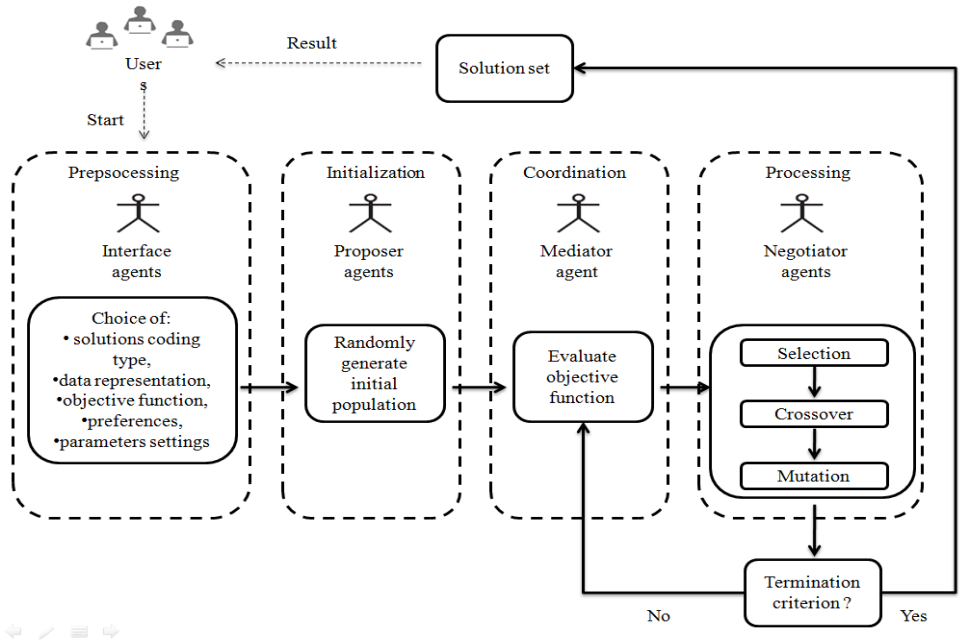


TABLE I. THE SATISFACTION RATIO

		Tightness				
		0.1	**0.3**	**0.5**	**0.7**	**0.9**
Density	**0.1**	1.34	1.37	1.27	1.3	1.31
	0.3	1.38	1.2	1.22	1.37	1.35
	0.5	1.19	1.26	1.35	1.29	1.3
	0.7	1.3	1.31	1.31	1.27	1.29
	0.9	1.38	1.35	1.34	1.34	1.23

REFERENCES

[1] J. Bowen, and G. V. Dozier, "Solving randomly generated fuzzy constraint networks using evolutionary/systematic hill-climbing," In IEEE, 1996.

[2] C. K. Chau, and K. M. Sim, "Engineering fuzzy constraint satisfaction agents for multi-user timetable scheduling," In International Computer Science Conference on Active Media Technology (pp. 244-254). Springer Berlin Heidelberg, December 2001.

[3] Dubois, H. Fargier and H. Prade, "The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction," [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems, 1993.

[4] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," U Michigan Press, 1975.

[5] R. Kowalczyk, "On solving fuzzy constraint satisfaction problems with genetic algorithms," In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on (pp. 758-762). IEEE, May 1998.

[6] R. Kowalczyk, "On negotiation as a distributed fuzzy constraint satisfaction problem," In Proceedings of the Third International Symposium on Soft Computing for Industry of the World Automation Congress (pp. 631-637), 2000.

[7] R. Kowalczyk and V. Bui, "On Fuzzy e-Negotiation Agents: Autonomous negotiation with incomplete and imprecise information," In Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on (pp. 1034-1038). IEEE, 2000.

[8] R. Kowalczyk and V. A. Bui, "FeNAs: A fuzzy e-negotiation agents system," In Computational Intelligence for Financial Engineering, 2000.(CIFEr) Proceedings of the IEEE/IAFE/INFORMS 2000 Conference on (pp. 26-29). IEEE, 2000.

[9] R. Lai and M. W. Lin, "Agent negotiation as fuzzy constraint processing," In Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on (Vol. 2, pp. 1021-1026). IEEE, 2002.

[10] K. R. Lai and M. W. Lin, "Modeling Agent Negotiation Via Fuzzy Constraints in E-Business," Computational Intelligence, 20(4), 624-642, 2004.

[11] K. R. Lai, M. W. Lin and B. R. Kao, "Modeling distributed scheduling via fuzzy constraint-based agent negotiation," In Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on (pp. 30-30). IEEE, June 2007.

[12] K. R., Lai, M. W. Lin and T. J. Yu, "Learning opponent's beliefs via fuzzy constraint-directed approach to make effective agent negotiation," Applied Intelligence, 33(2), 232-246, 2010.

[13] M. W. Lin, K. R. Lai and T. J. Yu, "Fuzzy constraint-based agent negotiation," Journal of Computer Science and Technology, 20(3), 319-330, 2005.

[14] X. Luo, N. R. Jennings, N. Shadbolt, H. F. Leung and J. H. M. Lee, "A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments," Artificial Intelligence, 148(1), 53-102, 2003.

[15] U. Montanari, "Networks of constraints: Fundamental properties and applications to picture processing," *Information sciences*, *7*, 95-132, 1974.

[16] X. T. Nguyen and R. Kowalczyk, "On solving distributed fuzzy constraint satisfaction problems with agents," In Intelligent Agent Technology, 2007. IAT'07. IEEE/WIC/ACM International Conference on (pp. 387-390). IEEE, November 2007.

[17] W. Shen, L. Wang and Q. Hao, "Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 36(4), 563-577, 2006.

[18] M. Yokoo, E. H. Durfee, T. Ishida and K. Kuwabara, "The distributed constraint satisfaction problem: Formalization and algorithms," IEEE Transactions on knowledge and data engineering, 10(5), 673-685, 1998.

A Robust Heuristic for Diffusion Source Localization in Complex Networks

M. LALOU[1,2] and H. KHEDDOUCI[2]

[1] Département d'Informatique, Faculté des Sciences Exactes,
Université A. Mira Béjaia, 06000 Béjaia, Algérie
mohammed.lalou@gmail.com
[2] Université Claude Bernard Lyon 1 - UFR-Informatique, Lab LIRIS
43 bd du 11 Novembre 1918, F-69622 Villeurbanne Cedex, France
{mohammed.lalou, hamamache.kheddouci}@univ-lyon1.fr

1 Introduction

Diffusion in networks is one of the most studied issues in complex networks analysis. Propagation of infectious diseases and rumors in social networks, diffusion of virus and misinformation in computer networks, spreading of ideas in communities and contaminations in water distribution networks are examples of diffusion in complex networks. This issue has been considered for different objectives, including: inferring the underlying diffusion network [2, 3], maximizing the spread of influence [4, 6], detecting the diffusion outbreaks [7, 11], blocking the contagion diffusion [8, 14, 9] and locating the diffusion sources [15, 13, 19]. The last two ones are the most studied.

Blocking diffusion is often considered in the case of harmful entities spreading, such as: diseases, extremist ideology, virus, rumor and leaked information, etc. in order to stop the diffusion and limit any possible risks. However, in most circumstances, the purpose is not only to block the diffusion, but also to identify its source. In the work-in-progress, we are mainly interested in developing an efficient algorithm for diffusion source localization.

Localizing the source of diffusion has recently received much attention. It has many applications in several areas, such as identifying the culprit by the authorities [15], determining the patient-zero of a pandemic [13], disclosing the person who started a rumor (in a social network) [17], finding the administrator of a cyber-attack [18], etc.

An intuitive solution to localize the source of diffusion is to observe all nodes states, however, this is unfeasible since almost all networks of interest are very large, also controlling nodes has usually a cost, and then observing all nodes is not cost-efficient. So, we consider the problem of localizing the diffusion source under the constraint that only a subset of nodes can be observed. We are therefore looking for, first optimally selecting a subset of nodes whose controlling leads to a good observability of the network, and second, an efficient method to accurately localize the diffusion source using only the partial information gathered at the observed nodes (called *observers* [12]). In doing so, we have developed, in a recent work [1] (submitted to ASONAM 16), an efficient heuristic to accurately estimates both the source and the start time of a diffusion, with limited observations. The heuristic takes advantage of the network structural properties, by selecting the critical nodes [10] as observers. The critical nodes are those whose deletion disconnects the network on connected components of at most a given number of nodes L. Observing these nodes ensures that each diffusion that spreads in the network is observed by at least one observer after at most L hops, and allows also to easily locate the part of the network where the outbreak occurred, since the critical nodes are the connection between the components and any path between two component pass through them.

For estimating the diffusion source, our proposed heuristic estimates both the source and the start time of a diffusion in the one-source diffusion case. It is based on the fundamental property of diffusion in complex networks showed by Brockman (2013) [5], where the relative infection time and the effective distance [5] of a node are linearly correlated:

$$t_u = \alpha \cdot D_{su} + c$$

where t_u is the relative infection time of node u, D_{su} the effective distance from s to u.

Based on this fundamental property, we estimate the real diffusion source using the well-known *Ordinary Least Squares method* [16]. Given a linear regression, *OLS* attempts to model the relationship (by fitting a linear equation to observed data) between two random variables X and Y, where X is the independent variable and Y is the dependent variable. In the case of a diffusion, the independent variable X is the effective distance, and the dependent variable Y is the infection time. Thus, the node with the high value of the correlation coefficient or the less value of the residual sum is the most likely to be the source.

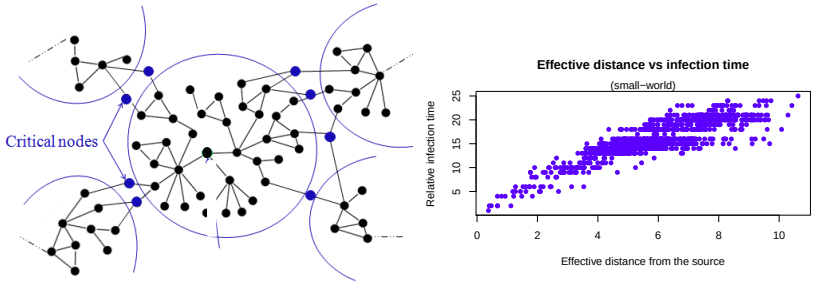


Fig. 1. (left) Critical nodes as observers where the infected part of the network is easily located (right) Correlation between the relative infection time of a node and its effective distance from the source for a small-world network of 1000 nodes.

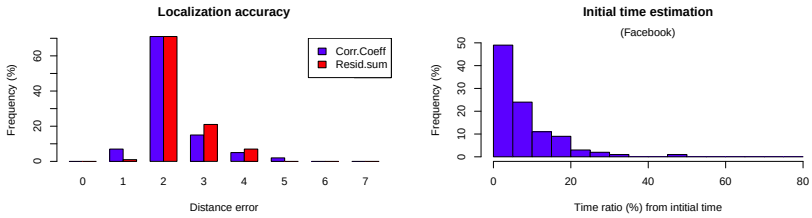


Fig. 2. (left) The diffusion source estimation in term of the number of hops from the real source using both the correlation coefficient and the residuals sum for a part of the Facebook network of 4039 nodes and 88 236 edges (right) the estimation of the diffusion start time for the same network

Although the proposed heuristic achieves a good record on estimating the diffusion source, it has an important drawback which is the sensibility to outliers (extreme observations). In *Figure 3*, we can notice that some bins (of distance error 10 and 11) lie far away from the distance error values center, this is due to the outliers. So, the *OLS* methods are often unstable on the presence of extreme observations, and to deal with that, we propose to explore other advanced techniques, such as *Robust Regression*, which are less sensitive to the effect of outliers.

References

1. M. Lalou, H. Kheddouci, et al. Least Squares Method for Diffusion Source Localization in Complex Networks. submitted to ASONAMÍ6 (2016).
2. E. Adar, L. Adamic, et al. Tracking information epidemics in blogspace. In *Web intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM international conference on*, pages 207–214. IEEE, 2005.
3. M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1019–1028. ACM, 2010.
4. D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

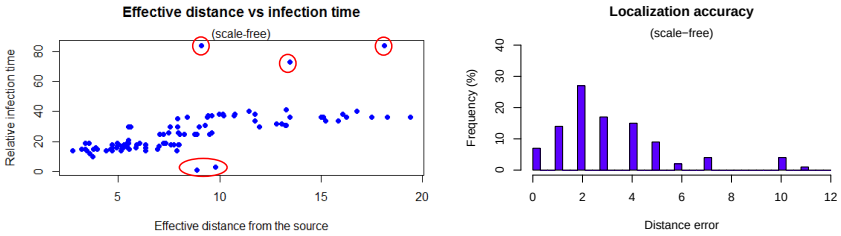


Fig. 3. (left) the relative infection time Vs the effective distance from the source for a of a scale-free network of 500 nodes, the outliers are surrounded (right) the impact of the outliers (in term of number of hops from the real source) on the source estimation of a scale-free network of 1000 nodes.

5. D. Brockmann and D. Helbing. The hidden geometry of complex, network-driven contagion phenomena. *Science*, 342(6164):1337–1342, 2013.
6. W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.
7. N. A. Christakis and J. H. Fowler. Social network sensors for early detection of contagious outbreaks. *PloS one*, 5(9):e12948, 2010.
8. C. J. Kuhlman, V. A. Kumar, M. V. Marathe, S. Ravi, and D. J. Rosenkrantz. Finding critical nodes for inhibiting diffusion of complex contagions in social networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 111–127. Springer, 2010.
9. C. J. Kuhlman, G. Tuli, S. Swarup, M. V. Marathe, and S. Ravi. Blocking simple and complex contagion by edge removal. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 399–408. IEEE, 2013.
10. M. Lalou, M. Tahraoui, and H. Kheddouci. Component-cardinality-constrained critical node problem in graphs. *Discrete Applied Mathematics*, 2015.
11. J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.
12. Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.
13. A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Physical Review E*, 90(1):012801, 2014.
14. R. Pastor-Satorras and A. Vespignani. Immunization of complex networks. *Physical Review E*, 65(3):036104, 2002.
15. P. C. Pinto, P. Thiran, and M. Vetterli. Locating the source of diffusion in large-scale networks. *Physical review letters*, 109(6):068702, 2012.
16. C. R. Rao and H. Toutenburg. *Linear models*. Springer, 1995.
17. E. Seo, P. Mohapatra, and T. Abdelzaher. Identifying rumors and their sources in social networks. In *SPIE defense, security, and sensing*, pages 83891I–83891I. International Society for Optics and Photonics, 2012.
18. D. Shah and T. Zaman. Detecting sources of computer viruses in networks: theory and experiment. In *ACM SIGMETRICS Performance Evaluation Review*, volume 38, pages 203–214. ACM, 2010.
19. D. Shah and T. Zaman. Rumors in a network: Who's the culprit? *Information Theory, IEEE Transactions on*, 57(8):5163–5181, 2011.

Hybrid Optimization Architecture Using Hopfield Recurrent Network and Genetic Algorithm Applied to Dynamic Programming Problems

[1] Ivan Nunes da Silva, [2] Matheus Giovanni Pires

[1] Department of Electrical and Computer Engineering, University of São Paulo,
USP/EESC/SEL, CP 359, 13566-590, São Carlos, Brazil
{insilva@sc.usp.br}
[2] Department of Exact Sciences, State University of Feira de Santana,
UEFS/DCE, 44036-900, Feira de Santana, Brazil
{mgpires@ecomp.uefs.br}

Abstract. Dynamic programming has provided a powerful approach to solve optimization problems, but its applicability has sometimes been limited because of the high computational effort required by the conventional algorithms. This paper presents an association between Hopfield neural networks, which are computing models capable of solving a large class of optimization problems, with genetic algorithms, which cover extensive search spaces and guarantee the convergence of the system to the equilibrium points that represent feasible solutions for various dynamic programming problems.

Keywords: Dynamic programming, genetic algorithms, Hopfield network.

1 Introduction

The parallel nature of artificial neural networks makes them suitable for solving of several classes of optimization problems. Among the kinds of optimization problems that can be solved by artificial neural networks, we point out: combinatorial optimization [1], linear programming [2], constrained nonlinear programming [3], unconstrained optimization [4] and dynamic programming problems [5].

Usually, the solving of optimization problems by dynamic programming involves the recurrence relations developed by Richard Bellman [6]. Although the dynamic programming has been used for solving several classes of optimization problems, it has computational inefficiency (e.g. CPU time, memory), so the neural networks become an alternative approach that can be applied in these problems.

Another approach that has been applied to optimization problems and has shown promise for solving such problems efficiently is Genetic Algorithm (GA). In this paper, we perform an analysis of a neurogenetic architecture, not depending on weighting and/or penalty parameters, for dynamic programming.

2 Shortest Path Problem

A typical dynamic programming problem can be modeled as a set of source and destination nodes with n intermediate stages, m states in each stage, and metric data

$d_{xi,(x+1)j}$, where x is the index of the stages, and i and j are the indices of the states in each stage, as shown in Fig. 1.

The goal of the dynamic programming considered here is to find a valid path, which starts at the source node, visits one and only one state node in each stage, reaches the destination node, and has a minimum total length (cost) among all possible paths.

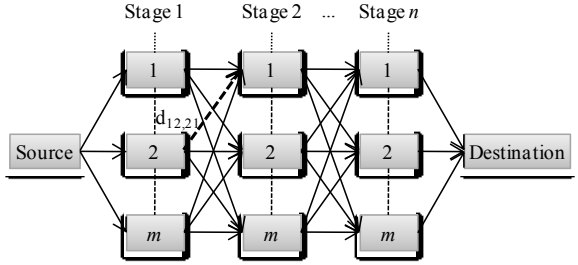


Fig. 1. Shortest path problem.

3 The NeuroGenetic Approach (NGA)

As introduced in [1], the node equation for the continuous-time network with N-neurons is given by:

$$\dot{u}_i(t)=\sum_{j=1}^{N} T_{ij}.v_j(t)+i_i^b \quad \textbf{(1)}$$

$$v_i(t)=g(u_i(t)) \quad \textbf{(2)}$$

where: $u_i(t)$ is the current state of the i-th neuron; $v_j(t)$ is the output of the j-th neuron; i_i^b is the offset bias of the i-th neuron; $\eta.u_i(t)$ is a passive decay term; and T_{ij} is the weight connecting the j-th neuron to i-th neuron. It is shown in [1] that the network equilibrium points correspond to values $v(t)$ for which the energy function (3) associated with the network is minimized:

$$E(t)=-\frac{1}{2}v(t)^T.T.v(t)-v(t)^T.i^b \quad \textbf{(3)}$$

The mapping of optimization problems using the Hopfield neural network consists of determining the weight matrix T and the bias vector i^b to compute equilibrium points. A modified energy function $E^m(t)$ is used here, which is defined by:

$$E^m(t)=E^{conf}(t)+E^{op}(t) \quad \textbf{(4)}$$

where $E^{conf}(t)$ is a confinement term that groups all the structural constraints associated with the problem, and $E^{op}(t)$ is an optimization term that conducts the network output to the equilibrium points corresponding to a cost constraint. Thus, the minimization of $E^m(t)$ of the modified Hopfield network is conducted in two stages:

i) Minimization of the term $E^{conf}(t)$:

$$E^{conf}(t)=-\frac{1}{2}v(t)^T.T^{conf}.v(t)-v(t)^T.i^{conf} \quad \textbf{(5)}$$

where: $v(t)$ is the network output, T^{conf} is weight matrix and i^{conf} is bias vector belonging to E^{conf}. This corresponds to confinement of $v(t)$ into a valid subspace generated from structural constraints imposed by the problem. An investigation associating the equilibrium points with respect to the eigenvalues and eigenvectors of

the matrix T^{conf} shows that all feasible solutions can be grouped in a unique subspace of solutions [8]. As consequence of the application of this subspace approach, which is named by valid-subspace method, a unique energy term can be used to represent all constraints associated with the optimization problem.

ii) Minimization of the term $E^{op}(t)$:

$$E^{op}(t)=-\frac{1}{2}v(t)^T.T^{op}.v(t)-v(t)^T.i^{op} \quad \textbf{(6)}$$

After confinement of all feasible solutions to the valid subspace, a GA is applied in order to optimize E^{op} by inserting the values $v(t)$ into the chromosomes population. Thus, the operation of this hybrid system consists of three main steps:

Step (I): Minimization of E^{conf}, corresponding to the projection of $v(t)$ in the valid subspace defined by:

$$v=T^{val}.v+s \quad \textbf{(7)}$$

where: T^{val} is a projection matrix (T^{val} . $T^{val} = T^{val}$) and the vector s is orthogonal to the subspace (T^{val} . $s = 0$). This operation corresponds to an indirect minimization process of $E^{conf}(t)$, i.e., $T^{conf} = T^{val}$ and $i^{conf} = s$.

Step (II): Application of a nonlinear 'symmetric ramp' activation function constraining $v(t)$ in a hypercube:

$$g_i(v_i)=\begin{cases}1, & if \quad v_i>1\\ v_i, & if \quad 0\le v_i\le 1\\ 0, & if \quad v_i<0\end{cases} \quad \textbf{(8)}$$

where $v_i \in [0,1]$.

Step (III): Minimization of E^{op}, which involves the application of a genetic algorithm to move $v(t)$ towards an optimal solution that corresponds to network equilibrium points, which are the solutions for the optimization problem considered.

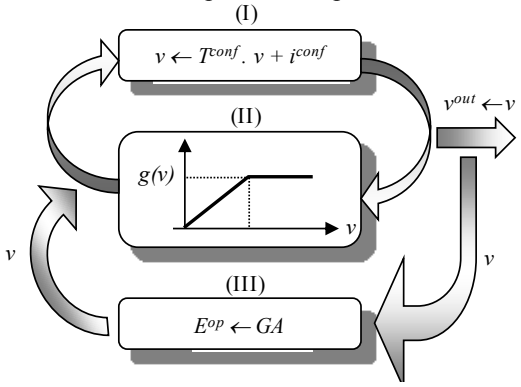


Fig 2. The neurogenetic approach.

As seen in Fig. 2, each iteration represented by the above steps has two distinct stages. First, as described in Step (III), v is updated using the GA. Second, after each updating given in Step (III), v is projected directly in the valid subspace by the modified Hopfield neural network. This second stage is an iterative process, in which v is first orthogonally projected in the valid subspace by applying Step (I) and then

thresholded in Step (II) so that its elements lie in the range defined by [0,1]. This process moves the network output to the equilibrium point that corresponds to the optimal solution for the optimization problem. The convergence process is concluded when the values of v^{out} during two successive loops remain practically constant, where the value of v^{out} in this case is equal to v.

4 Mapping of the Shortest Path Problem

4.1 Notation and Definitions

- The vector $p \in \Re^n$ represents the solution set of an optimization problem consisted of n nodes (neurons). Thus, the elements belonging to p have integer elements defined by:

$$p_i \in \{1,...,n\} \text{ where } i \in \{1..n\} \quad \textbf{(9)}$$

The vector p can be represented by a vector v, composed of ones and zeros, which represents the output of the network. In the notation using Kronecher products [9], we have:

- δ is a matrix ($\delta \in \Re^{nxn}$) defined by:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad \textbf{(10)}$$

$\delta(k) \in \Re^n$ is a column vector corresponding to k-th column of δ.

- $v(p)$ is an $n.m$ dimensional vector representing the form of the final network output vector v, which corresponds to the problem solution denoted by p. The vector $v(p)$ is defined by:

$$v(p) = \begin{bmatrix} \delta(p_1) \\ \delta(p_2) \\ \vdots \\ \delta(p_n) \end{bmatrix} \quad \textbf{(11)}$$

- $vec(U)$ is a function which maps the mxn matrix to the $n.m$-element vector v. This function is defined by:

$$v = vec(U) = [U_{11}\, U_{21} ... U_{m1} \quad U_{12}\, U_{22} ... U_{m2} \quad U_{1n}\, U_{2n} ... U_{mn}]^T \quad \textbf{(12)}$$

- $V(p)$ is an nxn dimensional matrix defined by:

$$V(p) = \begin{bmatrix} \delta(p_1)^T \\ \delta(p_2)^T \\ \vdots \\ \delta(p_n)^T \end{bmatrix} \quad \textbf{(13)}$$

where $[V(p)]_{ij} = [\delta(p_i)]_j$.

- $P \otimes Q$ denotes the Kronecher product of two matrices. If P is an nxn matrix, and Q is an mxm matrix, then $(P \otimes Q)$ is a (nm)x(nm) matrix given by:

$$P \otimes Q = \begin{bmatrix} P_{11}Q & P_{12}Q & \dots & P_{1n}Q \\ P_{21}Q & P_{22}Q & \dots & P_{2n}Q \\ \vdots & \vdots & \ddots & \\ P_{n1}Q & P_{n2}Q & \dots & P_{nn}Q \end{bmatrix} \quad \textbf{(14)}$$

• $w \otimes h$ denotes the Kronecher product of two vectors. If v is an n-element vector and h an m-element vector, then $(w \otimes h)$ is an $n.m$-element vector given by:

$$w \otimes h = \begin{bmatrix} w_1.h \\ w_2.h \\ M \\ w_n.h \end{bmatrix} \quad \textbf{(15)}$$

The properties of the Kronecher products utilized are [9]:

$$(\lambda w \otimes \gamma h) = \lambda\gamma(w \otimes h) \quad \textbf{(16)}$$

$$(w \otimes h)^T(x \otimes g) = (w^T x)(h^T g) \quad \textbf{(17)}$$

$$(P \otimes Q)(w \otimes h) = (Pw \otimes Qh) \quad \textbf{(18)}$$

$$(P \otimes Q)(E \otimes F) = (PE \otimes QF) \quad \textbf{(19)}$$

$$vec(Q.V.P^T) = (P \otimes Q).vec(V) \quad \textbf{(20)}$$

• o^n and O^n are respectively the n-element vector and the nxn matrix of ones, that is:

$$\left.\begin{matrix} [o]_i = 1 \\ [O]_{ij} = 1 \end{matrix}\right\} for\ i,j \in \{1..n\} \quad \textbf{(21)}$$

• R^n is an nxn projection matrix (i.e., $R^n.R^n = R^n$) defined by:

$$R^n = I^n - \frac{1}{n} O^n \quad \textbf{(22)}$$

The multiplication by R^n transforms the column sums of a matrix to zero.

4.2 Confinement of Structural Constraints Using Modified Hopfield Network

The equations of T^{val} and s are developed to force the validity of the structural constraints. These constraints, for dynamic programming problems, mean that one and only one state in each stage can be actived. Thus, the matrix $V(p)$ is defined by:

$$[V(p)]_{ij} \in \{1,0\}$$

$$\sum_{j=1}^{m} [V(p)]_{ij} = 1 \quad \textbf{(23)}$$

By using (7), a valid subspace ($V = T^{val}.V + S$) for the dynamic programming problem can be represented by:

$$S = V = \frac{1}{m} o^n.o^{m^T} \quad \textbf{(24)}$$

Equation (24) guarantees that the sum of the elements of each line of the matrix V takes values equal to 1. Therefore, the term $T^{val}.V$ must also guarantee that the sum of the elements of each line of the matrix V takes value equal to zero. Using the properties of the matrix R^n, we have:

$$V.R^m = T^{val}.V$$
$$I^n.V.R^m = T^{val}.V \quad \textbf{(25)}$$

Using (24) and (25) in equation of the valid subspace ($V = T^{val}.V + S$),

$$V = I^n.V.R^m + \frac{1}{m}.o^n.o^{m^T} \quad \textbf{(26)}$$

Applying operator *vec(.)* given by (20) in (26),

$$vec(V) = vec(I^n.V.R^m) + \frac{1}{m} vec(o^n.1.o^{m^T})$$

$$vec(V) = (I^n \otimes R^m).vec(V) + \frac{1}{m} (o^n \otimes o^m) \quad \textbf{(27)}$$

Changing *vec(V)* by *v* in equation (27), we have:

$$v = (I^n \otimes R^m).v + \frac{1}{m} (o^n \otimes o^m) \quad \textbf{(28)}$$

Thus, the parameters T^{val} and *s* are given by:

$$T^{val} = (I^n \otimes R^m) \quad \textbf{(29)}$$

$$s = \frac{1}{m} (o^n \otimes o^m) \quad \textbf{(30)}$$

Equations (29) and (30) satisfy the properties of the valid subspace, i.e., $T^{val}.T^{val}=T^{val}$ and $T^{val}.s=0$.

The energy function E^{op} of the modified Hopfield network for the shortest path problem is projected to find a minimum path among all possible paths. When E^{op} is minimized, the optimal solution corresponds to the minimum energy state of the network. The energy function E^{op} is defined by:

$$E^{op} = \frac{1}{4}[\sum_{x=1}^{n-1}\sum_{i=1}^{m}\sum_{j=1}^{m} \underbrace{\quad}_{1st.\ Term} \cdot \sum_{x=1}^{n}\sum_{i=1}^{m}\sum_{j=1}^{m} \underbrace{\quad}_{2nd.\ Term}$$
$$+[\sum_{x=1}^{1}\sum_{i=1}^{m} \underbrace{\quad}_{3rd.\ Term} \cdot \sum_{x=n}^{n}\sum_{i=1}^{m} \underbrace{\quad}_{4th.\ Term} \quad \textbf{(31)}$$

Therefore, optimization of E^{op} corresponds to minimize each term given by (31) in relation to v_{xi}. From (31), the matrix T^{op} and vector i^{op} can be given by:

$$[T^{op}]_{pq} = -[P]_{xi,yj} \cdot [Q]_{xy} \quad \begin{cases} [P]_{xi,yj} = \frac{1}{2} d_{xi,yj} \\ [Q]_{xy} = \delta_{(x+1)y} + \delta_{(x-1)y} \end{cases} \quad \textbf{(32)}$$

$$i^{op} = -[\underbrace{\quad}_{m} \underbrace{\quad}_{m.(n-2)}$$
$$\underbrace{\quad}_{m} \quad \textbf{(33)}$$

where: $T^{op} \in \Re^{nmxnm}$; $i^{op} \in \Re^{nm}$; $p = m.(x-1)+i$; $q = m.(y-1)+j$; $x,y \in \{1..n\}$; $i,j \in \{1..m\}$.

5 Genetic Algorithm for Objective Function Optimization

In GA, potential solutions to a problem are represented as a population of chromosomes and each chromosome stands for a possible solution.

Codification: the size of the chromosomes is equal *n*m* and each chromosome is encoded as a vector of floating point numbers in which each *m* components of the vector are the states of each stage, as illustrated in Fig. 3.

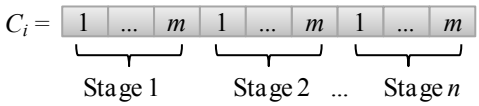


Fig. 3. Codification of chromosomes.

Population Size: The population size used here was 100 individuals, which allowed for a better coverage of the search space and was efficient in our experiments.

Initial Population: The initial population is generated by introducing a chromosome that represents the values *v(t)* obtained from Steps (I) and (II) described in Section 2. A total of 10% of the chromosomes are generated around *v(t)*, i.e., a random number between 0.001 and 0.5 is added or subtracted to *v(t)*. The remaining chromosomes are generated randomly.

Stopping Criterion: It is verified if either the average fitness or its standard deviation has not significantly changed after several generations, then the genetic algorithm stops.

Fitness Function: The fitness function evaluates each chromosome verifying its environment adaptation. The fitness function to be here minimized is that given in (31) and the most adapted individual will have the minimum fitness value.

Intermediate Population: The selection method used here to separate the intermediate population was the *N*-Way Tournament selection [10]. The selected individuals will constitute a population called intermediate population. The crossing and mutation methods used were the BLX-α crossing and uniform mutation, with rates defined, respectively, at 85% and 1%, as recommended in the literature [10].

6 Simulation Results

To ensure that the proposed approach can provide near-optimal solutions, variations on the number of stages and states have been simulated and compared with the approaches presented in [5] and [11].

The number of stages and the number of states in each stage were increased step by step. These numbers were obtained by using the values of the integer set defined by {2, 4, 8, 16, 32, 64}. The values of the weights $d_{xi,(x+1)j}$, which link the *i*-th state of stage *x* to the *j*-th state of the following stage (*x+1*) were randomly selected of the integer set defined by {1, 3, 5, 7, 9}. For those instances with number of stages and states less than 32, each experiment has been simulated 20 times with random initial conditions and metric data. Other cases have been simulated ten times.

Table 1 lists the simulation results. The columns *n* and *m* describe the number of stages and the number of state in each stage, respectively. The columns assigned as D^{NGA}, D^{MHA} and D^{DPN} show the average normalized path length obtained by the neurogenetic approach proposed in previous sections and the results obtained by [5] and [11], respectively. The average normalized path length is defined by:

$$D=\frac{S_c}{n_s(n+1)} \tag{34}$$

where S_c is the sum of the selected paths; n_s is the number of simulations e *n* is the number of stages.

Simulation results have demonstrated that proposed neurogenetic approach is an alternative for solving dynamic programming problems efficiently. All simulation results obtained by the NGA are near-optimal solutions.

Most of results presented in [11] were worse than those obtained by the neurogenetic approach. By other hand, the approach proposed in [5] obtained the best solutions. In spite of good results, it has a high computation time for problems with high number of stages and states in each stage, comparing with our approach.

Table 1. Simulation results.

n	m	D^{NGA}	D^{MHA}	D^{DPN}
2	2	3.67	3.13	3.25
4	4	2.60	2.03	3.12
8	8	1.50	1.34	2.00
16	16	1.35	1.06	1.85
32	32	1.54	1.03	1.61
64	64	1.73	1.02	1.39
16	2	3.71	3.14	3.21
16	4	1.45	1.79	2.98
16	8	1.54	1.26	1.85
16	32	1.35	1.13	1.79
2	16	1.67	1.17	1.53
4	16	1.16	1.02	1.60
8	16	1.33	1.09	1.76

7 Conclusions

The main advantages of using the modified Hopfield network proposed in this paper are i) the internal parameters of the network are explicitly obtained by the valid-subspace technique of solutions, ii) the valid-subspace technique groups all feasible solutions to the dynamic programming problem, iii) lack of need for adjustment of weighting constants for initialization, iv) for all classes of dynamic programming problems, a same methodology is adopted to derive the internal parameters of the network, and v) for industrial application, the neurogenetic approach offers simplicity of implementation in analog hardware.

References

1. Hopfield, J.J., Tank, D.W.: An Intelligent Algorithm based on Neural Network for Combinatorial Optimization Problems. In: 7th International Conference on Biomedical Engineering and Informatics, pp. 950-954 (2014)
2. Sanchez-Torres, J.D., Sanchez, E.N., Loukianov, A.G.: Recurrent Neural Networks with Fixed Time Convergence for Linear and Quadratic Programming. In: International Joint Conference on Neural Networks (IJCNN), pp. 1-5 (2013)
3. Zhang, H., Qin, C., Luo, Y.: A Neural-Network-Based Constrained Optimal Control Scheme for Discrete-Time Switched Nonlinear System Using Dual Heuristic Programming. IEEE Transactions on Automation Science and Engineering, vol. 11, no.3, pp. 839-849 (2014)
4. Li, G., Li, J., Xiao, H., Ding, S.: A Weak Filled Function Algorithm for Unconstrained Searching Optimization. In: 8th International Conference on Computational Intelligence and Security, pp. 31-34 (2012)
5. Silva, I.N., Goedtel, A., Flauzino, R.A.: The Modified Hopfield Architecture Applied in Dynamic Programming Problems and Bipartite Graph Optimization. International Journal of Hybrid Intelligent Systems, vol. 4, pp. 17-26 (2007)

6. Hillier, F.S., Lieberman, G.J.: Introduction to Operations Research. Holden Day, San Francisco (1980)
7. Gen, M., Cheng, R., Wang, D.: Genetic Algorithms for Solving Shortest Path Problems. In: IEEE International Conference on Evolutionary Computation, pp. 401-406 (1997)
8. Aiyer, S.V.B., Niranjan, M., Fallside, F.: A Theoretical Investigation into the Performance of the Hopfield Network. IEEE Transactions on Neural Networks, vol. 1, pp. 204-215 (1990)
9. Graham, A.: Kronecher Products and Matrix Calculus. Ellis Horwood Ltd., Chichester, UK (1981)
10. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Massachusetts (1996)
11. Chiu, C., Maa, C.Y., Shanblatt, M.A.: Energy Function Analysis of Dynamic Programming Neural Networks. IEEE Transactions on Neural Networks, vol. 2, pp. 418-426 (1991)

Heuristics strategies for intermodal transportation system

A Metaheuristic Approach for Solving the School Bus Routing Problem in Tunisia

Bochra Rabbouch[1], Foued Saadaoui[2], and Rafaa Mraihi[3]

[1] Higher Institute of Management of Tunis
bochra.rabbouch@gmail.com
[2] Saudi Electronic University
f.saadaoui@seu.edu.sa
[3] Higher Institute of Trade of Tunis
rafaa.mraihi@yahoo.com

Abstract. This position paper presents the urban school bus routing problem in the Tunisian case. It is a typical challenging variant of the vehicle routing problem in which three simultaneous decisions have to be made : determine the set of stops to visit, determine for each student which stop(s)he should walk to, and determine routes visited with the chosen stops, so that the total traveled distance is minimized. Due to the NP-hard nature of the problem and accordingly to the Tunisian case study and its difficulty, a population based metaheuristic algorithm is proposed.

Key words : Combinatorial optimization problem , NP-hard problem, meta-heuristics, Genetic algorithm, School bus routing problem, Tunisian case.

1 Introduction

The school bus routing problem (SBRP) is a problem of affectation of school buses in order to satisfy the demand of spatial distributed students from their residences to their school's destination. It was proposed, for the first time, by Newton and Thomas [5] and a detailed literature is well summarized in a paper written by Park et al. [6]
This problem deals with how to pick up students from bus stops and deliver them to their appropriate schools safely, in most economical and most convenient manner under many operational constraints such as :

- The student transport demand
- Buses parking areas
- The maximum capacity of a bus
- The maximum riding time of a student in a bus
- The time window of a school
- Maximum walking distance : a student's maximum walking distance from his/her home to the bus stop.

2 Case study

This research is motivated by a real-life school bus routing problem. In Tunisian city, transportation authorities have chosen to integrate school bus transportation in the public bus system and to make a strict regulation for public companies [2]. In this context, the Tunisian ministry of transportation indicates in a statement that the number of school subscribers to the public transport network for the 2014/2015 academic year reached 439.000 students in which 303.000 pupils and 136.000 university students and 1923 buses were reserved for these 439,000 students hence 52% of the total bus fleet under public companies account 3772 bus.

Authorities regulate the price of school bus service but give important subsidies to help companies in order to respect public obligations, namely, efficiency, effectiveness and equity. Despite these subsidies, budget deficits continue to increase and the school transportation costs become more important both for the ministry of transport as well as for the companies. Reducing these

costs through operational variables becomes the main objective of Tunisian bus companies. So these companies are determined to make many decisions about the bus stop selection, the bus route generation, the adjustment to school time and the turning buses.

This study allows a reduction on students travel time and hence improving their quality of life, since they can arrive at school and home early. Then, the objective of this study to find some solutions in order to minimize :

- The transportation cost
- The amount of time students spend in the buses on the way to the school
- Total travel time at pick-up points
- The total bus travel time

3 Proposed approach

3.1 Bus routing problem

The SBRP has a total of five sub-problems including Data Preparation, Bus Stop Selection, Bus Route Generation, School Bell Time Adjustment and Route Scheduling [1]:

- **Data preparation** : It is the step of preparation of data for the next sub-problems such the specification of the road network, data for students including the location of home, data for schools consisting on its location, the starting and the ending time of the schools for bus arrivals then the maximum riding time of a student in a bus, etc.
- **Bus stop selection** : The assignment of students distributed in different areas to nearby bus stops. In the literature, the locations of the bus stops are given in a wide number of studies and a few papers considered bus stop selection, and use heuristic algorithms to select them.
- **Bus route generation** : Bus routes are generated in order to visit the determined bus stops. This sub-problem is similar to the vehicle routing problem.
- **School bell time adjustment** : this step is necessary for the multi school configuration. It covers the adjustment of the time by which the students should arrive at school
- **Route scheduling** : this step is necessary to specify the time exact of starting and ending of each route then it is possible to obtain a sequence of routes that can be served by the same bus.

In this paper, it was assumed that the data preparation, the bus stop selection and the school bell time adjustment problems had already been implemented and thus the necessary informations had been prepared since the case study is a real-life Tunisian case and we will work on data from the TRANSTU company [4].

3.2 Genetic algorithm

As mentioned previously, the bus route generation problem is similar to the Vehicle Routing Problem and try to generate optimal operation routes of buses between stops. It is a Non-deterministic Polynomial time hard (NP-hard) problem for which meta-heuristics are often applied to produce good quality solutions for a such 'hard' problems in a reasonable computation time (polynomial time) rather than exact approaches which are not very suitable considering their limitations for large scale problems. In this study, we selected to apply genetic algorithms to resolve this combinatorial problem.

The Genetic Algorithms or evolutionary algorithms were first introduced by Holland [3] in 1975 and became popular as an intelligent optimization technique for solving many hard problems especially for different routing problems. Its original idea is to mimic the biological process of natural selection to artificial systems where a population of entities is maintained randomly and each entity is called a chromosome which consists of genes. Then, a reproductive process allows parent

[4] Transport Tunis Company

solutions to be selected from the population in every evolutionary step, denoted as a generation. In each generation, a fitness value (usually the value of the objective function in the optimization problem being solved) is assigned to each solution (every individual in the population) in order to evaluate them according to some predefined quality criterion. It is based on the principle of the survival of the fittest. Those highly fit individuals are selected and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Finally, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population which should converge to an optimal or near optimal solution.
When two or more parents are selected for reproduction, the operators of crossover and mutation are used.

Algorithm : The basic genetic algorithm

Begin

1 : **Coding**: initialize and encode a random population of solutions called chromosomes.
2 : Repeat
3 : **Fitness Assignment**: decode and evaluate the fitness of each chromosome.
4 : **Selection**: select some chromosomes from the current population for reproduction, the selection criterion is based on the fitness of the selected parents.
5 : **Recombination**: apply crossover between the selected parents to produce new children.
6 : **Variation**: apply mutation with a small probability to some genes of the newly produced offspring, or to selected members of the population.
7 : **Replacement**: integrate the newly generated offspring with the old population to create a new generation.
8 : until (a certain stopping condition is reached).

End

3.3 Chromosome initiation

Two types of chromosomes are suggested for the proposed genetic algorithm. The first one is a bus allocation chromosome that allocate buses to stops. The second is a voyage chromosome to design the order required to visit the intermediate points for buses.

3.4 Objective function

The results of a problem depend on its objective function which must be pertinent for the environments to which they are applied to efficiently solve SBRP. In this study, we suggest to minimize the number of buses used and the sum of the travel distance both for buses and students.

3.5 Initial solution generation

In genetic algorithms, initial solution is generated randomly and it has an important impact on the final solutions. Given that the SBRP is a large scale problem, it is suitable to ensure creating populations with good initial solution. Diverse initial solutions will be created in order to reduce the phenomenon of Premature Convergence [4].

3.6 Crossover operator

Crossover is analogous to a biological crossover : a process of taking more than one parent solutions and producing a child solution from them. The classical crossover operator is called one-point crossover, where the two coupled chromosomes are each cut once at corresponding points and the sections after the cuts are swapped. However, many crossover techniques have been invented like two-point crossover, k-point crossover and uniform crossover.
In this study, only the 1-point and 2-point crossover operators will be applied and the results will be compared.

3.7 Mutation operator

The second operator which used to maintain genetic diversity from one generation to the next is mutation. It is applied to each child produced from crossover where some of the genes are randomly selected, with some (small) probability, and the corresponding allele values are altered. There are two mutation operators : point mutation an inversion mutation.
In this study, these two methods will be used in combination in order to obtain good solutions.

3.8 Repair infeasible solutions

This step is required in order to alter infeasible solutions that may occur during the execution of the algorithm especially when the time constraint will not respected when buses arrive at school after the intended arrival time or when the limit of bus riding time is exceeded etc.
We will present method to repair the created voyage chromosomes to have shorter routes in order to found good solutions faster.

3.9 Elitism

The aim of this step is to allow the best entity from the current generation to carry over to the next, unaltered. This strategy guarantees that the solution quality obtained by the algorithm will not decrease from one generation to the next.

4 Conclusion

In this position paper, we have proposed a genetic algorithm based approach to generate bus routes for the school bus routing problem in Tunis city. We will test this approach on a real database from the TRANSTU company.

References

1. J. Desrosiers, J. A. Ferland, J.-M. Rousseau, G. Lapalme and L. Chapleau, (1981). An overview of a school busing system. In: Jaiswal, N.K. (Ed.), Scientific Management of Transport Systems. North-Holland, Amsterdam, 235-243.
2. J. Euchi and R. Mraihi, (2012). The urban bus routing problem in the Tunisian case by the hybrid artificial ant colony algorithm. Swarm and Evolutionary Computation 2:15-24.
3. J. Holland, (1975). Adaptation in Natural and Artificial Systems. Michigan Press.
4. M. Kang, S-K Kim, J. T. Felan, H. R. Choi and M. Cho, (2015). Development of a Genetic Algorithm for the School Bus Routing Problem. International Journal of Software Engineering and Its Applications, 9(5):107-126.
5. R. M. Newton and W. H. Thomas, (1969). Design of school bus routes by computer. Socio-Economic Planning Sciences, 3(1):75-85.
6. J. Park, and B. I. Kim ,(2010). The school bus routing problem: A review. European Journal of Operational Research, 202(2) : 311-319.

Metaheuristics for Electronic Design Automation, Embedded Systems and telecommunication field

A weighted sum multi-objective PSO for base station location in a cellular network

N. MEZHOUD[1,2,3], A. ZAATRI[2,] Z. HAMMOUDI[2], N. BOUZERA[1], M. OUSSALAH[3]

1. Electrical Engineering Laboratory of Bejaia (LGEB), Faculty of Technology, University of Bejaia, Targa Ouzamour ,06000 Bejaia, Algeria
mezhoud.naima@gmail.com

2. Faculty of Technology Sciences, University of Constantine 1, Compus Ahmed Hamani, Ain El bey Constantine, Algeria

3. University of Birmingham, School of Electronics, Electrical and System Engineering Edgbaston, Birmingham, B152TT, UK

Abstract : This paper deals with the problem of optimizing the base station location in a cellular network in a geographic area of interest, given the user density. The goal is to maximize coverage while the transmitted power and interferences are minimized. The PSO algorithm is used in order to identify the optimal sites coordinates. Several fitness functions have been tested. The results show that a good choice of the PSO algorithm parameters, leads to the convergence of the proposed algorithm. While, the use of the weighted sum multi-objective function constitutes the key of the success to solve the adressed problem.

Keywords : PSO algorithm, base station location, Cellular Network planning, weighted objective function.

1 Introduction

Optimization techniques have been widely used to solve problems in many engineering fields, such as electrical field, image processing, modeling, and telecommunication. The planning of a Cellular Network is an important stage before the installation process in order to achieve a good Quality of Service QoS with a minimum cost. The goal of coverage planning in any cellular network, is finding the optimal sites for the base stations in order to achieve total coverage with respect of the planning constraints and requirements. The base station location is critical task, particularly in the case of coverage limited network. Coverage planning is, generally, carried out with planning software using a digital map of the area to cover, with topography and morphography information. Before the coverage planning stage, a propagation loss model is chosen, and used with the collaboration with drive tests measurements, in order to tune the model parameters. The model selection is achieved according to the network cells characteristics, namely, the base station height, the frequency, the cell size and type, etc. the coverage optimization task include determining the minimum number of sites to cover the target area in order to achieve the best compromise between the goals of coverage and capacity [1].

Coverage optimization problems have been well investigated in different wireless networks types. E; Amaldi and all. [2], [3], [4] concentrate their works on the third generation network configuration. They used either variants of Integer Programming technique or hybridizations of it with Tabu Search and Greedy Algorithms. In [5], the authors have built a new technique with MADS and FUZZY C- MEANS algorithms. In literature, methaheuristic techniques show their superior performance over other optimization techniques. Genetic Algorithm (GA) has been used with success for our task in [6]. Therefore, it is known that GA needs big size of population and a big number of iterations to reach the convergence criterion. For this reason and due to its complexity, GA is a heavy technique to solve the complex problem of coverage optimization. At

the contrary, Particle Swarm Optimization algorithm (PSO) is a simple algorithm which can resolve our task with a little number of particles and in few iterations.

In this paper, Base Station sites coordinates, are optimized using PSO algorithm. The proposed approach identifies the locations coordinates in the within the geographical area to plan, by considering maximization of service coverage.

The organization of the paper is as follows. Section II discusses the Base Station placement model and enumerates the objectives of BS placement model. An overview of PSO algorithm is detailed in Section III. Section IV focuses on the implementations of PSO for base station location and section V discusses the simulation results. Section VI presents the conclusion.

2 Problem Formulation

We recall that a mobile is covered if the received signal strength is greater than a threshold determined experimentally The link budget calculation are used to estimate the maximum radius by service of a typical cell of the network. The output of these assessments is the maximum allowable loss, which depends on the type of environment, type of penetration and related effects due to UMTS (macro-diversity, shadowing, fast fading...). From this maximum allowable attenuation, is determined using a propagation model [7]: the cell range (radius), typical inter-site distance and the percentage of coverage (valid for three-sectorial sites).

$$d_{intersite} = 1.5 \times d_{CellRange}$$

$$S_{CoverageArea} = 1.95 \times d_{CellRange}^2$$

The coverage threshold is fixed to be the minimum signal strength allowed, and is the same in urban, suburban and rural areas: Rx=−94 dB [8].

The cell range R is then calculated using the Okumura-Hata model. After this, the site area can be calculated. The Okumura-Hata model is the most used empirical model. it is based on measurements made by Okumura in Tokyo. The model of Hata was spread in 2GHz to allow planning radio of networks DCS on 1800 [9]. Basically, the model provides an expression of the path loss of the signal strength transmitted by a base station (Tx) and the allowable minimum received signal signal strength at the mobile station receiver (Rx).

The following parameters will appear on the path loss equation of the model:

L : path loss=Tx-Rx (dB).

D : MS-BTS distance (Km).

f : carrier frequency (MHz).

hb :height of the base station (m).

hm : height of the mobile phone (m).

The domain of validity is given as follows:

30m ≤hb ≤300m

1m ≤hm≤20m

1km ≤ D≤20km

150MHz ≤f ≤2000MHz

In urban environment, the path loss in dB is given by:

L=69.55+26.16Log(f)-13.82Log(hb)-a(hm)+ [44.9-6.55Log(hb)]Log(d)

The parameter a(hm) is the mobile height correction factor or gain (dB). whose value is:

a(hm)=(1.1Log(f)–0.7)hm–(1.56Log(f)–0.8), for a city of medium size.

a(hm)=3.2log(11.75hm)² −4.97, for a big city.

In suburban environment, the Ls weakening expressed in dB is given while applying the formula urban environment affected of a correction:

Ls= L-2[Log (f/28)] -5,4

In farming environment, one distinguishes the case where the environment is cleared completely like in a desert (Lro weakening) or semi-free like in a nice country (Lrqo weakening):

Lro=L-4.78[Log(f)]2+18.33Log(f)-40,94

Lrqo=L-4.78[Log(f)]2+18.33Log(f)-35,94

Consequently, as all parameters are known, maximum distance D, which represents the cell rang R can be determined from the path loss (Tx − Rx) expression.

3 Particle Swarm Optimization

3.1 Main Theory

Particle Swarm Optimization is a stochastic optimization method, for the nonlinear functions, based on reproduction of a social behavior, and developed by Dr. EBERHART and Dr. KENNEDY [10], [11] in 1995. The algorithm generates a swarm of particles. Each particle is a possible solution of the optimization problem. This swarm flies in the search space (N-dimensional), and each member of it is attracted to his best solution and that of its neighbors (figure 1).

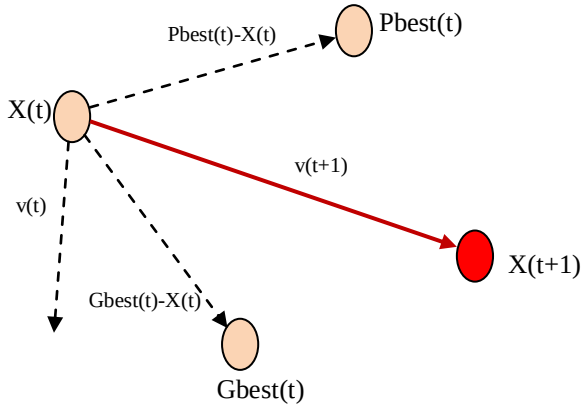


Figure 1: PSO principle

Each particle is equipped with a memory containing data on the flight (position and speed and better solution to the problem) and the ability to communicate (or socialize) with his entourage. At each iteration step, the result of the objective function is calculated for each member of the swarm to then determine its leader. The leader is the particle with the best solution of the swarm. And taking into account the inertial weight model, the position and velocity of a particle (i) is updated by:

$$\vec{v_i}(t+1) = w\,\vec{v_i}(t) + c1.rand.\left[\overrightarrow{pbest_i}(t) - \vec{x_i}(t)\right] + c2.rand.\left[\overrightarrow{gbest}(t) - \vec{x_i}(t)\right] \quad (1)$$

$$\vec{v_i}(t+1) = w\,\vec{v_i}(t) + c1.rand.\left[\overrightarrow{pbest_i}(t) - \vec{x_i}(t)\right] + c2.rand.\left[\overrightarrow{lbest_i}(t) - \vec{x_i}(t)\right] \quad (2)$$

$$\vec{x_i}(t+1) = \vec{x_i}(t) + \vec{v_i}(t) \quad (3)$$

In the above equations, rand is a random number between 0 and 1. The parameters c1 and c2 are cognitive parameter and social parameter respectively. The parameter w is the inertial weight of the particles. The value of Gbest is the best position of the swarm at iteration t, while Lbest is the best position in the geographical neighborhood to each particle. Pbest is the best position of the particle and ω inertia at time t.

The PSO algorithm has the distinction of being one of the simplest Meta-heuristic algorithms, in terms of complexity. Thus, only the memorization of Gbest (or Lbest) and Pbest are needed to calculate the next iteration with the previous equations. This feature is interesting in the context of an implementation in a system with low computing resources or subjected to real-time constraints.

Finally, the convergence towards the global optimal solution is not guaranteed in all cases. Therefore, it is strongly recommended to define a convergence criterion, which can be set as the maximum number of iterations to be reached or the minimum error criteria to be attained or the change in the speed to be very low.

3.2 The Basic PSO Algorithm

The main parameters of the PSO algorithm, which affects highly its performance, are

- The size of the problem: represents the resolution of the solution. The higher is the size of the particles the more precise is the solution, but at the expense of complexity.
- The number of particles: a small number of particles leads to low diversity of the solutions, and thus a high number of iterations is required to reach the convergence.
- The values of the social and cognitive coefficients: are generally set to be equal.
- The topology and size of the neighborhood: in the star topology, all particles are connected to all others,. In the ring neighborhood, each particle is connected to n particles (generally, n=3) in the radius topology, each particle communicate with only one core particle.
- The maximum velocity: to prevent the particles from moving too quickly in the search space, possibly passing next to the optimum, it may be necessary to establish a maximum velocity (Vmax) to improve the algorithm convergence
- Inertia weight: is employed to control the impact of the previous history of velocities on the current velocity.

The previous description can be summarized in the following algorithm:

PSO Algorithm
Initialize the number of particles
Initialize the convergence criterion
  For each particle i do
    Randomly initialize the position X_i
    Randomly initialize the velocity V_i
    Set $Pbest_i = X_i$
  end For
Repeat
  For each particle i do
    Evaluate the fitness of particle i
    Update $Pbest_i$
    Update gbest
    Apply velocity update using Equation (1) or (2)
    Apply position update using Equation (3)
  end For
Until the stop condition is satisfied
end Algorithm

4 PSO Algorithm for Base Station Location Problem

In this section, we present application of PSO technique for the base station location problem. The overall approach is summarized in the flowchart in Figure 2.

Some general assumptions have been adopted, without reducing the generality of the problem, which are:

- it has been considered an urban area with homogeneous traffic distribution;
- The base stations are placed at the same height, and have the same transmitted signal strength;
- The base stations have circular radiation diagrams of instead of three-lobes diagrams. This assumption, are made to simplify the implementation of the algorithm, does not influence the final result;
- The most popular propagation model is used, namely the Okumura-Hata model;
- The area of interest is calculated and the initial number of base stations is allocated based on the maximum area that a single base station covers. The maximum radius covered by a single base station using the Okumura-Hata model is 1.13 Km, and the area of interest is 39 km^2.

$$\frac{39}{\pi \times 1.13^2)} = 9.72$$

The minimum number of required base stations is 10.

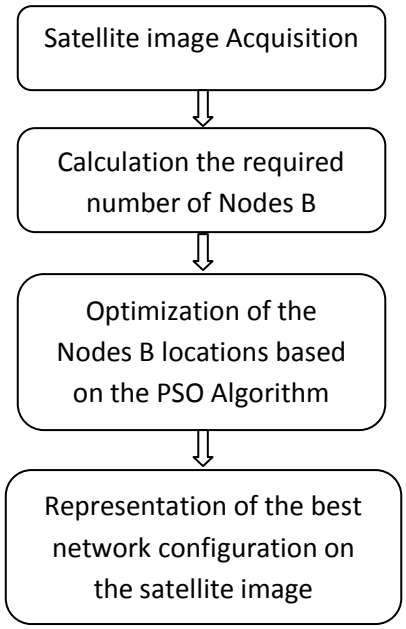


Figure 2. Flowchart of the optimisation process of the base stations location

Once the previous parameters are calculated, we apply our PSO based approach described as follows.

- *Encoding the particles:* a particle represent a configuration and is encoded by means of matrix *p*, with *p(x,y)*=1 if the site is selected, else it is set to 0. Note that the number of the selected sites is N=10.

- *Swarm initialization:* The number of particle (NPart) is set empirically. Then, we use fully random initialization in order to initialize the particles positions and velocities.

- *Convergence criterion*: The convergence criterion used in this paper is defined as the maximum number of iterations (NIter).

- *Fitness function:* the efficiency of an optimization algorithm depends mainly on the choice of its fitness function or goal. In this paper, we have tested several fitness functions, and the best one retained is the weighted sum of the outputs to minimize:

$$F_{fit} = (A \times UncovA + B \times OverCovA)/TotA$$

Where A and B are weighting parameters with their sum is equal to 1. The uncovered area is noted as $UncovA$ and the over covered area as $OverCovA$, and the total area as $TotA$.

5 Results

The coverage optimization process is carried out for an area in Algiers City (Algeria). The satellite image is acquired from GoogleEarth as in figure 3.

The parameters values used in our experiments are given in table 1.

Table 1. Experiments Parameters

Parameters	Values
Area to be covered: S	39 Km^2
Cell range: R	1.13 Km
Cell area: S_{cell}	4 Km^2
Required Number of NodesB: N	10
Number of particles: NPart	30
Number of iterations: NIter	200
Inertial weight: W	0.9 to 0.4
Cognitif parameter: C1	0.8
Social parameter: C2	0.8
Weighting parameter: A	0.8
Weighting parameter: B	0.2

Figure 4 shows the evolution of the fitness function over iterations, which decreases rapidly in the first iterations and remain stable for the rest of iterations. This proves the effectiveness of the proposed optimization approach. The best solution achieved represents a configuration that cover 90.39% of the considered area. This coverage could be enhanced by adding additional NodeBs.

Figure 5 represents the coverage of the retained configuration. The blue zone represents the uncovered area, while the covered area is represented by the green color. The overlapping areas are represented with the red colors.

The retained best configuration can be installed according to the best locations found after the optimization process using PSO, as in figure 6.

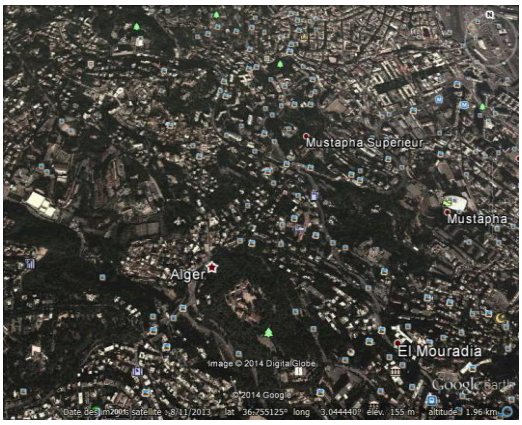


Figure 3. Area to cover in Algiers City

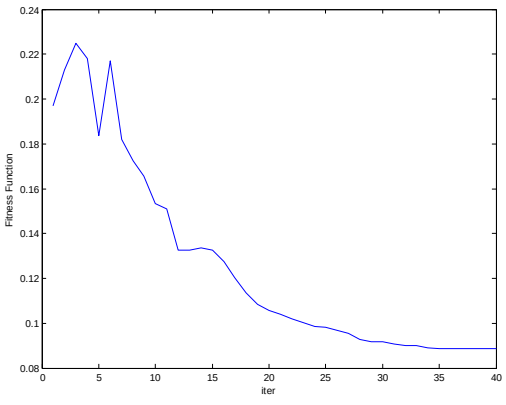


Figure 4. Fitness versus Iterations

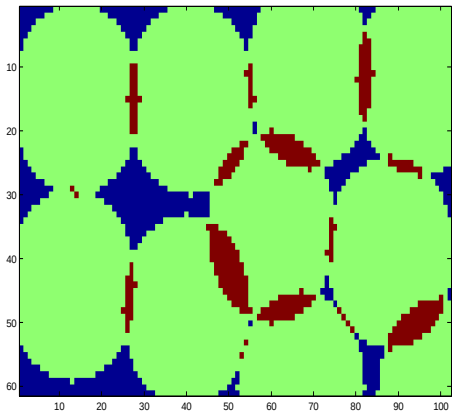


Figure 5. The Coverage representation

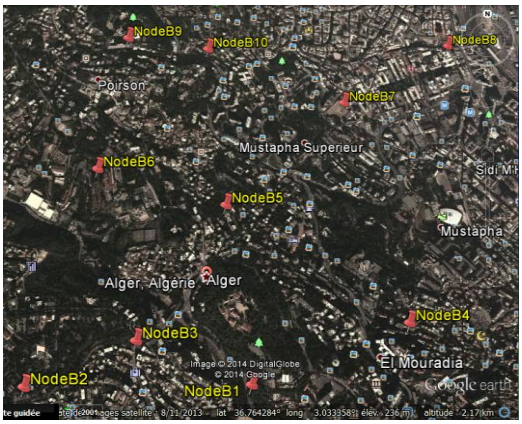


Figure 6. Installation of the optimal configuration

6 Conclusions

A particular swarm algorithm based technique to optimize the design of the third generation cellular network has been presented. The proposed method considers most of the limits imposed by the installation of the NodeBs necessary to guarantee an optimal service, also including environmental restrictions. In spite of some simplifications were made in our simulations, the adopted approach represents a powerful tool in designing the third generation network. Indeed, PSO algorithm can reach the convergence criterion even by using few particles and iterations.

References

[1] V. V. Veeravalli and A. Sendonaties (1999). The Coverage – Capacity Tradeoff in Cellular CDMA Systems. IEEE Trans. On Vehicular Technology, vol. 48, no. 5, Sep..

[2] E. Amaldi, A. Capone, F. Malucelli, and F. Signori (2002). UMTS Radio Planning: Optimizing Base Station Configuration. In Proceedings of IEEE VTC, Vol. 2, pp. 768–772.

[3] E. Amaldi, A. Capone 2003. Planning UMTS Base Station Location: Optimization Models With Power Control and Algorithms. IEEE Transactions on Wireless Communications, Vol. 2, No. 5, pp. 939-952.

[4] E. Amaldi, P. Belotti, A. Capone, F. Malucelli (2006). Optimizing base station location and configuration in UMTS networks. Annals of Operations Research, Volume 146, Issue 1, pp 135-151.

[5] A. O. Onim, P. K. Kihato, S. Musyoki (2013). Optimization of Base Station Location in 3G Networks using Mads and Fuzzy C-means. Journal of Network and Complex Systems, ISSN 2224-610X (Paper) ISSN 2225-0603 (Online), Vol.3, No.4.

[6] F. Garzia, C. Perna and R. Cusani (2010). Optimization of UMTS Network Planning Using Genetic Algorithms. Communications and Network, Vol. 2 No. 3, 2010, pp. 193-199. doi: 10.4236/cn.2010.23028.

[7] J. Laiho-Steffens, A. Wacker, K. Sipila (2000). Verification of 3G radio network dimentioning rules with static network dimensioning rules with static network simulations. VTS 2000, p. 478-482.

[8] Ajay R Mishra (2007). *Advanced cellular network planning and optimization*. John Wiley & Sons Ltd.

[9] N. Bouzera, A. Khireddine (2012). Comparison of Propagation Models for Small Urban Cells in GSM Network. International Journal of Communication and Networking System, Vol. 01, Issue: 01, ISSN: 2278-2427.

[10] J. Kennedy and R. Eberhart (2001). *Swarm Intelligence*. Morgan Kaufmann Publisher Inc.

[11] James Kennedy and Russell Eberhart(1995). Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks, pp.1942-1948, Piscataway, NJ, USA.

Ant Colony Optimization for Optimal Low-Pass Filters Sizing

KRITELE Loubna[1], BENHALA Bachir[2], ZORKANI Izeddine[1]
[1] University of Sidi Mohamed Ben Abdellah, Faculty of Sciences Dhar el Mahraz, Fez, Morocco.
[1]loubnakritele@gmail.com, izorkani@hotmail.com
[2] University of Moulay Ismail, Faculty of Sciences, Meknes, Morocco.
[2] b.benhala@fs-umi.ac.ma

Abstract.
In analog filter design, discrete components values such as resistors (R) and capacitors (C) are selected from the series following constant values chosen. Exhaustive search on all possible combinations for an optimized design is not feasible. In this paper, we present an application of the Ant Colony Optimization technique in order to selected optimal values of resistors and capacitors from different manufactured series to satisfy the filter design criteria. Three variants of the Ant Colony Optimization are applied, namely, the AS (Ant System), the MMAS (Min-Max AS) and the ACS (Ant Colony System), for the optimal sizing the for Low-Pass butterworth filter. SPICE simulations are used to validate the obtained results/performances.

Key-words: Optimization, Metaheuristic, Ant Colony Optimization, Low-Pass Butterworth Filter.

I. Introduction:

The optimal sizing of analog circuits is one of the most complicated activities, due to the number of variables involved, to the number of required objectives to be optimized and to the constraint functions restrictions. The aim is to automate this task in order to accelerate the circuits design and sizing. Recently, the used of the metaheuristics have proved a capacity to treat these problem efficiently, such as Tabu Search (TS) [1], Genetic Algorithms (GA) [2], Local search (LS) [3], Simulated Annealing (SA) [4], Ant Colony Optimization (ACO) [5,7] and Particle Swarm Optimization (PSO) [8].
Active analog filters are constituted of amplifying elements, resistors and capacitors; therefore, the filter design depends strongly of passive component values. While, the optimal selection of passive component values is very critical cause of the manufacturing constraints.
Furthermore, the search on all possible combinations in preferred values for capacitors and resistors is an exhaustive process, because discrete components are produced according to a series of values constants such as the series: E12, E24, E48, E96 or E192.
Consequently, an intelligent search method requires short computation time with high accuracy, must be used. Ant Colony Optimization is one such technique which is increasingly used in the domains of optimization and has been applied successfully in the field of analog circuits [5,7].
In this work, we propose to apply the AS, the MMAS and the ACS which are three variants of the ACO, for the optimal sizing of the Low-Pass Butterworth Filter.

II. Ant Colony Optimization: ACO technique: An over view:

ACO has been inspired by the foraging behavior of real ant colonies. Figure 1 shows an illustration of the ability of ants to find the shortest path between food and their nest [9,10]. It

is illustrated through the example of the appearance of an obstacle on their path. Every ant initially chooses path randomly to move and leaves a chemical substance, called pheromone in the path. The quantity of pheromone deposited will guide other ants to the food source. The indirect communication between the ants via the pheromone trail allows them to find shortest paths from their nest to the food source.

Fig. 1 Self-adaptive behavior of a real ant colony

II.1. Ant System:

The first variant of the ACO is « Ant System » (AS) which is used to solve combinatorial optimization problems such as the traveling salesman problem (TSP), vehicle routing problem... For solving such problems, ants randomly select the vertex to be visited. When ant k is in vertex i, the probability of going to vertex j is given by expression (1):

$$P_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha} * (\eta_{ij})^{\beta}}{\sum_{l \epsilon J_i^k} (\tau_{ij})^{\alpha} * (\eta_{ij})^{\beta}} & if \quad j \in J_i^k \\ 0 & if \quad j \notin J_i^k \end{cases} \tag{1}$$

Where J_i^k is the set of neighbors of vertex *i* of the *k*th ant, τ_{ij} is the amount of pheromone trail on edge *(i, j)*, α and β are weightings that control the pheromone trail and the visibility value, i.e. η_{ij} , which expression is given by (2):

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{2}$$

d_{ij} is the distance between vertices *i* and *j*.

Once all ants have completed a tour, the pheromone trails are updated. The update follows this rule:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{K=1}^{m} \Delta \tau_{ij}^{k} \tag{3}$$

Where ρ is the evaporation rate, *m* is the number of ants, and $\Delta\tau_{ij}^{k}(t)$ is the quantity of pheromone laid on edge *(i, j)* by ant *k*:

$$\Delta_{ij}^{k} = \begin{cases} {}^{Q}/_{L_k} & if\ ant\ k\ used\ edge\ (i,j) \\ & in\ its\ tour \\ 0 & Otherwise \end{cases} \tag{4}$$

Q is a constant and L_K is the length of the tour constructed by ant *k*.

II.2. Max-Min Ant System:

The Max-Min Ant System is another variant of ACO, which was developed by Stützle & Hoos [10, 11]. Max-Min ant system has always been to achieve the optimal path searching by

allowing only the best solution to increase the information and use a simple mechanism to limit the pheromone, which effectively avoid the premature stagnation.
MMAS which based on the ant system does the following areas of improvement:

i. During the operation of the algorithm, only a single ant was allowed to increase the pheromone. The ant may be the one which found the best solution in the current iteration or the one which found the best solution from the beginning of the trial. Consequently the modified pheromone trail update rule is given by:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij}^{best} \quad (5)$$

ii. In order to avoid stagnation of the search, the range of the pheromone trails is limit to an interval $[\tau_{min},\tau_{max}]$.

iii. The pheromone is initialized to τ_{max}in each edge.

II.3. Ant Colony System:

The ACS algorithm represents an improvement with respect to the AS. The ACS incorporates three main differences with respect to the AS algorithm:

i. ACS introduced a transition rule depending on a parameter q_0, which provides a direct way to balance between diversification and intensification. In the ACS algorithm, an ant positioned on node i chooses the city j to move to by applying the rule given by:

$$j=\begin{cases} argmax_{u\epsilon J_i^k} = [(\tau_{iu}(t)).\left(\mu_{ij}\right)^{\beta})] & if\ q \le q_0 \\ (1) & if\ q > q_0 \end{cases} \quad (6)$$

Where q is a random number uniformly distributed in [0, 1], q_0 is a parameter ($0 \le q_0 \le 1$).

ii. The global updating rule is applied only to edges which belong to the best ant tour. The pheromone level is updated as follows:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij} \quad (7)$$

$$\text{Where } \tau_{ij} = \begin{cases} \frac{1}{L} & if\ (i,j)\epsilon best-global-tour \\ 0 & otherwise \end{cases} \quad (8)$$

iii. while ants construct a solution a local pheromone updating rule is applied:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\tau_{int} \quad (9)$$

III. Application to the Optimal Design of Low pass Butterworth Filter:

The three proposed variants of ACO algorithm were used to optimize the analog circuit, namely low-pass Butterworth filter.
Analog active Filters are important building blocks in signal processing circuits. They are widely used in the separation and demodulation of signals, frequency selection decoding, and estimation of a signal from noise [12].
Analog active filters are characterized by four basic properties: the filter type (low-pass, high-pass, bandpass, and others), the passband gain (generally all the filters have unity gain in the passband), the cutoff frequency (the point where the output level has fallen by 3 dB from the

maximum level within the passband), and the quality factor *Q* (determines the sharpness of the amplitude response curve) [13].
Butterworth filters are termed maximally flat-espace filters, optimized for gain flatness in the passband. The transient response of a Butterworth filter to a pulse input shows moderate overshoot and ringing [14].
The fourth order low pass Butterworth filter formed by two operational amplifiers, four resistors and four capacitors. The schematic of this filter is given in Figure 2.

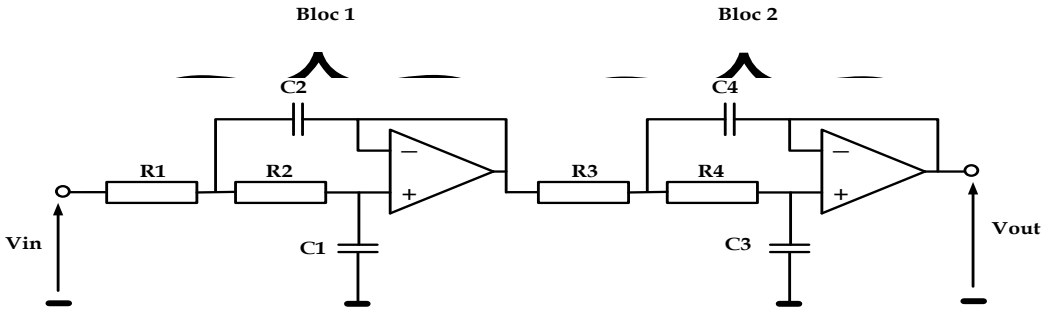


Fig.2. Butterworth fourth order low-pass filter [15]

The transfer function of this filter can be obtained as follows:

$$H(s) = \frac{w_{c1}^2}{s^2+\frac{w_{c1}}{Q_1}s+w_{c1}^2} \times \frac{w_{c2}^2}{s^2+\frac{w_{c2}}{Q_2}s+w_{c2}^2} \quad (10)$$

The cutoff frequency (w_{c1}, w_{c2}) and the selectivity factor (Q1 , Q2) of filter, which depend only on the values of the passives components, are given as follows [15]:

$$w_{c1} = \frac{1}{\sqrt{R_1R_2C_1C_2}} \qquad w_{c2} = \frac{1}{\sqrt{R_3R_4C_3C_4}} \quad (11)$$

$$Q_1 = \frac{\sqrt{R_1R_2C_1C_2}}{R_1C_1 + R_2C_1} \qquad Q_2 = \frac{\sqrt{R_3R_4C_3C_4}}{R_3C_3 + R_4C_3} \quad (12)$$

For comparison reasons, the specification chosen here is [13]:

- $\omega_{c1} = \omega_{c2} = 10\,000\ rad/s$ $(1591.55Hz)$
- $Q_1 = 1/0.7654$
- $Q_2 = 1/1.8478$

In order to generate ω_{c1} , ω_{c2}, Q_1 and Q_2 approaching the specified values; the values of the resistors and capacitors to choose should be able to satisfy desired constraints. For this, we define the Total Error (TE) [9] which expresses the offset values, of the cut-off frequency and the selectivity factor, compared to the desired values, by:

$$Total - error = 0.5\Delta\omega + +0.5\Delta Q \quad (13)$$

Where:

$$\Delta\omega = \frac{|w_{c1}-w_c|+|w_{c2}-w_c|}{w_c} \quad \text{and} \quad \Delta Q = \left|Q_1 - \frac{1}{0.7654}\right| + \left|Q_2 - \frac{1}{1.8478}\right| \quad (14)$$

The objective function considered is the Total Error (15). The decision variables are the resistors and capacitors forming the circuit. Each component must have a value of the standard series (E12, E24, E48, E96, E192). The resistors have values in the range of 10^3 to $10^6\Omega$. Similarly, each capacitor must have a value in the range of 10^{-9} to 10^{-6}F.
The aim is to obtain the exact values of design parameters ($R_{1...4}$; $C_{1...4}$) which equate the TotalError to a very close value to 0.

$$Total-error = 0.5\frac{\left|\frac{1}{\sqrt{R_1R_2C_1C_2}}-\omega_c\right|+\left|\frac{1}{\sqrt{R_3R_4C_3C_4}}-\omega_c\right|}{\omega_c}+0.5\left(\left|\frac{\sqrt{R_1R_2C_1C_2}}{R_1C_1+R_2C_1}-\frac{1}{0.7654}\right|+\left|\frac{\sqrt{R_3R_4C_3C_4}}{R_3C_3+R_4C_3}-\frac{1}{1.8478}\right|\right) \quad (15)$$

IV. Comparison results:

In this section we applied ACO algorithms to perform optimization of a low-pass Butterworth fourth order filter.
The studied algorithms parameters are given in Table 1 with a generation algorithm of 1000. The optimization techniques work on C codes and are able to link SPICE to measure performances.

Table 1. The ACO algorithms parameters

Evaporation rate (ρ)	0
Quantity of deposit pheromone (Q)	0.4
Pheromone Factor (α)	1
Heuristics Factor (β)	1
τ_{min}	0.5
τ_{max}	1.5

The optimal values of resistors and capacitors forming the low-pass Butterworth filter and the performance associated with these values for the different series using the three variants of the ACO technique: The MMAS,the AS and the ACS are shown in Tables 2, Tables 3 and Tables 4 respectively.

Table 2: Values of components and related butterworth filter performances using the "MMAS":

	R1 (KΩ)	R2 (KΩ)	C1 (nF)	C2 (nF)	R3 (KΩ)	R4 (KΩ)	C3 (nF)	C4 (nF)	$\Delta\omega$	ΔQ	**Erreur totale**
Linear values	4.900	2.750	10.0	74.2	3.750	1.150	37.7	61.6	0.0008	0.0011	**0.0010**
E12	4.700	2.700	10.0	68.0	3.900	1.200	39.0	56.0	0.0874	0.0841	**0.0857**
E24	5.100	2.700	10.0	75.0	3.900	1.200	39.0	62.0	0.0759	0.0010	**0.0430**
E48	4.870	2.740	10.0	75.0	3.830	1.150	38.3	61.9	0.0218	0.0135	**0.0176**
E96	4.870	2.740	10.0	75.0	3.740	1.150	37.4	61.9	0.0025	0.0125	**0.0075**
E192	4.930	2.740	10.0	74.1	3.740	1.150	37.9	61.9	0.0049	0.0029	**0.0039**

Table 3: Values of components and related butterworth filter performances using the “AS”:

	R1 (KΩ)	R2 (KΩ)	C1 (nF)	C2 (nF)	R3 (KΩ)	R4 (KΩ)	C3 (nF)	C4 (nF)	$\Delta\omega$	ΔQ	**Erreur totale**
Linear values	4.100	2.750	11.2	79.3	1.050	2.100	58.6	77.5	0.0014	0.0031	**0.0022**
E12	3.900	2.700	12.0	82.0	1.000	2.200	56.0	82.0	0.0227	0.0410	**0.0318**
E24	4.300	2.700	11.0	82.0	1.100	2.200	56.0	75.0	0.0309	0.0269	**0.0289**
E48	4.020	2.740	11.0	78.7	1.050	2.150	59.0	78.7	0.0473	0.0078	**0.0276**
E96	4.120	2.740	11.3	78.7	1.050	2.100	59.0	76.8	0.0024	0.0173	**0.0098**
E192	4.120	2.740	11.1	79.6	1.050	2.100	58.3	77.7	0.0019	0.0081	**0.0050**

Table 4: Values of components and related butterworth filter performances using the“ACS”:

	R1 (KΩ)	R2 (KΩ)	C1 (nF)	C2 (nF)	R3 (KΩ)	R4 (KΩ)	C3 (nF)	C4 (nF)	$\Delta\omega$	ΔQ	**Erreur totale**
Linear values	4.450	4.250	8.80	60.1	2.250	1.300	52.1	65.7	0.0007	0.0004	**0.0005**
E12	4.700	3.900	8.20	56.0	2.200	1.200	56.0	68.0	0.0926	0.0201	**0.0564**
E24	4.300	4.300	9.10	62.0	2.200	1.300	51.0	68.0	0.0250	0.0181	**0.0215**
E48	4.420	4.220	8.66	59.0	2.260	1.270	51.1	64.9	0.0493	0.0021	**0.0257**
E96	4.420	4.220	8.87	60.4	2.260	1.300	52.3	64.9	0.0017	0.0069	**0.0043**
E192	4.420	4.270	8.76	60.4	2.260	1.300	52.3	65.7	0.0054	0.0077	**0.0066**

From the results, we notice that the MMAS achieved a smaller design error.

Table 5: Component Values and Performance of GA, ABC, PSO and ACO Techniques for Butterworth Filter Design

	GA [13]	ABC [13]	PSO [16]	ACO ‘MMAS’	ACO ‘AS’	ACO ‘ACS’
R1 (KΩ)	6.80	4.70	4.58	4.93	4.12	4.42
R2 (KΩ)	6.80	4.70	4.70	2.74	2.74	4.22
C1 (nF)	5.60	8.20	8.20	10.0	11.1	8.87
C2 (nF)	39.0	56.0	56.0	74.1	79.6	60.4
R3 (KΩ)	39.0	1.00	1.10	3.74	1.05	2.26
R4 (KΩ)	1.00	39.0	1.00	1.15	2.10	1.30
C3 (nF)	4.70	4.70	87.6	37.9	58.3	52.3
C4 (nF)	56.0	56.0	102.2	61.9	77.7	64.9
Δω	0.0179	0.0201	0.0135	0.0049	0.0019	0.0017
ΔQ	0.0153	0.0024	0.0018	0.0029	0.0081	0.0069
TE	0.0166	0.0113	0.0076	**0.0039**	0.0050	0.0043

A comparison between these four techniques shows that the ACO algorithm achieved a smaller design error, in particular the MMAS.

The table 5 shows the comparison between the theoretical values and those practices for the error on the cut-off frequency for different series.

The following figure shows the PSPICE simulation of the filter gain for the optimal values of the E192 series. The practical cut off frequency are equal to 1590 Hz.

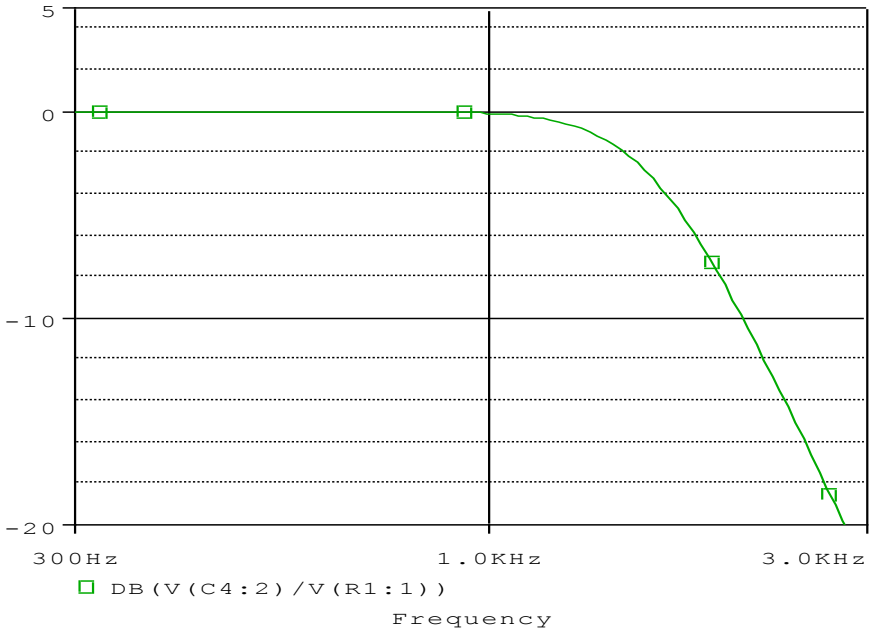


Fig.3. Frequency responses of low-pass filter

Table 5: Comparisons between the theoretical and practices for the error on the cut-off frequency

	MMAS		AS		ACS	
	$\Delta\omega$ theoretical	$\Delta\omega$ practical	$\Delta\omega$ theoretical	$\Delta\omega$ practical	$\Delta\omega$ theoretical	$\Delta\omega$ practical
E12	0.0874	0.1100	0.0227	0.0126	0.0926	0.0214
E24	0.0759	0.1307	0.0309	0.0126	0.0250	0.0100
E48	0.0218	0.0340	0.0473	0.0013	0.0493	0.0327
E96	0.0025	0.0100	0.0024	0.014	0.0017	0.0126
E192	0.0049	0.0013	0.0019	0.0100	0.0054	0.0013

V. Conclusion:

We presented in this paper an application of the Ant Colony Optimization technique for optimal sizing of a fourth order Butterworth low pass analog filter. The performance of the proposed method was also compared to already published evolutionary techniques. The design of the analog filter with high accuracy and short execution time is successfully realized using the ACO method.

References:

[1] F. Glover, Tabu search-part II, *ORSA Journal on computing*, Vol. 2, N° 1, 1990, pp. 4–32.

[2] J. B. Grimbleby, Automatic analogue circuit synthesis using genetic algorithms, *IEE Proceedings-Circuits, Devices and Systems*, Vol. 147, N° 6, 2000, pp. 319–323.

[3] E. Aarts and K. Lenstra, Local search in combinatorial optimization, *Princeton: Princeton University Press*, 2003.

[4] M. Fakhfakh, M. Bougharіou, A. Sallem and M. Loulou, Design of Low Noise Amplifiers through Flow-Graphs and their Optimization by the Simulated Annealing Technique, *Book: Advances in Monolithic Microwave Integrated Circuits for Wireless Systems: Modeling and Design Technologies*, *IGI global*, pp 89-103, 2012.

[5] B. Benhala, A. Ahaitouf, M. Kotti, M. Fakhfakh, B. Benlahbib, A. Mecheqrane, M. Loulou, F. Abdi and E. Abarkane, Application of the ACO Technique to the Optimization of Analog Circuit Performances, *Chapter 9, Book: Analog Circuits: Applications, Design and Performance, Ed., Dr. Tlelo-Cuautle, NOVA Science Publishers*, pp. 235–255. 2011.

[6] B. Benhala, A. Ahaitouf, A. Mechaqrane, B. Benlahbib, F. Abdi, E. Abarkan and M. Fakhfakh, Sizing of current conveyors by means of an ant colony optimization technique, *The IEEE International Conference on Multimedia Computing and Systems (ICMCS'11),* 2011, pp. 899–904, Ouarzazate, Morocco.

[7] B. Benhala, A. Ahaitouf, A. Mechaqrane and B. Benlahbib, Multiobjective optimization of second generation current conveyors by the ACO technique, *The International Conference on Multimedia Computing and Systems (ICMCS'12),* 2012, pp. 1147–1151, Tangier, Morocco.

[8] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry Analog Circuit Design Optimization through the Particle Swarm Optimization Technique, *Journal of Analog Integrated Circuits & Signal Processing, Springer,* Vol. 63, N° 1, 2010. pp. 71–82.

[9] M. Dorigo and S. Krzysztof, "An Introduction to Ant Colony Optimization," a chapter in Approximation Algorithms and Metaheuristics, a book edited by T. F. Gonzalez, 2006.

[10] M. Dorigo, G. DiCaro and L. M. Gambardella, "Ant algorithms for discrete optimization", Artificial Life Journal, vol. 5, 1999, pp. 137-172.

[11] T. Stützle, H. Hoos, "MAX-MIN Ant System", Future Generation Computer System, 16(2000), 889-914.

[12] L. D. Paarman, Design and Analysis of Analog Filters, Norwell, MA: Kluwer, 2007.

[13] R.A. Vural, T. Yildirim, T. Kadioglu and A.Basargan, Performance Evaluation of Evolutionary Algorithms for Optimal Filter Design, IEEE transactions on evolutionary computation Vol. 16, No. 1, 2012, pp. 135-147.

[14] R. Schaumann and M. V. Valkenburg, Design of Analog Filters. New York: Oxford Univ. Press, 2001.

[15] B. Benhala, Ant Colony Optimization for Optimal Low-Pass Butterworth Filter Design, WSEAS transactions on circuits and systems Vol. 13, 2014.

[16] [21] R. A. Vural and T. Yildirim, Component value selection for analog active filter using particle swarm optimization, in Proc. 2nd ICCAE, Vol. 1. 2010, pp. 25–28.

Heuristic and metaheuristic methods for complex systems scheduling

Non-dominated Sorting Genetic algorithms for a Multi-Objective Resource Constraint Project Scheduling Problem

Xixi Wang, Farouk Yalaoui and Frederic Dugardin

University of Technology of Troyes, ICD-LOSI, UMR CNRS 6281,
12 rue Marie Curie, CS 42060, 10004 TROYES CEDEX, FRANCE
xixi.wang@utt.fr

Mots-Clefs. Scheduling problem, Resource constraints, Multi-objective optimization, Non-dominated Sorting Genetic Algorithm, Metaheuristic methods

1 Introduction

Firstly introduced in 1969 [7] and later proven NP-hard in the strong sense [1], the Resource Constraint Project Scheduling Problem (RCPSP) has attracted more and more attention in the last decades. Scheduling problem with a set of non-preemptive jobs submitted to a set of precedence relationships is considered. Let us denote the job set $J = \{0, 1, ..., n+1\}$ where the start and the end of the project is represented by two dummy activities 0 and $n+1$. A job $i \in J$ in process may require certain resources with limited capacities. A resource is renewable if its availability is brought to the maximum at the beginning of each period. If a resource is invested only once or progressively during the whole project, it is called non-renewable. The basic RCPSP aims at minimizing the project makespan with respect to the precedence relationships and resources constraints.
Several extensions of RCPSP are discussed in the literature as well, more specific job or project properties are involved in the extended problems such as the multi-mode RCPSP (MMRCPSP), where jobs can be proceeded with several possible modes that require different resources quantities. In this paper we tackle a multi-objective RCPSP (MORCPSP) with renewable resources.
Lot of works are devoted to solving the RCPSP with single objective in the literature, while the MORCPSP is not yet well documented.
Exact methods can be found in only few papers. In our previous work [8] (accepted by proceedings), the Two Phase Method (TPM) was applied by considering the makespan and total job tardiness criteria. The TPM is an exact method that allows finding the *optimal* Pareto front by solving a series of *weighted sum* mono-objective problems. To this purpose, we have presented two Mixed Integer Programming (MIP) formulations for the bi-objective problem. For instance, in an efficient mathematical model, we define a binary variable f_{it} that takes the value 1 if a job $i \in J$ has already finished before time t in the scheduling horizon H. Its completion time can thus be written as $C_i = H - \sum_{0 \leq t \leq H} f_{it} + 1$. Therefore, minimizing the makespan is to minimize the completion time of the last job, $C_{n+1} = H - \sum_{0 \leq t \leq H} f_{(n+1)t} + 1$. The tardiness is only taken into account if a job $i \in J$ finishes after its due date $\bar{d}_i$, so that the total job tardiness is $\sum_{i \in J} \max(C_i - d_i, 0)$. For detailed explanations, readers can refer to [8] where our method was presented from A to Z.
More works proposed to solve the MORCPSP with approximated methods. A hybrid Genetic Algorithm was developed in [4] to optimize the makespan and the total tardiness penalty. A problem with the makespan, the cost and the project quality criteria was solved with an imperialist competitive algorithm in [5].
Our study aims at solving a MORCPSP with three criteria, namely the makespan, the total job tardiness and the workload balance. The makespan concerns about the productivity, as it represents the project duration. The service quality relies on the job tardiness, high tardiness may lead to poor satisfaction of clients. The workload balance evaluates the resource utilization. Inspired by [6], the workload balance level is measured by the difference between the maximum and minimum resource consumptions. Maximizing the workload balance is thus to minimize the gap between the maximum and minimum resource utilizations. To our best knowledge, this third criterion has not been integrated to the RCPSP topic so far in the literature. Our problem is solved with Non-dominated Sorting Genetic Algorithm II and III (NSGAII and NSGAIII, respectively). Experiments are carried out with 1080 instances randomly generated under different parameters.

2 Metaheuristic methods - NSGAII and NSGAIII

The NSGAII is one of the most widely used metaheuristic methods for multi-objective optimization [2]. It follows the general scheme of genetic algorithms and integrates the notion of non-dominated relationships during the selection procedure. An approximated Pareto front is provided at the end of the algorithm.
Despite the fact that the NSGAII can be adapted to solve problems with several criteria, the original algorithm was developed for the bi-objective case. An enhanced version, the NSGAIII is proposed recently in [3], aiming at the solving optimization problems with many objectives. The difference between the NSGAII and the NSGAIII lies in the selection procedure. When the last front is involved, the NSGAII computes the *crowding distance* of solutions to measure their distribution in the solution space and promotes individuals with larger crowding distances. In the NSGAIII, a hyperplane is created based on the whole population so as to normalize the solutions. A set of *reference points* are evenly located on the hyperplane. Each solution is then associate with a reference point and we can thus estimate their distribution. Finally, the individuals associated with less visited reference points are chosen in priority.

3 Experiments

Our programs are tested with 1080 instances generated under different parameters. The *net complexity* (NC), defined by the ratio between the number of precedence relationships and the number of jobs, measures the complexity of the precedence graph. The *delay proportion* (DP) stands for the proportion of jobs whose due dates are set as their earliest finishing time. The *demand rate* (DR) is the probability of generating non null resource requirement for a given pair of job and resource. For a positive resource requirement, the quantity is chosen randomly between 1 and the maximum value, calculated as the product of the resource capacities and the *demand factor* (DF).
In earlier studies [8], we found that the exact method is only able to solve small instances within 1800s (until 20 jobs for certain parameters). On the contrary, the approximated methods allows to solve problems even larger size with much less computation time. Our metaheuristic methods are tested with 30, 60, 90 and 120 jobs, including 1080 instances. The results are evaluated by the hypervolume indicator [9] who takes the value in (0,1). It measures the coverage levels of the Pareto fronts on solution space, knowing that 1 stands for the perfect coverage and 0 represents extremely poor coverage.
In this paper we present briefly the computational results for medium size instances with 60 jobs, called G60. We set $NC = 1$, DP$\in \{0.3, 0.5, 0.7\}$, DR$\in \{0.4, 0.6, 0, 8\}$ and DF$\in \{0.5, 0.7, 0, 9\}$. Denote the sub-groups as $G60.\gamma.\alpha.\beta$ where γ, α and β stand for the position of chosen value in the DP, DR and DF sets, respectively. For example, if we set DP = 0.3, then we have $\gamma = 1$.
In table 1, we present the hypervolume results and highlight the better solutions for each sub-group. Generally speaking, the two methods provide alternatively better approximated Pareto fronts. The NSGAIII shows more advantages in 16 out of 27 sub-groups, not including the two where both methods are equally efficient. More precisely, for $\gamma = 1, 2, 3$, the NSGAIII gives more promising Pareto fronts in respectively 4, 5 and 7 sub-groups out of 9. This shows that the NSGAIII gives is more efficient for instances with tighter due dates. Regarding the DR, when $\alpha = 1, 2, 3$, the NSGAIII provides better solutions for 5, 6 and 5 sub-groups, respectively. Finally, when β takes the value 1, 2, and 3, the NSGAIII outperforms the NSGAII in respectively 6, 5 and 5 sub-groups. These tendencies also draw attention to the sub-groups G60.1.x.3, G60.2.x.2, G60.3.x.1 and G60.3.x.2, where the NSGAIII is equally or more promising than the NSGAII for all DR values. In the meantime, we can see that the DP seems to have a stronger effect than DP and DF on the results. The computational times of the two methods are very close so that they are not included in the table. For each generation, the NSGAII and NSGAIII require 0.17s and 0.18s in average, respectively.

4 Conclusion and perspectives

In this study the NSGAII and the NSGAIII are applied to solve a MORCPSP by optimizing the makespan, the total job tardiness and the workload balance level. Through the experiments we can see that the NSGAIII has better overall performance than the NSGAII but

Table 1: Hypervolumes for G60 with NC = 1

G60		$\beta = 1$		$\beta = 2$		$\beta = 3$	
		NSGAII	NSGAIII	NSGAII	NSGAIII	NSGAII	NSGAIII
$\gamma = 1$	$\alpha = 1$	0.25	**0.34**	**0.32**	0.28	0.29	**0.36**
	$\alpha = 2$	**0.36**	0.33	**0.32**	0.26	0.33	**0.36**
	$\alpha = 3$	**0.30**	**0.30**	**0.26**	0.25	0.36	**0.46**
$\gamma = 2$	$\alpha = 1$	0.34	**0.36**	0.31	**0.36**	**0.35**	0.30
	$\alpha = 2$	0.32	**0.36**	**0.27**	**0.27**	0.33	**0.34**
	$\alpha = 3$	**0.36**	0.32	0.25	**0.31**	**0.43**	0.39
$\gamma = 3$	$\alpha = 1$	0.33	**0.34**	0.27	**0.33**	**0.34**	0.30
	$\alpha = 2$	0.32	**0.36**	0.27	**0.30**	**0.38**	0.36
	$\alpha = 3$	0.26	**0.32**	0.26	**0.28**	0.34	**0.40**

this advantage is not absolute. In the meantime, the delay parameter seems to have stronger effects on the results than the requirement parameters. More tests will be carried out in our future works, in order to observe more general performances of these two methods on our problem.

References

1. J. Blazewicz, J.K. Lenstra, and A.H.G.R. Kan. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5:11–24, 1983.
2. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
3. K. Deb H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18:577–601, 2014.
4. K. Kim, Y. Yun, J. Yoon, M. Gen, and G. Yamazaki. Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Computers in Industry*, 56:143–160, 2005.
5. E. Nabipoor Afruzi, A.A. Najafi, E. Roghanian, and M. Mazinani. A multi-objective imperialist competitive algorithm for solving discrete time, cost and quality trade-off problems with mode-identity and resource-constrained situations. *International Journal of Project Management*, 50:80–96, 2014.
6. Y. Ouazene, F. Yalaoui, H. Chehade, and A. Yalaoui. Workload balancing in identical parallel machine scheduling using a mathematical programming method. *International Journal of Computational Intelligence Systems*, 7:58–67, 2014.
7. A.A.B. Pritsker, L.J. Watters, and P.M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16:93–108, 1969.
8. X. Wang, F. Dugardin, and F. Yalaoui. An exact method to solve a bi-objective resource constraint project scheduling problem. In *8th IFAC Conference on Manufacturing Modelling, Management and Control*, France, June 2016 (accepted).
9. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.

Efficient Fault-Tolerant Scheduling On Computer Network by using Particle Swarm Optimization

S.Khernane[1] A.Khernane[2] C.Arar[3]
[1]Student in Master at Department of Computer Science
University of Batna ,Algeria
khernane.sara@gmail.com
[2]University of Batna and LMA Laboratory
UMK Biskra,Algeria
khernane_a@yahoo.fr
[3] LASTIC laboratory,University of Batna, Algeria
chafik.arar@gmail.com

1 Introduction

The design and framework of a computer network, including the characteristics of individual hardware, software and transmission system components and how they interact in order to ensure the reliable transfer of information.With the increasing demands on system reliability,fault-tolerant scheduling is even more important when computer network is used to run real-time applications since failure of any router might produce disastrous results.

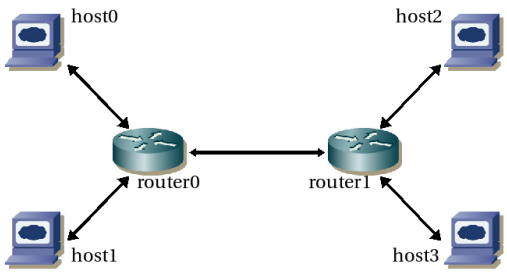


Figure 1: Basic components of a typical computer network.

The goal of fault-tolerant is to avoid the failure of the overall system when some of its router fail.
In this work we introduce how to find an optimal fault-tolerant schedule such that all the data can be guaranteed to transfered correctly with active and pas-

sive backup copies. The main objective is to minimize the scheduling length in the worst case. The metaheuristic algorithm which can achieve close to optimal results and the experimental results show the effectiveness of our technique.

2 The Proposed Approch

We assume that we have n-router in a system with names $R_1, R_2, R_3, ...R_{nrouter}$. Each router may fail due to hardware fault or software error. It is assumed that at most one router will fail to transmit properly in our proposed algorithm.
The input to our problem is defined by DAG, G=(D,E,Exe(D))
where D=$\{d_1, d_2, d_3, ...d_n\}$ represents data, E represents the data dependencies and Exe(D) represents the transmission cost.
Replication backup is a backup copy of d_i which trasmits independently no matter whether d_i trasmit successfully or not. In the case that the primary copy of d_i fails to transmit properly,the results of the replica copy can be used instead of those of the primary. It is an active backup.
Deallocating backup is a backup copy of d_i which transmits only if d_i fails. That is to say, the deallocationg copy must be scheduled to start after data d_i finishing transmitting. It is a passive backup.
We model data scheduling with binary variables to determine the order of data.$X_{i,j,k}$ is a binary variable such that $X_{i,j,k}$=1 if and only if data d_i start to be transmitted on router k. Similary,$X^R_{i,j,k}$ and $X^D_{i,j,k}$ are used for data d_i's replica and deallocationg backup copy.
The objective function can be written as follows

$$min \sum_{j=1}^{\lambda} \sum_{k=1}^{nrouter} j \times x_{d_0,j,k} \tag{1}$$

Where,
λ = an upper bound of the scheduling length.
nrouter = the maximum number of routers.
x_{d_0}= a dummy node add to DAG to compute the scheduling length of the whole graphe.
note:the values of $X_{i,j,k}$,$X^R_{i,j,k}$ and $X^D_{i,j,k}$ are between 0 and 1.

3 The experimental results

We used λ =3 and nrouter =3,the problem we consider here is to optimize the scheduling length.For do it, we try to determine $x_{d_0,1,1}$,$x_{d_0,2,2}$and $x_{d_0,3,3}$. The PSO method provides us the solution:$x_{d_0,1,1}$=0.2584,$x_{d_0,2,2}$=0.1151 and $x_{d_0,3,3}$=0.1006.

References

[1] Lima,G.d.A. and Burns, A.(2003)'An optimal fixed-priority assignment algorithm for supporting fault-tolerant hard real-time systems',IEEE Transactions on Computers, Vol.52,No.10,pp.1332-1346.

[2] Liestman, A.L. and campbell, R.H. (1986)'Afault tolerant scheduling problem',IEEE transaction on software Engineering,Vol.12,No.1,pp46-61.

[3] Zhang Jun, Efficient fault-tolerant scheduling on multiprocessor systems via replication and deallocation , Embedded Systems, Vol.6, Nos.2/3, 2014.

[4] Yang,C-H.,Deconinck,G.and Gui,W-H.(2004)'fault-tolerant scheduling for real-time embedded control systems',journal of computer science and technology,Vol.19,No.6,pp.800-812,2011.

[5] Zhu,X.,Qin,X.and Qiu,M.(2011)'Qos aware fault-tolerant scheduling for real time tasks on heterogenous clusters',IEEE Transactions on Computers,Vol.60,No.6,pp.800-812,2011

A new hybrid Heuristic for the Two-Dimensional Bin Packing Problem with Partial Conflicts

S.Mezghani[1], B. Haddar[1], and H. Chabchoub[1]

1. LOGIQ – Higher Institute of Industrial Management. University of Sfax, Tunisia
Salma_mezghani@hotmail.com, boukthir.haddar@gmail.com, habib.chabchoub@fsegs.rnu.tn

Keywords: Bin Packing problem, Conflicts, Relaxation, Heuristic.

1 Introduction

One of the extensively studied problems is the Two-Dimensional Bin Packing Problem (2DBP) (see [1],[2] that belongs to the classification of cutting and packing problems, has received an increasing amount of attention in the literature. Recently, there is a clear trend towards considering more complex problems by adding more constraints that are closer to reality for example conflicts constraints.(see [3], [4] and [5]. According to the literature; two types of conflicts have been treated: the hard conflict, where two items in conflicts cannot be packed in the same bin and, recently, the partial conflict, where two items must be separated by a given distance if they are placed in the same bin.

In this work, we will focus on a new variant Two Dimensional Bin Packing problem with Partial Conflict (2D- BPPC), introduced by [6]. The BPPC arises in many areas of industrial and transportation contexts such as scheduling communication systems, load balancing, database replicas storage, the assignment of processes or tasks to processors, etc.

A 2BPPC instance consists in a set $A = \{1, ...,n\}$ of items which have to be packed in a minimum number of identical bins. A bin is defined by its height H and its width W. An item i has a height h_i and a width w_i. A solution of the problem consists in assigning each item i to a bin and defining its position, denoted by (x_i, y_i) which corresponds to the coordinates of its bottom left-hand corner in the bin of its insertion, without overlapping while keeping a safety distance D between partially conflicting items. Let $P = (P_{ij})$ be the matrix of partial conflicts. P_{ij} is equal to one if and only if the items i and j have to be separated with a minimal given distance D_{ij} if they are assigned to the same bin, zero otherwise.
The following decision variables are used in the model:

- Binary variables V_k, equal to one if and only if the bin k is used.
- Binary variables Z_{ik}, equal to one if and only if the item i is packed in the bin k.
- Binary variable B_{ij} equal to 1 if the items i and j are placed in the same bin k
- Real variables x_i and y_i, determine the position of the item i in the used bin.
- Binary variables a^l_{ij}, equal to one if and only if the equation l connecting the items i and j is used.

In this paper we propose a new hybrid heuristic that combines the basic version of the convergent algorithm, with a local search strategy proposed recently by [7] for solving the 2D-BPPC problem.

Model 1:

$$Minimize: Z = \sum_{k}^{m} V_k \quad (1)$$

$$\sum_{k}^{m} Z_{ik} \leq n\, V_k, \quad (\forall k = 1,...,m) \quad (2)$$

$$\sum_{k}^{m} Z_{ik} = 1, \quad (\forall i = 1,...,n) \quad (3)$$

$$\sum_{i=1}^{n} Z_{ik}\,(w_i\, h_i) \leq W\, H, \quad (\forall k = 1,...,m) \quad (4)$$

$$x_i + w_i \leq W, \quad (\forall i = 1,...,n) \quad (5)$$

$$y_i + h_i \leq H, \quad (\forall i = 1,...,n) \quad (6)$$

$$| x_j - x_i - w_i | \geq D_{ij}P_{ij} + (M + D_{ij}P_{ij}) * (a_{ij}^1 + B_{ij} - 2), \quad (\forall i, j = 1,...,n) \quad (7)$$

$$| x_i - x_j - w_j | \geq D_{ij}P_{ij} + (M + D_{ij}P_{ij}) * (a_{ij}^2 + B_{ij} - 2), \quad (\forall i, j = 1,...,n) \quad (8)$$

$$| y_j - y_i - h_i | \geq D_{ij}P_{ij} + (M + D_{ij}P_{ij}) * (a_{ij}^3 + B_{ij} - 2), \quad (\forall i, j = 1,...,n) \quad (9)$$

$$| y_i - y_j - h_j | \geq D_{ij}P_{ij} + (M + D_{ij}P_{ij}) * (a_{ij}^4 + B_{ij} - 2), \quad (\forall i, j = 1,...,n) \quad (10)$$

$$a_{ij}^1 + a_{ij}^2 + a_{ij}^3 + a_{ij}^4 \geq B_{ij}, \quad (\forall (i, j) \in n) \quad (11)$$

$$V_k \in \{0,1\}, \quad (\forall k = 1,...,m) \quad (12)$$

$$B_{ij} \in \{0,1\}, \quad (\forall i, j = 1,...,n) \quad (13)$$

$$Z_{ik} \in \{0,1\}, \quad (\forall i = 1,...,n)(\forall k = 1,...,m) \quad (14)$$

$$a_{ij}^l \in \{0,1\}, \quad (\forall i, j = 1,...,n)(\forall l = 1,\ 2,\ 3,\ 4) \quad (15)$$

$$x_i,\ y_i \in IN, \quad (\forall i = 1,...,n) \quad (16)$$

Expression (1) minimizes the number of bins needed to contain all the items. Constraints (2) ensure that items are assigned only to a used bin. The constraints (3) imply that each item i is placed in exactly one bin. The constraints (4) means that the total capacity of items placed into bin k doesn't exceed the bin capacity. Constraints (5) and (6) are used to make sure that the items are really placed into the boundaries of the bins. Constraints (7)-(10) means that items are placed without overlapping with respecting the conflicts between items. The constraints (11) guarantee that at least one constraint among the last four is satisfied when the two concerned items are assigned to the same bin. Finally, the Constraints (12)-(16) indicate the nature of decision variables.

Iterative Linear Programming-based Heuristic

The Iterative LP-based Heuristic (ILPH) solves a series of small sub-problems generated by exploiting information obtained from a series of relaxations of the 2D-BPPC problem. We also apply local search methods to improve the solutions generated by these algorithms. At each iteration, the ILPH algorithm solves the LP-relaxation of the current problem P to obtain the optimal solution $\bar{x}$ (see line 4). The corresponding reduced problem), obtained from the 2DBPPC by fixing binary variable $= \overline{z_{ik}}$ for $\overline{Z_{ik}}$, is then exactly solved to generate a feasible solution Z^0 for the original problem $2D$ (see line 8). After that, a pseudo-cut is added to the current problem P to prohibit the search from revisiting the optimal solutions of the LP-relaxation that have already been evaluated in the past (see line 10). If the difference between the upper and the lower bounds is less than 1, then the process is stopped (see line 11). A local search is lunched when the convergent algorithm fails to prove its optimality (see line 13). Our local search uses the well-known Best-Fit Decreasing strategy. In our experiments, the LP-

relaxation of a current problem and the reduced problems are solved by CPLEX 12.5 solver. An illustration of the ILPH procedure is given in **Algorithm 1**.

Algorithm 1 A new hybrid heuristic algorithm for the 2D-BPPC problem

Input: A 2D-*BPPC* instance.
Output: An optimal solution of the .
1: Let be a feasible solution of if one is available;
2: PPC ; stop = False; $iter = 1$;
3: **while** stop = False **do**
4: Solve the LP-relaxation of *P* to obtain an optimal solution $\bar{z}$;
5: **if** **then**
6: $\bar{Z}$; stop = True;
7: **end if**
8: Generate an optimal solution of the reduced problem
$\bar{Z}$ for $\bar{Z}$
9: Update the best known-solution: **if** $cZ^0 > cZ^*$ **then** Z^* ^{0};
10: Update the current problem *P* by adding the pseudo-cut:
$\{\sum_{\bar{Z}} = \sum_{\bar{Z}}$
11: Check stopping criteria: **if** $\lfloor c$ 1⌋ or $r \leq Max$ **then** stop = True;
12: **end while**
13: *Z* = Local Search* (*Z**);
14: Return the best solution Z^* of the .

Assuming that all data are integers, if the condition $\lfloor c\bar{Z} - cZ^* < 1 \rfloor$ is satisfied, then the solution Z corresponding to the lower bound is an optimal solution for the problem P (where, for a real number α, $\lfloor \alpha \rfloor$ identifies the greatest integer$\leq \alpha$).

The proposed approach is tested on a wide set of 2DBPPC instances proposed by [8] and the obtained results are compared with those produced by the commercial solver CPLEX 12.5. Experimental results show that it can produce solutions of very good quality (optimal and near-optimal solutions) in a short and reasonable amount of computation time.

References

[1] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *Eur. J. Oper. Res.*, vol. 141, no. 2, pp. 241–252, 2002.

[2] N. Ntene and J. H. van Vuuren, "A survey and comparison of guillotine heuristics for the 2D oriented offline strip packing problem," *Discret. Optim.*, vol. 6, no. 2, pp. 174–188, 2009.

[3] M. Maiza and M. S. Radjef, "Heuristics for Solving the Bin-Packing," *Appl. Math. Sci.*, vol. 5, no. 35, pp. 1739 – 1752, 2011.

[4] A. Khanafer, F. Clautiaux, and E. G. Talbi, "Tree-decomposition based heuristics for the two-dimensional bin packing problem with conflicts," *Comput. Oper. Res.*, vol. 39, no. 1, pp. 54–63, 2012.

[5] R. Sadykov and F. Vanderbeck, "Bin packing with conflicts: A generic branch-and-price algorithm," *INFORMS J. Comput.*, vol. 25, no. 2, pp. 244–255, 2013.

[6] K. Hamdi-Dhaoui, N. Labadie, and A. Yalaoui, "Algorithms for the two dimensional bin packing problem with partial conflicts," *RAIRO - Oper. Res.*, vol. 46, no. 1, pp. 41–62, 2012.

[7] B. Haddar, M. Khemakhem, S. Hanafi, and C. Wilbaut, "A hybrid heuristic for the 0-1 Knapsack Sharing Problem," *Expert Syst. Appl.*, vol. 42, no. 10, pp. 4653–4666, 2015.

[8] J. O. Berkey and P. Y. Wang, "Two-Dimensional Finite Bin-Packing Algorithms," *J. Oper. Res. Soc.*, vol. 38, no. 5, pp. 423–429, 1987.

Scheduling meeting by CHN and Min-Conflicts heuristic

A. Bouhouch, C. Loqman, and A. El Qaddi

Team TIM High School of Technology
Moulay Ismail University,Meknes, Morocco
bouhouch.adil@gmail.com

Abstract. The Meeting scheduling problem (MSP) is an important vehicle of communication in business, factories and team w1orks. MSP requires a careful balance between the individual personal preferences and the organization. In this paper we propose a new neural network approach to solve meeting scheduling problem. The proposed network combines the characteristics of neural networks and minimizing conflicts approach. Our approach is divided into three steps: The first concerns formulating a meeting scheduling problem as a CSP problem, then we reformulate this CSP as a quadratic problem under linear constraint(QP). The second step applies the continuous Hopfield network to solve the QP. The later step involves the Min-conflict heuristic to improve the solution given by the second step. For experimentation, we implement two solver (CHN-MNC and CHN-BMNC) based our approach and we compare them with direct approach which involves CHN. Comparison is down on random generated meeting instances.

Mots-Clefs. Constraint Satisfaction Problem; Continuous Hopfeild Neural Network; Meeting Scheduling; Quadratic Problem;Min-Conflict.

1 Introduction

The Meeting Scheduling problem(MSP)is a very relevant problem for large organizations meetings. Each meeting involve group of agents and include at least two participants. Agents representing the peoples participating in meeting and the solution is to found start time for each meeting with respecting some constraints. Each constraint limits for two meetings do not overlap each other if they share a last agent, they can't be started at the same times. The MSP can be viewed as a set of temporal and disjunctive constraints and so they can be formulated as a Constraint Satisfaction Problem. An instance of the CSP involves a set of variables, a domain for each variable and a set of constraints. The goal is to assign values to variables such that all constraints are simultaneously satisfied. There has been a long history of using neural networks for combinatorial optimization and constraint satisfaction problems [1, 2]. Recently, a new approaches based on continuous hopfeild neural network [3] have proved an efficient to solve optimization problem [4][5]. However, symmetric Hopfield networks and similar approaches use steepest descent dynamics that converge to the closest local minimum of the energy, which result of its deterministic input-output interaction of the units in the network. Consequently, the network is not able to escape from local minima closed to starting point. Also, for Hopfield neural network with continuous dynamics case, each unit output can take any value between 0 and 1. So, the network can be stranded at a local minima which contains some units that remains in real values. If the last problem appear we get incomplete solution specially for affectation problem such as MSP. To overcome those problems, we introduce a new hybrid approach to solve MSP, the mean idea of this approach is to repair the solution given by the CHN and improve it by a known Min-conflict algorithm [6]. In this paper, our main objective is to present a new method to solve the MSP problem by operating on it reformulation as CSP problem, then we involve the continuous Hopfeild network and Min-Conflict heuristic to solve the equivalent CSP. This paper is organized as follows: In section 2, we provide a formulation of MSP problem as a CSP and the CHN model to solve it. In section 3, we introduce two hybrid approaches to solve MSP (CHN-MNC and CHN-BMNC). The experimental results are presented in the last section.

2 CSP and Hopfield neural network to solve Meeting scheduling problems

2.1 Meeting scheduling problem

MSP is a one of recurrent and real problems. It has been widely studied in several works for a long time. It is also easier to reformulate it as a constraints satisfaction problem (CSP). But many of work resolve MSP as Distribute Constraint Satisfaction Problem Most of them take into consideration only time and not location/geometry, [7] [8] [9]. On the other hand, There are many work like [10] which propose an efficient meeting-location algorithm that considers the time in-between two consecutive meetings. However, all private information about users is public. The systems in [11], [12] apply distributed meeting scheduling algorithms. Rasmussen and Trick [13] define the timetable constrained distance minimization problem, which is a sports scheduling problem applicable for tournaments where the total travel distance must be minimized. They use integer programming and constraint programming formulation for the problem. Integer programming is also applied by Wang et al. [14] for scheduling meeting among students and teachers in universities. For another scheduling approach to school meetings (between teachers and parents), Rinaldi et al. [15] set weights and build directed graphs based on the time slots, then they find shortest paths in the resulting graphs. Another kind of resolving approaches use meta-heuristic methods, tsuchiya [16] use a parallel algorithm for solving meeting schedule problems, the proposed system is composed of two maximum neural networks which interact with each other. The general definition of the MSP is as follows:

- A group S of m agents
- A set T of n meetings
- The duration of each meeting m_i is $duration_i$
- Each meeting m_i is associated with a set s_i of agents in S, that attend it.

So, each meeting attend a set of agent, each meeting is associated with a location. An agent must particeps at all meeting who attend him, since the scheduled time-slots for meetings in T must enable the participating agents to travel among their meetings.

Example 1. The table below presents an example of a MSP, and the traveling time in time-units between different meeting locations.

Meeting	**Location**	**Attending agents**
m1	L1	A1 , A3
m2	L2	A2, A3, A4
m3	L3	A1, A4
m4	L4	A1, A2

Table 1. Example of meeting scheduling

And distance between each location (see Fig.1):

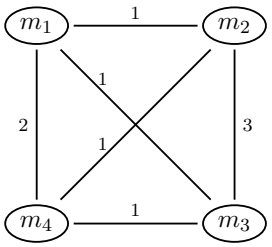


Fig. 1. Distances between each location in times unites

2.2 Quadratic model of Constraint satisfaction problem

A large number of real problems such as artificial intelligence, scheduling, assignment problem can be formulated as a Constraint Satisfaction Problem. A CSP consist to find an assignment of all variables problem under constraints restriction. The CSP can be formulate as three set:

- Set of N variables: $X = \{X_i, 1 \leq i \leq N \}$.
- Set of N variables domains: $D = \{D_i, 1 \leq i \leq N\}$ where each D_i contains set of d_i range values for X_i.
- Set of M constraints: $C = \{C_i, 1 \leq i \leq M\}$.

Each constraint C_i associate an ordered variables subset which called the scope of C_i. The variables number of this subset is noted the arity. The meeting scheduling problem as described above can be naturally represented as a constraints satisfaction problem (CSP) in the following way:

- A set of variables $T = m_1, m_2..., m_n$ the meetings to be scheduled
- Domains of values D - all weekly time-slots
- A set of constraints $C = C_{ij}, 1 < i < j < n$ - for every pair of meetings m_i,m_j that attend at last the same agent. For each constraint C_{ij} is given as :let t_i, t_j the selected time-slots for m_i and m_j respectively, there is a conflict if $|t_i - t_j| - duration_i < \tilde{D}(L_i, L_j)$

With $\tilde{D}$ is traveling time on unite slot-times between two meeting location. Then we can easily reformulate CSP as a Quadratic Problem, by introducing for each CSP variable m_i a binary variable x_{ik}, where k varies over the range of m_i, given as follows:

$$x_{ik} = \begin{cases} 1, & \text{if the time-slot } t_k \text{ is assined to } m_i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

We deduce the objective function of its equivalent QP:

$$f(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} x_{ir} x_{js}\, Q_{ijrs}$$

With the quadratic term:

$$Q_{ijrs} = \begin{cases} 1 & \text{if}(|r - s| - duration_i < \tilde{D}(L_i, L_j)) \\ 0 & otherwise \end{cases} \tag{2}$$

Valid solutions must satisfy some strict constraints that can be written as linear equations: $\sum_{r=1}^{d_i} x_{ir} = 1$, for $i = 1..N$.

2.3 Hopfield neural network

Hopfield neural network was proposed by Hopfield and Tank [3][17][18]. It was first applied to solve the combinatorial optimization problems.

This neural network model consist with n interconnected neurons completely and it dynamic stats is governed by The following differential equation:

$$\frac{dy}{dt} = -\frac{x}{\tau} + T\,x + i^b \tag{3}$$

where

x vector of neurons input
y vector of output
T the Matrix of weight betwen each neurones pairs

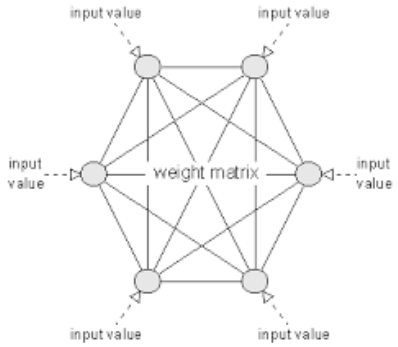


Fig. 2. Hopfeild Neural Network Architecture

For each neurones, the input is governed by an activation function $x = g(y)$ which varies between 0 and 1. g(y) is given by:

$$g(y) = \frac{1}{2}(1 + \tanh(\frac{y}{u_0}))$$

we define the energies function of CHN by:

$$E(x) = x^t\, T\, x + (i^b)^t\, x \tag{4}$$

In this paper we use the capacity of CHN to resolve Quadratic model optimization. So we choose a CHN energy function similar to [19], which is adapted to solve the quadratic formulation of the binary CSPs:

$$\begin{aligned} E(x) = \frac{\alpha}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} xirx_{js}\, Q_{ijrs} + \\ \beta \sum_{i=1}^{N} \sum_{r=1}^{d_i} x_{ir} + \gamma \sum_{i=1}^{N} \sum_{r=1}^{d_i} x_{ir}(1 - x_{ir}) \end{aligned} \tag{5}$$

The parameter settings are given by solving [20]:

$$\begin{cases} \phi \geq 0, \;\; \alpha > 0 \\ -\phi + 2\gamma \geq 0 \\ 2\phi + \beta - \gamma = \varepsilon \\ \alpha\, d + \beta + \gamma = -\varepsilon \end{cases}$$

We notice that this function relaxed by the aggregation of all linear constraints cited above. Empirically, we constat that applying CHN to solve this kind of problem network have 70% success rate and low quality of the given solution. To overcome this wakens, we propose to improve and repair the solution by a approach the known Min-conflict algorithm.

3 CHN and Min conflict heuristic to solve CSPs

The MNC [21] algorithm is very simple and fast local repair method to resolve CSPs, which consist to randomly assigned all variables. Next, it select iteratively one variable from the set of variables with conflicts, which violating one or more constraints of the CSP. Then it assigns to this variable the value with that minimizes the number of conflicts. MNC have demonstrated to be able to solve the million queens problem in minutes [6]. MNC is widely used to construct hybrid algorithms with other optimization [22–25]. In this way, the basic idea of our proposed approach is to use MNC to improve the solution reached by CHN(Figure 3). This will be done in two step. First, MNC(dettail of Min-Conflict in Figure 4) visit all assigned variables, for each one the decision will be tacked by computing penalty of it associated neurones. Second, we propagate this assignment to other set

of no assigned variables iteratively. We study to variant of execution of our methods: the first is CHN-MNC which execute CHN and MNC consecutively for each instance of problem, the second is CHN-BMNC, during it we run CHN many times to solve the given instance, then we execute min-conflict on the best solution founded by all run.

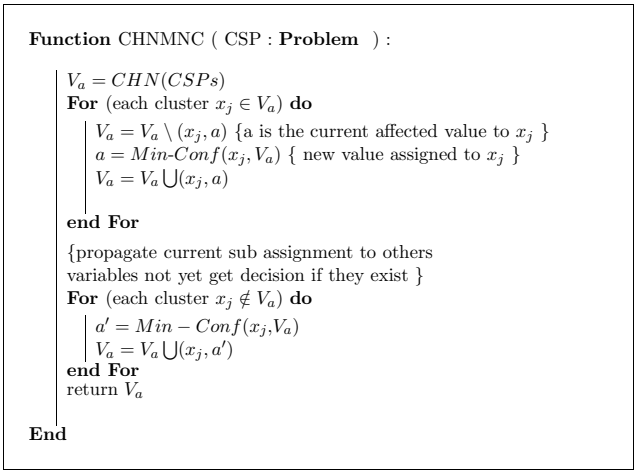


Fig. 3. Main function witch improve solution by Min-conflict algorithm

Remark 1. CHN as described in [19] with same values setting and start point.

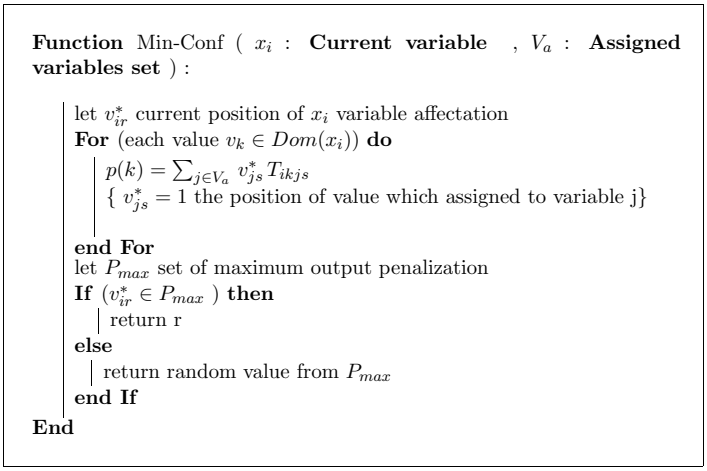


Fig. 4. Selected the most coherent neuron of current cluster with others variables clusters already affected

4 Numerical results

For showing the practical interest of our approach, we study also it performance over meeting instances provided in [26]. Table 1 show the comparison between our approach CHNMNC and original CHN with [20] parameters settings and for stabilisation point we have used the Variable Updating Step (VUS) technique proposed by Talaván and Yánnez in [27]. The description of table columns is fallow:

– M: the number of instance meetings

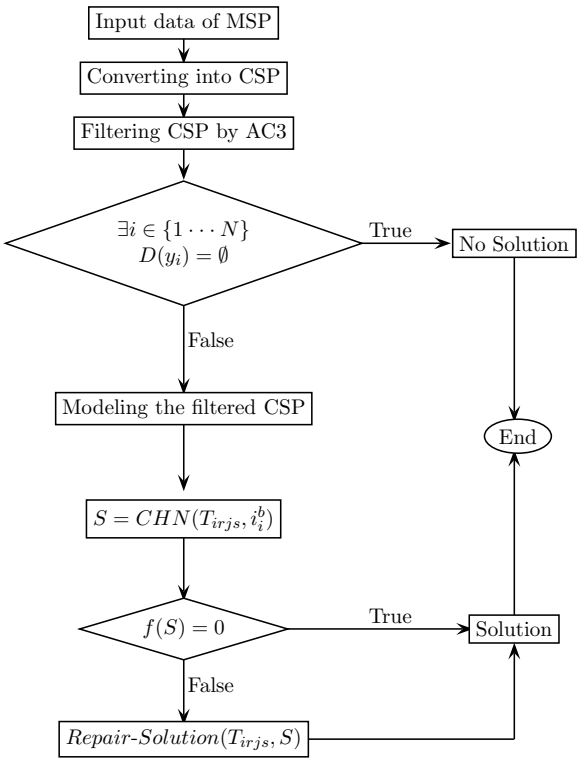


Fig. 5. Diagram of the proposed algorithm CHN-MNC

- C: the number of arrival constraints
- Ratio min the better results obtained by each solver
- Ratio mean: the average of the optimal value in a number of run.
- Ratio mod: the most repetitive (mode) optimal value obtained by CHN in number of run

For each instance we run solvers 200 time.

Fig.6, Fig.7 and Fig.8 plot the performance of CHN-BMNC against table 1 result on two class of instance, the classification is down by considering the number of instance variables, so the first class of instance have N=20, and for the secon class N=40. Result show the effectiveness of CHN-MNC variant than CHN-BMNC and CHN alone. This is due to the fact that is not necessaries to find a good minimum closed to the best solution founded by CHN solving phase.

In Fig.6, we observe that according to a minimum values over 200 run our approach variant CHN-MNC out perform significatively CHN alone and the second variant CHN-BMNC.

According to means values founded by two variant, Fig.7 report that CHN-MNC improve the solution meanly with 81%. Also for class N=20 and N=40 variant CHN-MNC is beter than CHN-BMNC by 50

5 Conclusion

In this paper, we have considered the meeting scheduling problem. We have solved it as a constraint satisfaction problem. After, reformulation we apply on AC3 algorithm to reduce the search space. Then, we use the a hybrid meta-heuristic search to solve it, which is based on Continuous Hopfeild network and Mon-conflict. Some numerical examples assess the effectiveness of the theoretical results are shown in this paper, and also the advantages of this new approach which improve considerably the solution quality and avoid network crush. Other studies are in progress to apply this approach to many problems such as tim tabling and resource allocation.

Table 2. CHN performance of CHN-MNC and CHN alone [26].

Instances nams	M	C	CHN-MNC			CHN					Number of iteration
			Ratio Min	Ratio Means	Ratio Mod	Ratio Min	Ratio Means	Ratio Mod	CPU(s)	Success rate	
Instance #1	20	78	0	2,6	3	10	18,36	20	0,42496	0.68	128.0
Instance #2	20	72	1	1,9	2	14	19,2	15	0,39195	0.62	112.0
Instance #3	20	75	0	1,5	1	12	15,54	15	0,41117	0.78	116.0
Instance #4	20	70	0	0,6	1	8	15,72	14	0,43679	0.83	155.0
Instance #5	20	66	0	0	0	8	13,48	13	0,42822	0.78	122.0
Instance #6	20	97	2	3,8	2	16	21,98	23	0,37535	0.70	126.0
Instance #7	20	93	2	4,8	4	15	21,96	21	0,43264	0.65	124.0
Instance #8	20	71	1	1,7	1	12	16,08	16	0,44782	0.67	127.0
Instance #9	20	80	0	1,9	1	12	17,91	17	0,38811	0.88	115.0
Instance #10	20	70	0	1,5	2	10	16,42	13	0,4034	0.79	117.0
Instance #11	20	72	0	2,2	3	11	17,15	19	0,42603	0.79	120.0
Instance #12	20	73	1	3,5	2	14	20,17	19	0,34246	0.72	96.0
Instance #13	20	64	1	1,9	1	11	17,35	16	0,37739	0.87	112.0
Instance #14	20	66	4	4,8	4	14	18,92	20	0,40289	0.81	125.0
Instance #15	20	70	0	2	2	9	14,88	15	0,42734	0.88	121.0
Instance #16	20	73	0	2,5	2	11	16,82	19	0,39188	0.67	121.0
Instance #17	20	74	1	2,7	1	12	16,81	17	0,41901	0.90	117.0
Instance #18	20	104	5	6,3	8	19	25,5	26	0,34146	0.60	102.0
Instance #19	40	125	3	6,3	6	20	29,47	30	0,48332	0.79	576.0
Instance #20	40	125	4	5,6	6	20	28,55	28	0,47398	0.74	553.0
Instance #21	40	121	4	6	8	23	30,09	33	0,48786	0.78	573.0
Instance #22	40	123	3	5,3	4	19	27,48	26	0,4796	0.84	572.0
Instance #23	40	96	0	3,2	4	14	23,5	22	0,47992	0.72	558.0
Instance #24	40	96	2	4,3	2	16	26,78	26	0,47745	0.77	577.0
Instance #25	40	96	2	4,6	7	10	26,65	25	0,47181	0.71	544.0
Instance #26	40	157	7	9,7	11	27	36,87	30	0,45777	0.81	532.0
Instance #27	40	75	4	6,8	8	18	26,86	23	0,50308	0.72	598.0

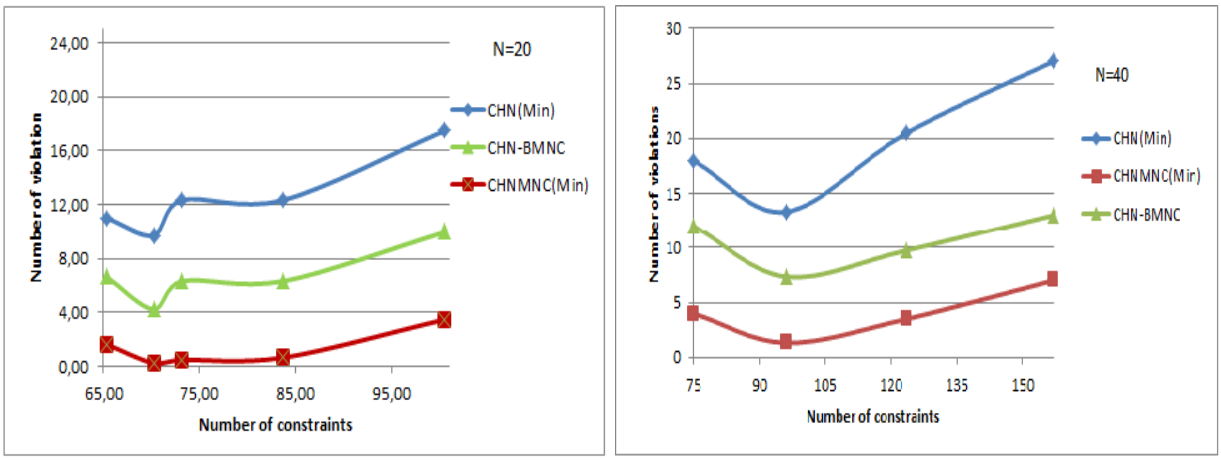


Fig. 6. Comparison between minimums founded by CHN-MNC and CHN against CHN-BMN.

-

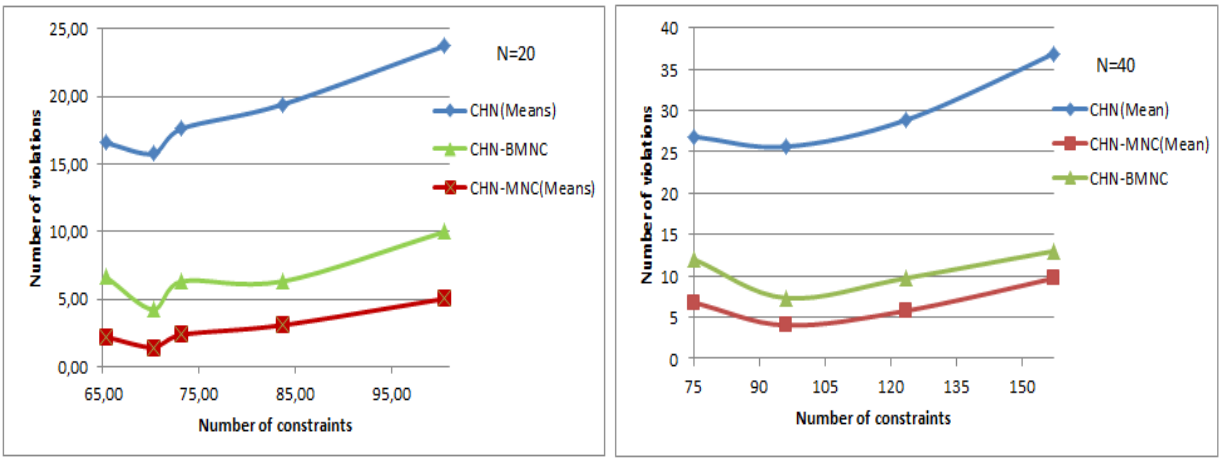


Fig. 7. Comparison between means values founded by CHN-MNC and CHN against CHN-BMN.

References

1. Lee, J.H., Leung, H., Won, H.: Extending genet for non-binary csp's. In: Tools with Artificial Intelligence, 1995. Proceedings., Seventh International Conference on, IEEE (1995) 338–343
2. Davenport, A., Tsang, E., Wang, C.J., Zhu, K.: Genet: A connectionist architecture for solving constraint satisfaction problems by iterative improvement. In: AAAI. (1994) 325–330
3. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences **79** (1982) 2554–2558
4. Ettaouil, M., Loqman, C., Haddouch, K.: Job shop scheduling problem solved by the continuous hopfield networks. Journal of Advanced Research in Computer Science **2** (2010) 31–47
5. Ettaouil, M., Loqman, C., Hami, Y., Haddouch, K.: Task assignment problem solved by continuous hopfield network. International Journal of Computer Science Issues **9** (2012) 206–2012
6. Minton, S., Philips, A., Johnston, M.D., Laird, P.: Minimizing conflicts: A heuristic repair method for constraint-satisfaction and scheduling problems. Journal of Artificial Intelligence Research **1** (1993) 1–15
7. BenHassine, A., Ho, T.B.: An agent-based approach to solve dynamic meeting scheduling problems with preferences. Engineering Applications of Artificial Intelligence **20** (2007) 857–873
8. Sen, S.: Developing an automated distributed meeting scheduler. IEEE Expert **12** (1997) 41–45
9. Shakshuki, E., Koo, H.H., Benoit, D., Silver, D.: A distributed multi-agent meeting scheduler. Journal of Computer and System Sciences **74** (2008) 279–296
10. Berger, F., Klein, R., Nussbaum, D., Sack, J.R., Yi, J.: A meeting scheduling problem respecting time and space. GeoInformatica **13** (2009) 453–481

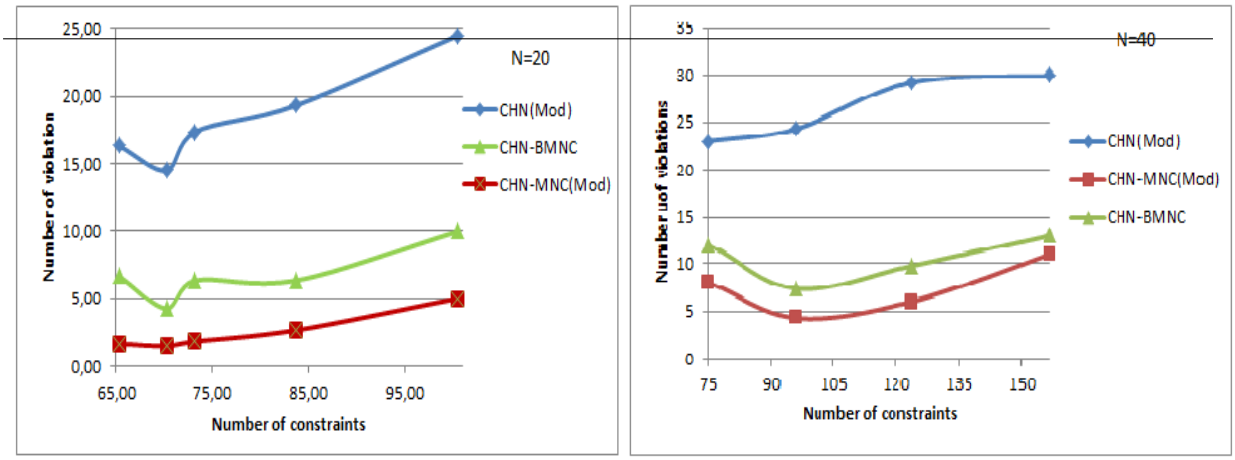


Fig. 8. Comparison Between the most values founded by CHN-MNC and CHN against CHN-BMN.

11. Chun, A., Wai, H., Wong, R.Y.: Optimizing agent-based meeting scheduling through preference estimation. Engineering Applications of Artificial Intelligence **16** (2003) 727–743
12. Jennings, N.R., Jackson, A.: Agent-based meeting scheduling: A design and implementation. IEE Electronics Letters **31** (1995) 350–352
13. Rasmussen, R.V., Trick, M.A.: The timetable constrained distance minimization problem. In: International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming, Springer (2006) 167–181
14. Wang, J., Niu, C., Shen, R.: Scheduling meetings in distance learning. In: International Workshop on Advanced Parallel Processing Technologies, Springer (2007) 580–589
15. Rinaldi, F., Serafini, P.: Scheduling school meetings. In: International Conference on the Practice and Theory of Automated Timetabling, Springer (2006) 280–293
16. Tsuchiya, K., Takefuji, Y.: A neural network parallel algorithm for meeting schedule problems. Applied Intelligence **7** (1997) 205–213
17. Hopfield, J.J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of the national academy of sciences **81** (1984) 3088–3092
18. Hopfield, J.J., Tank, D.W.: neural computation of decisions in optimization problems. Biological cybernetics **52** (1985) 141–152
19. Haddouch, K., Ettaouil, Mohamed Loqman, C.: Continuous hopfield network and quadratic programming for solving the binary constraint satisfaction problems. Journal of Theoretical & Applied Information Technology **56** (2013) 362–372
20. Haddouch, K., Ettaouil, M., Loqman, C.: Continuous hopfield network and quadratic programming for solving the binary constraint satisfaction problems. Journal of Theoretical & Applied Information Technology **56** (2013) 362–372
21. Minton, S., Johnston, M.D., Philips, A.B., Laird, P.: Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. Artificial Intelligence **58** (1992) 161–205
22. Handa, H.: Hybridization of estimation of distribution algorithms with a repair method for solving constraint satisfaction problems. In: Genetic and Evolutionary ComputationGECCO 2003, Springer (2003) 991–1002
23. Hayakawa, D., Mizuno, K., Sasaki, H., Nishihara, S.: Improving search efficiency adopting hill-climbing to ant colony optimization for constraint satisfaction problems. In: Knowledge and Systems Engineering (KSE), 2011 Third International Conference on, IEEE (2011) 200–204
24. Liu, T., Liu, M., Zhang, Y.B., Zhang, L.: Hybrid genetic algorithm based on synthetical level of resource conflict for complex construction project scheduling problem. In: Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on. Volume 9., IEEE (2005) 5699–5703
25. Zou, H., Choueiry, B.Y.: Characterizing the behavior of a multi-agent search by using it to solve a tight, real-world resource allocation problem. In: Workshop on Applications of Constraint Programming, Kinsale, County Cork, Ireland. (2003) 81–101
26. Shapen, U., Zivan, R., Meisels, A.: CSPLib problem 046: Meeting scheduling. (http://www.csplib.org/Problems/prob046)
27. Talaván, P.M., Yáñez, J.: A continuous hopfield network equilibrium points algorithm. Computers & operations research **32** (2005) 2179–2196

Multimodal home-healthcare scheduling with interdependent services

S. Lasfargeas[1], A. Sioud[1], C. Gagné[1, *]

[1]. *Université du Québec à Chicoutimi (UQAC)*
555 boul. Université, Saguenay (QC), Canada, G7H 2B1
{Sophie.Lasfargeas1, Aymen_Sioud, Caroline.Gagne}@uqac.ca

Keywords: Home health care, Metaheuristics, Routing and Scheduling, Optimization.

1 Introduction

Health systems in most of the developed countries struggle to solve the complex problem of aging population, growing number of chronic pathologies, hospitals' overcrowding, and risks of nosocomial infections. Home healthcare (HHC) appears to be a viable solution to tackle these economic, organizational and sanitary issues [1]. Indeed, HHC aims to improve patients' living conditions by delivering medical, paramedical and social services while allowing them to stay at their own home. Home environment lets them keep their habits and social bonds. Moreover, patients can be distracted by their family caregivers, reinforcing a feeling of comfort and security.

HHC services are carried out by different skilled caregivers such as nurses, physicians, and therapists. To achieve this, they move from patient to patient, with a particular transportation mode. Hence, many relevant constraints like staff workload constraints and routing constraints may appear [1]. A provided service has a skill requirement that should match with the caregivers' qualifications. To enhance the service quality offered to patients, some HHC providers try to make a trade-off between the preferences of their staff and those of their patients. In this context, operations management is challenging and time-consuming.

In this study, we extend the multimodal home-healthcare scheduling (MHS) problem proposed by [3], with temporal precedence-constraints and synchronized services. These newly added constraints involve the need of focusing on neighborhood structures. Therefore, a new solution approach is presented based on the variable neighborhood search (VNS) algorithm [2]. A comparison of neighborhood structure efficiency is done. Although the best MHS solution method proposed in [3] is a memetic algorithm, our computational results tend to confirm that their crossover used is no longer accurate in a case of precedence constraints and synchronized services. Finally, we are validating the proposed method using two publicly available datasets of the multimodal home-healthcare scheduling and the home healthcare routing and scheduling problem.

2 Related work

Operational research literature has recently been focused on at least the five following HHC related issues: partitioning the territory, dimensioning human resources, allocating resources to districts, assigning operators to patients or visits, scheduling and routing optimization [3]. Particularly, most of the released researches in HHC context has been devoted to assignment problem, scheduling and routing problem, or both. The objective of the assignment problem is to find an ideal pairing of caregivers and patients. The routing problem aims to design caregivers' tours. Analogously, the scheduling problem generates timetables to visit patients [4]. These problems are often solved separately, but obtained solutions are further from optimal [5]. Therefore, it seems intuitively more valuable to consider both. Some models do this in the literature, but very few are tested on publicly available benchmarks.

In [5], they present a detailed formulation of the multimodal home-healthcare scheduling problem. It aims to design a daily planning considering multimodality (i.e. transportation mode) and satisfactions of nurses and patients. Pre-allocated services, assigned to the proper nurses only, are fixed

in advance in MHS (e.g., team meeting, administrative work). The objective function is a weighted sum of normalized hard and soft constraints. Here, invalid solutions and tardy services are permitted. They propose a two stage solving method. On the one hand, an initial solution, either with a random procedure or a constraint programming based approach, is generated. In the case of constraint programming based approach, a constraint model extends a well-known vehicle routing problem with time windows model, see [6]. On the other hand, the solving method improves this initial solution by selecting one of four metaheuristic approaches: general variable neighborhood search (GVNS), simulated annealing hyper-heuristic, memetic algorithm (MA) and scatter search. The best algorithm turns out to be the MA followed by the GVNS. For privacy reasons, they have randomly generated instances based on the real-world ones provided by their partner (Sozial Global AG). In this paper, a heterogeneous team, synchronized services (also called shared visits) and services with precedence relation (e.g., services needing a time distance between them) have not been taken into account.

In [7], they have also proposed a new model formulation, the home healthcare routing and scheduling problem (HHCRSP) with interdependent services. The HHCRSP mathematical model formulation aims to generate suitable daily planning for caregivers. Many HHC characteristics are taken into account including staff workload and qualification constraints, routing constraints, heterogeneous staff, and interdependent services. Specifically, interdependent services can be single or double services. A double service meets the need for services with precedence relation and synchronized services. The objective is to minimize a weighted sum of three performance measures. Moreover, tardy services are allowed to prevent invalid solutions. A new method accommodating interdependent services has been proposed to solve this model. An initial solution is designed promoting urgent services and small distances. Then, an adaptive variable neighborhood search method (AVNS) improves this solution by applying several neighborhoods. Computational results show that the AVNS method outperforms two local search methods in finding optimality or near optimal solutions for instances with up to three hundred patients and forty caregivers in less than four hours. The test instances are purely randomly generated for the experiments. Nevertheless, their work lacks considering the preferences of the caregivers and patients, transportation modes, pre-allocated services and multiple services.

3 Proposal methodology

Our contribution is organized as follows. We first propose a solving method to tackle the MHS problem, incorporating interdependent services. This method has two steps: a greedy algorithm generates an initial solution, which is improved by a VNS algorithm. The proposed model considers significant HHC constraints: temporal precedence-constraint and synchronized services. Afterwards, the underlying neighborhood structures are analyzed. At last, an evaluation of the performance of our method, considering known results on publicly available benchmarks, is done.

Let us define $\mathcal{S} = \{1, \dots, S\}$ the set of services.

Temporal precedence-constraint. Some services (e.g., take a blood sample or administrate drugs), must be done a certain amount of time before providing a meal which has a direct impact on the planning of services. Currently, temporal precedencies of services are taken into account by considering an additional side constraint. They are given in advance for each service as a feature. A precedence-constraint specified for a service s with another service s' is noted as $Pre_s = s'$, where $(s, s') \in \mathcal{S}$. Besides, δ_s^{min} and δ_s^{max} represent the minimal distance time and the maximum distance time between their starting times respectively.

Synchronized services. When two caregivers are needed to perform together, the exact same service to the same patient, it is two simultaneous services. We translate it as two services s and s' with a specific temporal precedence-constraint ($Pre_s = s'$ and $Pre_{s'} = 0$) implying $\delta_s^{min} = 0$ and $\delta_s^{max} = 0$. Synchronized services start when both caregivers have arrived at the patient's home.

Neighborhood structures. Our neighborhood structures are from different moves: a caregiver swap (swaps two caregivers 'tour), a service shift at the best position (within the current tour or in another) and a service swap (within the current tour or in another). Each move between two different caregivers 'tour is performed solely if the concerned caregivers have the same qualification. The service

shift and service swap moves may concern both independent and interdependent services. Otherwise, in a case of services with a precedence constraint, a simple test ensures to comply with it. In a case of synchronized services, both are moved together.

Solution approach. First, an initial solution is generated by a greedy algorithm. The algorithm assigns interdependent services beforehand. It considers hard constraints and prioritizes services depending on their starting time. Secondly, the initial solution is improved by a VNS algorithm with new neighborhood structures, tailored to the precedence-constraints. Soft constraints are taken into account in the objective function, which is a weighted sum of normalized hard and soft constraints. Finally, our VNS is compared to the ones proposed by [3] and [5].

Experimental procedure and future work. We generalized the MHS model by considering new constraints. The efficiency of our model and solution method is currently tested on the datasets of the MHS and HHCRSP models. The next step will be to generate our custom dataset specifying preferences, transportation modes and precedence relations between services. Results of our neighborhood study and tests will be presented.

4 Conclusion

In this extended abstract, we have briefly outlined the main features of HHC problems considered in the literature. We presented some important formulations and their solving methods. We analyzed their limitations considering a large amount of features that need to be studied to comply with HHC providers' quality requirements. The purpose of this contribution is twofold. First, we propose to extend the multimodal home-healthcare scheduling problem to consider some essential HHC features and then, we investigate a new solution method tested on the MHS and HHCRSP available instances. This contribution is a part of a brand new research field; therefore, a comparison with existing models on public datasets would be worthy.

References

[1] Bashir, B., C. Caux, and M. Chabrol. *Literature review in home care*. in *International Conference on Modeling, Optimization and Simulation*. 2012. Citeseer.

[2] Hansen, P., N. Mladenović, and J.A.M. Pérez, *Variable neighbourhood search: methods and applications.* 4OR, 2008. **6**(4): p. 319-360.

[3] Benzarti, E., E. Sahin, and Y. Dallery, *Operations management applied to home care services: Analysis of the districting problem.* Decision Support Systems, 2013. **55**(2): p. 587-598.

[4] Yalcindag, S., A. Matta, and E. Sahin. *Human resource scheduling and routing problem in home health care context: a literature review*. in *Proceedings of 37th Conference on Opisto Research Applied to Health Services, Cardiff*. 2011.

[5] Hiermann, G., et al., *Metaheuristics for solving a multimodal home-healthcare scheduling problem.* Central European Journal of Operations Research, 2015. **23**(1): p. 89-113.

[6] Rossi, F., P. Van Beek, and T. Walsh, *Handbook of Constraint Programming*. 2006: Elsevier Science Inc. 978 pages.

[7] Mankowska, D.S., F. Meisel, and C. Bierwirth, *The home health care routing and scheduling problem with interdependent services.* Health Care Manag Sci, 2014. **17**(1): p. 15-30.

Simultaneous lot-sizing and scheduling in a flow-shop system with energy consideration

Oussama Masmoudi[1], Alice Yalaoui[1], Yassine Ouazene[1] and Hicham Chehade[2]

[1] University of Technology of Troyes, ICD, LOSI (UMR-CNRS 6281), 12 rue Marie Curie, CS 42060, 10004, France
oussama.masmoudi, alice.yalaoui, yassine.ouazene@utt.fr
[2] OptaLP, 2 rue Gustave Eiffel, 10430 Rosières près Troyes
chehade@opta-lp.com

1 Introduction

In this paper, a multi-item multi-level capacitated lot-sizing problem with energy consideration is studied. Lot-sizing deals with determining the quantity to produce of each item at each period, while scheduling is to determine the production sequence in which the products are manufactured on a machine.
Due to the increase of energy costs and the intensification of the global warming, the aim of the manufacturers is to decrease of the production costs while respecting environmental regulations.
For lot-sizing problems in a flow-shop system, Babaei *et al.* [1] proposed a multi-item multi-level capacitated lot-sizing and scheduling problem with sequence-dependent setups, setup carry over and backlogging. Ramezanian *et al.* [6] developed a mixed-integer programming model with availability constraints where the objective consists of minimizing the production, storage and sequence-dependent setup costs. Mortezaei and Zulkifki [5] developed a mixed-integer model for a lot-sizing problem where the objective is to minimize the production, storage and makespan costs. Masmoudi *et al.* [3] considered the energy aspect for the single item lot-sizing problems in a flow-shop system. To our knowledge, it it the first paper dealing with this type of problem.
In this study, a generalized case, with multi-item, is considered. Since this type of problem is known to be NP-hard (Florian *et al.* [2]), a genetic algorithm is developed to provide near optimal solutions in a reasonable time (Mohammadi *et al.*[4]).

2 Problem description

The manufacturing system consists of M machines separated by M buffers with infinite capacity. The horizon of planification is composed of T periods where each one is characterized by a duration L_t, a price of electricity Co_t, a price of power θ_t and an external demand $d_{j,t}$ where j represents the product index and t represents the period index.
The objective is to determine the quantity to be produced by each machine at each period. These entities are determined in a way that minimizes the total cost composed of electricity, power, storage and setup while respecting demand satisfaction, precedence constraints and power limitation.
In order to formulate the model, some assumptions are considered :

- Multi-item are produced in a flow-shop system,
- The external demands are know in advance,
- The capacity of machine is constrained,
- Each machine can produce at most one product at the same time,
- For each period, the demands are satisfied at the end of each period and shortage is not permitted,
- Vertical interaction : A machine m cannot begin to produce a quantity $x_{j,m,t}$ of product j in period t if this quantity is not available at the previous buffer stock,
- The power required by the system is related to the power of machines running in parallel. It should not exceed a defined maximal power.

3 Problem resolution

Florain *et al.* [2] proved that the capacitated lot-sizing problem is NP-hard. For medium and large size problems, the computational times are important and increase significantly. Solvers, as Cplex, are not able to find optimal solutions in a reasonable time especially for large size problems. Therefore, il it recommended to develop approximating methods. A genetic algorithm is developed in order to deal with this problem. This metaheuristic is inspired by the process of natural evolution. It starts with an initial population composed of individuals where each one has its fitness. A selection operation is considered to choose the potential parents to obtain different offsprings. The second generation is obtained after the crossover of the selected parents and mutation.

3.1 Initial population and chromosome representation

The quantities produced by each machine, at each period and for each product are generated randomly. As the initial population can affect the performance of the genetic algorithm, we use a heuristic proposed by Mohammadi *et al.*[4] described as follows:
for t=1 to T:
1- The items are sorted in the decreasing order of

$$W_{j,m} = \sum_{i=1}^{N} W_{i,j,m} \quad \forall j = 1, ..., N \tag{1}$$

2- Let [i] corresponds to the i th item in ordered sequence in this heuristic.
for [i]=1 to N:
(a) Let insert the item [i] into every position.
(b) Compute the sum of setup costs for the each alternatives.
(c) Place item i in the position with the lowest sum of setup costs.

In this heuristic, $W_{i,j,m}$ corresponds to the setup cost of machine m from production of item i to j and N corresponds to the number of different items.

The chromosomes are presented in a matrix format with 4 lines and $N \cdot M \cdot T$ columns. The lines represent the produced quantities, the sequence of the production of items, the starting times and finally the completion times of production. A sample example (for the sequence) with T=2, $M = 2$ and $N = 3$ is shown in figure 1.

We start with a first scenario which consists of scheduling items without having overlaps between the operating machines.

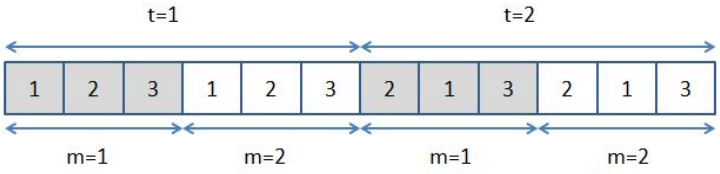


Fig. 1. A sample chromosome of production sequence

3.2 Fitness function

To classify the quality of the generated chromosomes, we compute the fitness of each one that is equal to the objective function. When the obtained chromosomes present non feasible solutions, a penalty function, using additive form, is considered (for more details, we refer the readers to Yeniay [7] .

3.3 Crossover and mutation operators

As a first step, the two-point crossover and the swap period positions operators are considered.

3.4 Population updating

After the crossover and mutation procedures, each couple of parents provides two offsprings. The size of the enlarged population is equal to double the initial population ($2 \cdot NB_{pop}$). The selected chromosomes for the next generation are the best NB_{pop} in terms of fitness.
These procedures are repeated for a defined number.

4 Computational experiments

In order to evaluate the performances of the proposed approximating method, a first testing campaign is considered. In table 1, some instances are presented. This table shows the objective functions obtained by Cplex, the corresponding computational time CPU, the objective functions obtained by the Genetic Algorithm AG, the corresponding CPU and finally the gap between these methods.

Instance (N,M,T)	Z_{Cplex}	CPu_{Cplex} (s)	Z_{AG}	CPu_{AG} (s)	$Gap_{AG}(\%)$
(2,2,2)	133.89	1	147.49	< 1	10.15
(2,4,2)	233.53	1	235.36	< 1	0.78
(2,4,4)	334.19	1	390.26	< 1	16.77
(4,4,4)	232.37	7	292.27	1	25.77

Table 1. Comparaison of exact and heuristic results

The results presented in this table show that, for some instances, near optimal solutions are obtained. The tuning of the parameters could also improve the solutions. Therefore, more tests will be considered.

5 Conclusion

In this paper a multi-item lot-sizing problem in a non-permutation flow-shop with energetic aspects is studied. Due to the problem complexity, it is appropriate to develop approximating methods. In this paper, a first part of the research is presented. In order to improve the quality of the solutions, we are currently testing other alternatives as considering other crossover and mutation operators and establishing more tests to better set the parameters of the proposed algorithm.

References

1. M Babaei, M Mohammadi, and S.M.T FatemiGhomi. A genetic algorithm for the simultaneous lot sizing and scheduling problem in capacitated flow shop with complex setups and backlogging. *The International Journal of Advanced Manufacturing Technology*, 70(1-4):125–134, 2014.
2. M. Florian, J. Lenstra, and A.R. Kan. Deterministic production planning : algorithms and complexity. *Manage Sci*, pages 669–679, 1980.
3. Oussama Masmoudi, Alice Yalaoui, Yassine Ouazene, and Hicham Chehade. Lot-sizing in flow-shop with energy consideration for sustainable manufacturing systems. *IFAC-PapersOnLine*, 48(3):727 – 732, 2015.
4. Mohammad Mohammadi, S.M.T. Fatemi Ghomi, and Niloofar Jafari. A genetic algorithm for simultaneous lotsizing and sequencing of the permutation flow shops with sequence-dependent setups. *International Journal of Computer Integrated Manufacturing*, 24(1):87–93, 2011.
5. Navid Mortezaei and Norzima Zulkifli. Integration of lot sizing and flow shop scheduling with lot streaming. *Journal of Applied Mathematics*, 2013, 2013.
6. Reza Ramezanian, Mohammad Saidi-Mehrabad, and Parviz Fattahi. Mip formulation and heuristics for multi-stage capacitated lot-sizing and scheduling problem with availability constraints. *Journal of Manufacturing Systems*, 32(2):392–401, 2013.
7. Ozgur Yeniay. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications*, 10:45–56, 2005.

Driven Workflow Orchestration of Patient pathway in Hospital Emergencies

S. BEN OTHMAN and S. Hammadi

Ecole Centrale de Lille, CRIStAL Laboratory
Cité scientifique, 59650 Villeneuve d'Ascq
France
{sara.ben-othman, slim.hammadi}@ec-lille.fr

Abstract: Since the heat wave of the summer of 2003 in France, regional and national commitments to face the struggle against the peaks of activity and the overcrowding in health care systems have multiplied in order to guarantee a better patient health care. Our objective in this paper is to identify the overcrowding situations through key indicators, model patient pathway using workflow approach and finally offer agent-based decision support solutions for managers to provide better control of activity. In case of uncertainties in hospital emergencies, a coalition of agents is formed to collaborate and negotiate to provide decisions concerning the dynamic orchestration of workflow and minimize the waiting time of patients during their care. This work is a part of the project HOST (Hôpital: Optimisation, Simulation et évitement des Tensions), funded by the French National Research Agency (ANR).

Keywords: collaborative workflow, orchestration, health care systems, overcrowding, multi-agent systems, coalition.

1 Introduction

Hospital emergencies are facing problems related to the increasing demand in their services. Patients arrive to the services randomly and their flow keeps growing. The management of patients is carried out according to different modes related to the type of care required [1]: planned care requiring or not hospitalization, unscheduled urgent and non-urgent health care operations. One of the major problems faced by hospitals is due to the permanent interference between the planned activities, non-planned health care activities and especially urgent and non-planned care activities. The emergency term involves two different phenomena:

- Recurring flows which may present seasonal variations: Even if these flows are apprehended, quantified in volume and nature, the establishment of a management organization and a short-term control is an issue, leading to an efficient healthcare production system.
- Random arrival flows: In this case, the flows are completely unexpected in volume and in nature. If it is possible to adapt and implement traditional and organizational optimization methods for health care tasks scheduling, it is much more complex to control patient flows at emergencies [2].

The problem is not easy in a context where the emergencies are considered by the actors in the health system as the main care system entropy generator. Patients want the best care at the best price. These requirements have produced an interest in optimizing technical and human resources while mastering the costs [3]. Particularly, Pediatric Emergency Departments (PED) have limited resources, in addition to their sensitive and critical environment, joint with acuity. Logistics in health care institutions is therefore needed to improve efficiently the organization by reducing costs and meeting the needs of the different actors. The main challenge of this work is use some tools and optimization approaches in order to manage and anticipate the overcrowding situations in the PED. The experimental field of our study is the PED of the Regional University Hospital of Lille (CHRU de Lille), where patients wait for long hours (sometimes more than 10 hours) before treatment. These long delays can endanger patients' lives. This phenomenon highlights the need to review the emergency management process and the implementation of measures

to preserve the quality of care for patients. In cooperation with the University Hospital of Lille (CHRU de Lille), in order to meet these challenges, we are interested in the development and the implementation of a decision support system to model the architecture of the PED and optimize its operation. This tool offers a refined model for the logistical needs in emergency conditions, taking into account the existing operation.

Our paper is organized as follows: a state of the art about health care management will be presented in the second section. In Section 3, we describe some approaches to improve the functioning of emergencies. The proposed approach is presented in both of the sections 4 and 5. Performance evaluation is given in section 6 followed by simulation and results in section 7. This article is closed with conclusions and an outlook to further work.

2 State of the art

2.1 Health care systems management

The analysis of health care institutions dysfunctions shows that these are due in part to an organization poorly adapted to the constraints, an evolution of their missions and a poor management of patient flow [4] [5] [6]. Thus, the optimization of management and information systems are important levers for the development of these organizations. Therefore, the implementation of strategic, tactical and operational monitoring systems has become essential. To manage the transition, it is necessary to define the new organizational paradigms, new trades for the governance and management of these new organizations and support mechanisms. Reconfiguring and improving the health care system require "reorganization." These are problems whose complexity calls for innovative approaches. Today, emergencies in hospitals have a strategic position in modern healthcare systems and represent the main gateway to the hospital [7] [8]. This key role will be strengthened in the coming years due to the continuous growth in the arrivals of patients who are increasingly demanding. These changes generate many problems to different actors in public health, including operating problems to cope with the increase in consultations and the high cost of health expenditures [9] [10] [11]. However, if the majority of health care systems are facing these problems, the way to deal with them differs from one organization to another [12]. Thus, to better understand the problems and the current challenges, it is important to model the health care activities in the emergencies in order to identify the dysfunctions in patients' pathways and health care processes.

2.2 Strategies for improving the functioning of hospital emergencies

Different strategies have been proposed to improve the patient journey at the emergencies including the change of work schedule, human resources and their roles adjustments and modification of health care management procedures in order to reduce patients' waiting time.

2.2.1 Adjustment of human resources

Although the change of the schedule of the medical staff or the increase of human and material resources can improve the capacity of the emergencies, they may not be achievable in most existing emergencies because of space restrictions and/or budgetary constraints. Alternative methods try to maximize the use of the existing medical staff. The adjustment of human resources includes the planning of the medical staff workload for day shifts as well as the changes in the schedule over a longer period such as one month. The workforce management adjustments can better match available human resources with the different requirements of the fluctuation in arrivals of patients. The observed results show that the pattern of the daily demand is constantly at a low level during the night, and develops a peak at midday [13]. For example, the number of doctors can be programmed to ensure maximum coverage corresponding to the peak demand. The other method for the rescheduling of the medical staff considers more time slots in accordance with the preferences of the medical staff [14].

2.2.2. The adjustment of the roles of the emergencies actors

The "role" of the medical staff can be adjusted to improve the efficiency of the functioning of the emergencies [15]. It is suggested that each medical staff member must mainly perform care tasks adapted to his skills, and be affected according to a required level of skills. The priority to assign a health care task to a medical staff depends on the skills level of this latter. For example, the list of priorities helps to ensure that medical specialists mainly focus on advanced problems and diagnoses rather than on the basic procedures that other doctors are able to perform.

2.2.3 Patient journey improvement

Hospitals can improve their efficiency by improving patient journey. A strategy based on creating pathways for outpatient care was first introduced to speed up the treatment of patients with non-urgent care operations [16]. Five alternative scenarios show that the implementation of the fast track to patients who need minor care could include reducing overall patient length of stay. Research indicates that 85% of emergency requests are due to non-urgent patients or patients with non-serious conditions [17]. Patient flow can be accelerated if the low acuity patients can be evacuated quickly. Many studies have reported that the establishment of a Rapid Care Unit (RCU) can reduce the waiting time of patients [18]. A team dedicated to patients sorting in the RCU can reduce the waiting time of patients and the length of stay in the emergencies [19].

3 Approaches to improve the functioning

Simulation is one of the most used tools in operational research. It has been used to identify potential areas for improvement through an analysis of the various surveys related to health care systems [20] [21]. Simulation can help the understanding of optimal allocation of resources and their use, and the estimation and evaluation of existing and proposed systems. The simulation provides managers of health institutions with a tool to evaluate system performance improvement methods. It can help to understand the optimal allocation of resources, and the estimation and the evaluation of existing and proposed systems [22].

Different research and improvement strategies have been proposed to reduce the overcrowding phenomenon in health care institutions [23] [24]. Previous studies indicate that patients at emergencies spend most of their time waiting for treatment due to an imbalance between resources and requirements [25]. Available improvement strategies include modifying the work plan of the emergencies, the adjustments of human resources and changes in treatment procedures. As the PED is a complex system because of the random flow of arrivals of patients, the uncertain time management and the randomness of the decision, it is difficult to change the treatment protocol without disrupting normal operating procedures. Inspired by the field of industry, different tools and techniques such as workflow were therefore used to model the flow of patients for health care systems' management. These methods have contributed to improve the performance of the service [26]. The workflow was used within emergencies to test their activities and help managers to analyze their adopted management process and the performance of their health facility in order to identify the causes of long waiting times for patients [27].

3.1 Health care workflow

Health care workflow is an approach to model and manage medical and administrative activities in a health care institution, involving several actors, documents and tasks. It consists of working models for coordinating the activities of each medical staff and ensures their interconnection relying on information systems and existing databases.

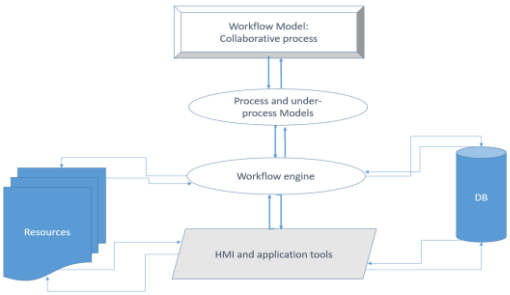


Fig. 1. Phases of Workflow management

In the literature there are several definitions of the workflow approach that most distinguishes documents, information and tasks. This distinction is useless to a data standpoint, since documents and tasks are part of the information, especially in a health information system. We prefer the definition set out in [28]: "The management of a workflow is achieved by a proactive system for the management of a series of tasks that are transmitted to the appropriate participants in the correct order and are supplemented in given time ..."

We consider two main steps in the management of a workflow. The first concerns the design and process definition; and the second is the control and management of process execution. Figure 1 illustrates the relationship between the two main stages.

3.2 Agent-based Systems (ABS)

According to the Oxford dictionary, a behavior is the way in which a person or an object responds to a particular situation. In modeling the emergencies, behaviors related to patients and medical staff can be modeled to monitor and assess the evolution of patient flow in order to reduce waiting time. Patients, nurses and doctors are considered as dynamic and decision-making entities, called agents. Therefore, a model of ABS consists of three main entities: 1) agents, their characteristics and behaviors 2) the relationship between agents, methods and trading results and interaction and 3) the environment of agents or the agent which is in charge of managing agents' coalitions to achieve a specific goal. An ABS uses intelligent and interactive agents to make decisions [29]. An agent can be described as an autonomous entity that makes decisions based on a set of rules. The agent also communicates with other agents in the system and can adapt and change its behavior based on the results of its interactions with its environment [30]. Therefore, these entities are proactive, autonomous and intelligent in a model of ABS. Management rules can be created and applied for the decision. The basic rules are the individual conduct of an agent; as secondary rules can be added to change the ground rules for the different interactions of agents in the system. Agents can also learn from past impacts of their actions on their environments.

The ABS are used to model the emergencies to evaluate the performance with different parameters [31]. The study concluded that the number of patients treated can be increased by approximately 26% following the patient sort adjustment. Optimization techniques can be used with the ABS for optimum configuration of emergencies staff. Another example is the ABS modeling scenarios involving considerable human decision making [32]. A strategy called re-sorting has been tested in which patients who have waited a long time in the waiting room are re-sorted. This research used the ABS to model patients with dynamic attributes and behavior triage nurses to determine the evolution of patient's health. The results show that the re-sorting strategy can improve the average waiting time in the emergency room.

4 Agent oriented approach for collaborative workflow

ABS are provided with autonomy and reactivity. Under the term "ABS for workflow" we gather every distributed approach for executing workflow that involves cooperative entities. Health collaborative workflow models are equipped with workflow execution features in a distributed manner. In fact, ABS are composed of distributed entities (many services at PED), capable of communicating, which may influence the development of process execution. These entities have a specific condition which is not perceived by other entities. Interaction between different entities may generate new solutions for optimization problems, such as, bringing a medical deed to a satisfactory conclusion.

Workflow agents can be classified into three categories:

- Agents who cooperate at the service of patients. Here, each agent plays a similar role to a physical agent (doctor) at PED, to improve patient care.
- Distributed agents, responsible for the reactive coordination of care tasks. They are based on activities, not on roles. Their coordination is managed by a workflow diagram, without a central execution engine.
- Agents going from one "service point" to another. For example, a doctor agent may migrate from a care team to another team, in favor of the patient having priority.

5 Agent coalition for executing collaborative workflow instances

Collaborative workflow allows to model in a realistic way the patient's position and movements between the PED sites. Because of the unpredictable nature of the care activity and uncertain environment of the PED that require dynamic information management, the formation of coalitions of agents is necessary to solve the health care planning problem, the management of resources and the supervision of the Workflow.

The logic of a coalition is based on agents acting as cooperating actors. Each agent of the coalition is able to monitor the execution of a workflow instance. An instance is composed of operators defining constraints (sequence, parallelism, etc.) and conditions to create complex care tasks. These care tasks can be nested together at all levels [33]. The agents of a coalition must conclude agreements between them to decide whether they need services provided by other agents of another coalition in favor of a health condition of one or more patients. These agreements are called "negotiation". A negotiation protocol was proposed, allowing agents to provide appropriate decisions and set their execution conditions (single or repeated execution of a workflow instance, as signment of medical staff, etc.).

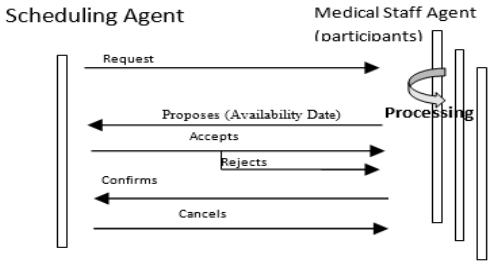


Fig. 2. Negotiation protocol between agents

The negotiation strategy explicitly uses two knowledge bases: the first is declarative for describing the objectives and the context of the negotiation (e.g. adapt the treatment room depending on the health state of a patient). The second includes basic negotiation rules, such a strategy of medical staff assignment for a good quality of health care treatment [43]. The objective is to define a negotiation protocol of agents' coalition for distributed execution of a workflow instance. We propose a two-step approach for the management of a collaborative workflow. The interest is to separate in the model thanks to a coalition of agents, a first phase of assignment and scheduling of health care tasks, which will then be used by the Scheduling Agent (SA) to monitor a second phase of dynamic orchestration, based on a process of negotiation between the coalition of agents (e.g. SA and medical staff agents) [34]. The idea is to separate coalitions depending on the nature (predictability) of knowledge they handle. The second phase is the scheduling and tasks assignment phase in which the SA analyzes the description of a workflow scheme. This scheme defines the inherent properties complying with treatment protocols, set before execution. It includes specifications on existing resources, a list of tasks to be performed, and the links between the precedence tasks. The dynamic orchestration stage is based on a dynamic scheduling methodology. In addition to the time constraints, the availability of resources is also taken into account. During this stage, tasks are scheduled according to priorities calculated while running the workflow. The SA solves the problem of tasks scheduling through an Evolutionary Algorithm (EA). It calculates the starting date of execution of health care tasks regarding the availability of medical staff members and their skills.

Table 1: Assignment: Example of a chromosome pattern characterized by {0, 1, *}

$S^{ch} = \{ S^{ch}_{i,j,k} / 1 \le j \le N ; 1 \le i \le n_j ; 1 \le k \le M \}$					
		MS1	MS2	MS3	MS4
T 1	O 1 ,1	0	*	*	0
	O 2 ,1	0	1	0	0
	O 3 ,1	0	*	*	*
T 2	O 1 ,2	*	0	0	*
	O 2 ,2	0	0	1	0
	O 3 ,2	1	0	0	0
T 3	O 1 ,3	*	*	0	*
	O 2 ,3	*	*	*	0

The value " $S^{ch}_{i,j,k} = 0$" indicates that the medical staff member MS_k is not enough qualified for this health care operation, " $S^{ch}_{i,j,k} = 1$" indicates that the assignment of the operation $O_{i,j}$ to the medical staff member MS_k is *obligatory* because he is the only one qualified for this task and the symbol: " * " indicates that the assignment is *possible.*

This allocation scheme (Table 1) becomes the chromosome model of our EA. This chromosome is well adapted to the properties and the constraints of the problem and is used in the construction of individuals in order to integrate good properties and good performance of medical protocols. Having a valid chromosome pattern, EA are more efficient and faster by building solutions promoting the reproduction of individuals respecting the right patterns. In the case of our scheduling problem, the difficulty of the implementation of this technique is important because it needs to develop a particular coding that could both describe the problem

data and use schema theory. This chromosome pattern covers the entire interesting possibilities to affect the care operations. However, it has prohibitions that can sometimes be costly in terms of workload of health care staff. However, assignments belonging to this model do not guarantee an optimal solution in terms of workload balancing. Some assignments can be spread by EA when they do not lead to good solutions. Indeed, during the phase of reproduction, whether the individual violating the chromosome pattern it will be automatically removed. Such an approach allows to restrict the search space that will accelerate the convergence of the algorithm and ensure good quality of solutions.

6 Performance evaluation of dynamic orchestration

The assessment of the orchestration performance is measured by the satisfaction of the waiting time criterion Cr for a currently running workflow.

$Cr = \sum_{j=1}^{N} \max(0, c_j - d_j)$ with:

N: the total number of patients to treat,

c_j: the theoretical completion time of patient P_j treatment,

d_j: the theoretical ending time for P_j treatment.

Indeed, this criterion is evaluated by orchestrating the workflow in a static way (where the order of care tasks remains unchanged during the entire execution) and in a dynamic way. For each type of orchestration, we calculate the average value of the criterion for a number of patients during the simulation period.

Let GCr be the gain corresponding to the criterion Cr for a dynamic orchestration of the workflow $WD(Cr)$ relative to a static orchestration $WS(Cr)$.

$$GCr = \frac{WD(Cr) - WS(Cr)}{|WS(Cr)|} \times 100$$

If $GCr < 0$ then Cr has been enhanced with a dynamic orchestration,

If $GCr > 0$ then Cr has been degraded with a dynamic orchestration compared to a static orchestration,

If $GCr = 0$ then there are no changes in the values of the criterion.

7 Simulations and results

In order to have an objective view on the reasons for bottlenecks, databases provided by the PED list all the patients registered in the emergencies, their pathologies their emergency degree, their length of stay in the service as well as the medical exams that have been allocated to them. From this database we were able to identify some characteristics related to the patient journey in the PED such as the annual number of visits which reached 25 000, the number of times the patient's location was changed during his stay at the PED (5.5 times) and the average waiting time of patients which is a key data for the analysis of the overcrowding phenomenon, it is about 4 hours, including additional examinations and awaiting the results.

7.1 Patient pathway description of the PED (CHRU de Lille)

Patient pathway at a PED may have 2 types of entries:

- Emergencies: it is the entry mode for the majority of patients, whether they have come by their own means (parents, family, etc.) or via an ambulance which may not have informed the hospital of its arrival. This entry is common to all the persons entering the CHRU through Emergencies, children or adult. Parents or other family members must follow the administrative admission procedure before being redirected to PED.
- SAMU1 : the arrival of a patient is done directly on the ambulance entry platform, the patient is immediately redirected to PED, while a family member will be sent to the administrative office in order to complete hospital admission files.

[1]NT: Ambulance Emergency Services

If the patient has entered the hospital via Emergencies, the moment he arrives to PED, he undergoes a first diagnose established by a nurse, which determines priority order to access services. This priority order depends on a 5 level scale for immediately treating the most severe cases. These emergency levels are represented by the CCMU (Classification Clinique des Malades aux Urgences). After a more or less important waiting time, depending on service saturation and emergency condition, the patient is finally treated by a doctor.

After the medical diagnose established by a doctor, he may choose –in function of the medical condition and available infrastructure- to let the patient go back home or to keep him in observation, even to hospitalize him. During observation, the patient is practically constraint to stay in hospital for a short time at PED because of the lack of top-down structures allowing for reception. In fact, there generally are no available beds under ongoing supervision at a hospital. What is more, there is no central patient alarm. Equally, if the doctor decides to hospitalize the patient for a period longer than 12 hours (maximum short hospitalization stay), he is obliged to keep the patient at PED waiting for a bottom-up structure to be available. Finally, patient reorientation to less charged hospital structures, if that is available, are very rare, due to the lack of communication between hospitals, and the impossibility to know the overcrowding level of their structures.

7.2 Collaborative workflow for patient circuit under overcrowding state

The specificity of PED functioning under overcrowding state has led us to define a new patient pathway which considers the possibility of using resources which have not been assigned to tasks, because doctors can relocate them. These resources mainly concern service rooms: we focus on this aspect for redefining patient pathway. In this way the entry -which has so far remained administrative- does not change at all as regards permanent modeling. In fact, it is important to correctly register the patient according to administrative and legislative criteria. As regards the exit, this cannot be modified because there is no reason for reoriented patients to come back to the service. On the other hand, exits are, as entries, purely formal administrative stages which imply purely external bureaucratic services and which serve to ensure correct patient monitoring, from admission to exit. These services enable the hospital to be prepared in case of conflict with a patient, following complications in the patient's condition.

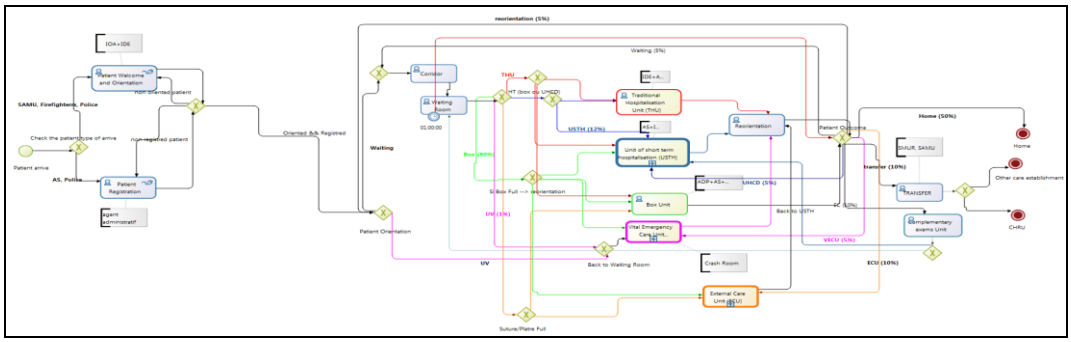


Fig. 3. PED process modeling under overcrowding state

Each service (simple consultation, operation, plaster, short term hospitalization, etc.) can be performed at different rooms. It is then convenient to have classified patients in function of the care they have to be provided for and to separate them again in function of the room where they will receive treatment. This operation multiplies the possibilities of care and makes his pathway more complex. On the other hand, classification must be correctly modeled, because it does not follow explicit rules. Indeed, doctors choose, at the last minute, which room the patient shall be led to. For example, Vital emergency rooms are only used in the last resort, because it is devoted to treating patients whose life is in danger. Evidently, the preferred room will be the one indicated for a special type of care. Then, other rooms are automatically assigned, without particular priority.

7.3 The impact of dynamic orchestration over Workflow

In order to evaluate the impact of dynamic orchestration over Workflow, we have chosen the real scenario of a tense day which took place at PED on February, 7th, 2013. Throughout this day, PED received around 119 patients, registering a remarkable annual activity peak. The difficulty lies in smoothing these activity peaks and reducing

patients' waiting time. To start simulation, we define a patient arrival flow and we calculate the necessary resources for executing care operations, taking into account the skills of medical staff and resource availability.

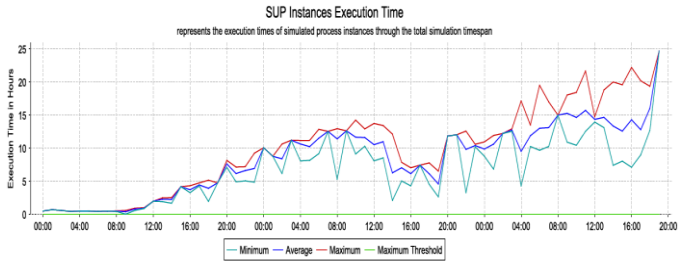


Fig. 4. Average waiting time using only the EA

The results obtained represent curves displaying overall minimal, maximal and average waiting time of patients at PED. In the first place, we introduce the results given by the SA, who is in charge of schedules using an evolutionary approach, with static orchestration. Then, we show that the communication of SA with other agents so as to dynamically orchestrate patient pathway Workflow makes it possible to reduce waiting time. According to Figure 4, we observe that on this overcrowding day, average waiting time differs. Average waiting time is estimated in 10 hours. Medical staff is then incapable of facing the increase in patient flow arriving at PED.

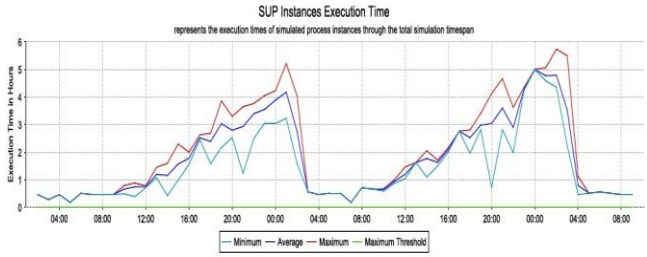


Fig. 5. Average waiting time with dynamic orchestration

Figure 5 shows that the optimization approach based on the alliance between scheduling and ABS approaches has made it possible to smooth up activity peaks at PED and to minimize patient average waiting time. Indeed, the SA plays the role of orchestra director, coordinating his actions with the other agents so as to ensure a dynamic orchestration adapted to the real situation of PED. In this case, waiting time fluctuates between 1 hour and 4 hours. Nevertheless, we observe that there is always a midnight activity peak that disturbs PED.

The previous orchestration actions have not succeeded in absorbing all the PED activity peaks. Because of this, it is necessary to resort to negotiation between the agents of the formed coalition in order to dynamically reorchestrate the Workflow of remaining patients. This negotiation is based on the consideration of additional examination results (biological tests, X-ray, echography, RMI, etc.) for making decisions about the patient's orientation.

8 Conclusion

The use of workflow tools for managing processes at a health establishment is today largely accepted. However, its use in the inter-hospital context bumps into numerous obstacles, for example, the difficulty in transferring a patient file between the different services and the lack of standardization in the interconnection of different local workflows. Workflow tasks are subject to temporary constraints (precedence constraints, deadlines, etc.) and resource constraints (human or machines). Considering the distributed nature of collaborative workflow, unforeseen events may happen at every moment. Thus, the coordination of the execution of tasks must be done in a dynamic manner, in order to neutralize or to reduce the impact of disturbances. Many simulation results have proved the efficiency of the alliance between optimization and ABS. The strength of our system relies on the formation of agent coalitions and their negotiation protocols, in order to make orchestration decisions and thus improve waiting time during patient care, as well as to face the hazards which may come up at PED. However, the use of these approaches in the context of inter-hospital still faces some challenges which will be treated in our future work.

References

[1] DMES (1998). « Classification de l'activité des unités de prise en charge des urgences ». Dossier du Ministère de l'Emploi et de la Solidarité - Direction des Hôpitaux. Mission PMSI, mars 1998.

[2] Chodosas (2002) "Etude de l'organisation des différents systèmes d'urgence dans cinq pays européens : la France, le Royaume-Uni, la Belgique, la Suède, la Finlande et la Norvège".

[3] Navas J-F, Artéta C, Hadjes PS, Jiménez F. Construction et simulation d'un modèle de flux de patients dans le service d'urgences d'un hôpital colombien, « Formato de Atencion Inicial de Urgencias, en Procesos Operativos de Urgencias », Documentos confidenciales Servicio de Urgencias Fundacion Cardiolnfantil. Agosto; 2003.

[4] Ducq Y, Vallespir B, Doumeingts G. Utilisation de la méthodologie GRAI pour la modélisation, le diagnostic et la conception d'un système hospitalier. 2ème conférence francophone en Gestion et Ingénierie de Systèmes Hospitaliers (GISEH). Mons; 2004.

[5] Colombier G. 50 propositions pour une meilleure prise en charge des Urgences médicales. N.p., 2007. Print.

[6] Gentil, Stéphanie. « Les « agencements organisationnels » des situations perturbées : la coordination d'un bloc opératoire à la pointe de la rationalisation industrielle ». *Communiquer. Revue de communication sociale et publique* 8 (2013): 65-80. *communiquer.revues.org*. Web.

[7] Kadri, Farid et al. « Time Series Modelling and Forecasting of Emergency Department Overcrowding ». *Journal of Medical Systems* 38.9 (2014): 1-20. *link.springer.com*. Web.

[8] Bellou, A et al. « Place des services d'Urgences dans la régulation des hospitalisations publiques ». *La Revue de Médecine Interne* 24.9 (2003): 602-612. *ScienceDirect*. Web.

[9] Cooke, Matthew (Professor of clinical systems design) et al. *Reducing attendances and waits in emergency departments : a systematic review of present innovations*. [London]: [National Co-ordinating Centre for NHS Service Delivery and Organisation R & D (NCCSDO)], 2004. *wrap.warwick.ac.uk*. Web. 25 sept. 2015.

[10] Pines, Jesse M., Joshua A. Hilton, et al. « International Perspectives on Emergency Department Crowding ». *Academic Emergency Medicine: Official Journal of the Society for Academic Emergency Medicine* 18.12 (2011): 1358-1370. *PubMed*. Web.

[11] Schuur, Jeremiah D., et Arjun K. Venkatesh. « The Growing Role of Emergency Departments in Hospital Admissions ». *New England Journal of Medicine* 367.5 (2012): 391-393. *Taylor and Francis+NEJM*. Web.

[12] Pines, Jesse M., Sanjay Iyer, et al. « The Effect of Emergency Department Crowding on Patient Satisfaction for Admitted Patients ». *Academic Emergency Medicine: Official Journal of the Society for Academic Emergency Medicine* 15.9 (2008): 825-831. Print.

[13] McCaig, Linda F., et Catharine W. Burt. « National Hospital Ambulatory Medical Care Survey: 2002 Emergency Department Summary ». *Advance Data* 340 (2004): 1-34. Print.

[14] Carter, Michael W., et Sophie D. Lapierre. « Scheduling Emergency Room Physicians ». *Health Care Management Science* 4.4 (2001): 347-360. *link.springer.com*. Web.

[15] Hay, A.M., E.C. Valentin, et R.A. Bijlsma. « Modeling Emergency Care in Hospitals: A Paradox - The Patient Should not Drive the Process ». *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*. N.p., 2006. 439-445. *IEEE Xplore*. Web.

[16] Devaraj, Sarv, Terence T. Ow, et Rajiv Kohli. « Examining the impact of information technology and patient flow on healthcare performance: A Theory of Swift and Even Flow (TSEF) perspective ». *Journal of Operations Management* 31.4 (2013): 181-192. *ScienceDirect*. Web.

[17] McGuire, F. « Using simulation to reduce length of stay in emergency departments ». *Simulation Conference Proceedings, 1994. Winter*. N.p., 1994. 861-867. *IEEE Xplore*. Web.

[18] Wiler, Jennifer L. et al. « Optimizing Emergency Department Front-End Operations ». *Annals of* Emergency Medicine 55.2 (2010): 142-160.e1. PubMed. Web.

[19] Walley, P. « Designing the Accident and Emergency System: Lessons from Manufacturing ». *Emergency Medicine Journal* 20.2 (2003): 126-130. *emj.bmj.com*. Web.

[20] Mahdavi, Mahdi et al. « Generic operational models in health service operations management: A systematic review ». *Socio-Economic Planning Sciences* 47.4 (2013): 271-280. *ScienceDirect*. Web.

[21] Khare, Rahul K. et al. « Adding More Beds to the Emergency Department or Reducing Admitted Patient Boarding Times: Which Has a More Significant Influence on Emergency Department Congestion? » *Annals of Emergency Medicine* 53.5 (2009): 575-585. *PubMed*. Web.

[22] Barrick, Ivan J. Transforming Health Care Management: Integrating Technology Strategies: Integrating Technology Strategies. Jones & Bartlett Publishers, 2009. Print.

[23] Zeng, Zhen et al. « A Simulation Study to Improve Quality of Care in the Emergency Department of a Community Hospital ». *Journal of Emergency Nursing* 38.4 (2012): 322-328. *ScienceDirect*. Web.

[24] Abo-Hamad, Waleed, et Amr Arisha. « Simulation-based framework to improve patient experience in an emergency department ». *European Journal of Operational Research* 224.1 (2013): 154-166. *ScienceDirect*. Web.

[25] Peck, Jordan S. et al. « Predicting Emergency Department Inpatient Admissions to Improve Same-Day Patient Flow ». *Academic Emergency Medicine: Official Journal of the Society for Academic Emergency Medicine* 19.9 (2012): E1045-1054. *PubMed*. Web.

[26] Deelman, Ewa et al. « Workflows and e-Science: An overview of workflow system features and capabilities ». *Future Generation Computer Systems* 25.5 (2009): 528-540. *ScienceDirect*. Web.

[27] Ines Ajmi, Hayfa Zgaya, Lotfi Gammoudi, Slim Hammadi, Alain Martinot, Régis Beuscart, Jean-Marie Renard, « Mapping patient path in the Pediatric Emergency Department: A workflow model driven approach", Journal of Biomedical Informatics 54 (2015) 315–328.

[28] Hales, Keith, Mandy Lavery, et Ovum Ltd. *Workflow Management Software: The Business Opportunity*. Ovum Ltd, 1991. Print.

[29] Stormer, Henrik. Ein flexibles und sicheres agentenbasiertes Workflow-Management-System. N.p., 2003. Print.

[30] Kamath, Mohan Umesh. Improving Correctness And Failure Handling In Workflow Management Systems. N.p., 1998. Print.

[31] Zheng, Kai et al. « Quantifying the impact of health IT implementations on clinical workflow: a new methodological perspective ». *Journal of the American Medical Informatics Association : JAMIA* 17.4 (2010): 454-461. *PubMed Central*. Web.

[32] Macal, Charles M., et Michael J. North. « Agent-based Modeling and Simulation: ABMS Examples ». *Proceedings of the 40th Conference on Winter Simulation*. Miami, Florida: Winter Simulation Conference, 2008. 101–112. *ACM Digital Library*. Web. 25 sept. 2015. WSC '08.

[33] Escudero-Marin, Paula, et Michael Pidd. « Using ABMS to Simulate Emergency Departments ». *Proceedings of the Winter Simulation Conference*. Phoenix, Arizona: Winter Simulation Conference, 2011. 1239–1250. *ACM Digital Library*. Web. 25 sept. 2015. WSC '11.

[34] Ben Othman S., Hammadi S., Quilliot A., Martinot A., Renard J.M., Health Care Decision Support System for the Pediatric Emeregency Department Management, in proceedings of The 15th World Congress on Health and Biomedical Informatics (MEDINFO'2015) (2015-b)São Paulo, Brésil.

Metaheuristics and big data

Metaheuristic Based Clustering Algorithms for Biological Hypergraphs

Boonyarit Changaival[1], Gregoire Danoy[1], Marek Ostaszewski[2], Kittichai Lavangnananda[3], Franck Leprevost[1] and Pascal Bouvry[1]

[1]Computer Science and Communications Research Unit, University of Luxembourg, Luxembourg City, Luxembourg
boonyarit.changaival@uni.lu, gregoire.danoy@uni.lu, pascal.bouvry@uni.lu

[2]Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Belval, Luxembourg
marek.ostaszewski@uni.lu

[3]School of Information Technology ,King Mongkut's University of Technology Thonburi, Bangkok, Thailand
kitt@sit.kmutt.ac.th

Keywords: combinatorial optimisation, linear programming.

1. Introduction

Hypergraphs are widely used for modeling and representing relationships between entities, one such field where their application is prolific is in bioinformatics [1] and [2]. In the present era of big data, sizes and complexity of these hypergraphs grow exponentially, it is impossible to process them manually or even visualize their interconnectivity superficially. A common approach to tackle their complexity is to cluster similar data nodes together in order to create a more comprehensible representation. This enables similarity discovery and hence, extract hidden knowledge within the hyper graphs. Several state-of-the-art algorithms have been proposed for partitioning and clustering of hypergraphs. Nevertheless, several issues remain unanswered, improvement to existing algorithms are possible, especially in scalability and clustering quality. This article presents a concise survey on hypergraph-clustering algorithms with the emphasis on knowledge-representation in systems biomedicine. It also suggests a novel approach to clustering quality by means of cluster-quality metrics which combines expert knowledge and measurable objective distances in existing biological ontology.

2. Hypergraph-based knowledge representation and clustering: the Parkinson Disease case study

Knowledge representation using graph formalism is often achieved by encoding entities as graph nodes and relations between them as graph entities. A research domain greatly benefiting from such an approach is systems biomedicine, where entire (patho)biological processes are perceived as systems, and information about them is encoded as graphs. However, graph representation is insufficient for complex relationships between multiple elements, for instance enzyme-catalyzed biochemical reactions. In such situations the notion of graph has to be generalized to a hypergraph. For instance, Open Biological Expression Language (OpenBEL) format [3], used for biomedical knowledge representation, is based implicitly on hypergraph formalism.

A hypergraph is defined as H = (V, X), where V is a set of all vertices in the graph and X is a non-empty subset of vertices in V which are named as a hyperedges. We illustrate our problem by the hypergraph created from curated Parkinson disease (PD) knowledge from literature. A biomedical resource, called PD map [4] is available in OpenBEL format. Each OpenBEL statement contains the ID of biological entities (proteins, genes, etc.) and their relationships (catalyticActivity, directlyIncreases, etc.). By connecting entities in the same statement based on their relationships and/or connecting different statements together, the result is a hypergraph where subsets in E are connected by an arc (in this case, for simplification, the orientation of the arcs are not considered and thus treated as edges). By utilizing the given information in each statement, we extend the notation of the hypergraph by attributes, defining as H = (V, X, E, A_V, A_E) where E is set of all edges (or arc), A_V is introduced as the set of all possible attributes of the vertices in V and A_E is the set of all possible attributes of the edges in E. The example graph is shown in figure 1. Each shaded area represent hyperedge and one hyperedge can be a part of another hyperedge or connect to another hyperedge. Let us highlight that the PD map hypergraph is sparse continues to grow at an alarming rate as more is known about the parkinson-disease is known. As commonly known, graph-clustering problem is NP -Hard [2] and [5], meta-heuristic is a suitable candidate for providing possible satisfactory solutions.

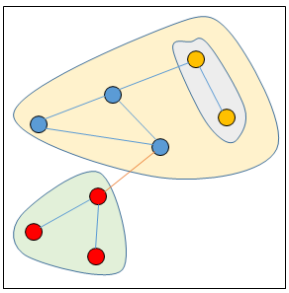


Figure 1 Hypergraph

3. Related Work

A hypergraph can be transformed into a collection of graphs, this allows set of vertices in the hypergraph be treated as set of vertices in an ordinary graph. This transformation makes it possible to apply graph algorithms in order to extract useful information. Edges can then be added to connect vertices in the same hyperedge resulting in a clique for each hyperedge.

Partitioning algorithms such as Kernighan-Lin (KL) algorithm and Fidduccia-Mattheyses (FM) as described in [7] are two of the most used vertex-swapping approaches used for hypergraph clustering, i.e. after applying the above-mentioned transformation from hypergraph to graph. These two algorithms have their application in VLSI circuits design. Moreover, general clustering algorithms like “K-mean”, “Minimum Cut” and “Spectral clustering” can also be applied [5]. This extends to other community-based clustering algorithm such as MetaFac [8], Louvain Algorithm [9], RG+[10], MCL [11] and K-Clique-Community [12]. Abovementioned algorithms cluster the graph based on its structure and leave out the node-based attributes of the graph, which may improve clustering quality. From the survey, several algorithms had been proposed such as k-SNAP [13] and SA-Cluster [14] for attributed graph clustering. Both algorithms reported a better result comparing to clustering algorithm that find clusters based on the structure of the graph alone [13,14]. There are also works that include the edge attribute in clustering process as can be seen in [6] where the authors applied clustering algorithm on the signed social network. One of the critical weakness of all of these algorithms is the computation complexity is such that they do not scale well when the graph is large. Most of the algorithms were tested against sparse graphs with about 1000-10000 nodes [5, 7, 10, 11, 12] which cannot be compared to the size of biomedical graphs since they are exceedingly larger than a thousand nodes. In fact, we have experimented on the Louvain Algorithm [9] and K-Clique-Community [12], only to find that they cannot handle Erdos-Renyi Random Graph with 10000 nodes and p = 0.025 - 0.1 [15] on a robust notebook (Intel core i7, 16GB RAM). This stresses once more the need of heuristic and metaheuristic algorithms.

As the clustering problem can be reduced to cluster quality metric optimization, mataheuristic algorithms can be applied easily. Examples of metaheuristic algorithms that can be used are Tabu Search and Simulated Annealing [7], Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization and other hybrid algorithms [16, 17, 18, 19]. In [17], the authors compared existing metaheuristic clustering algorithms for wireless networks with some of the work used a graph to represent the network. These algorithms tried to solve the energy consumption problem to prolong the network lifetime in a dynamic environment where nodes move from one region to another. Most algorithms also adapted to dynamic environments by allowing a variable number of clusters.

More efforts have been put in to make the metaheuristic based clustering algorithm become more autonomous or parameterless to overcome the parameter problem in many state-of-the-art clustering algorithm. As in [18], the author proposed a genetic algorithm which employs different number of clusters per solution along with four fitness functions that are mostly based on structural aspects. This eliminate the burden of deciding number of cluster and eliminate the bias that may hinder the analysis. Another GA-based-clustering technique was proposed in [19]. The authors Evolutionary Algorithm resulted in a two-phase clustering algorithm where EA is used to refine the solutions before feeding to GA and improve the end result. Yet again, the fitness function is based on the graph structure (i.e. ratio of Intra- and Inter-connectivity). Even though many efforts have been focusing on clustering techniques, little has been done on improving the current quality metrics such as Modularity, Connectivity, and Coverage [10, 16].

In [20], the authors explore user-guided clustering. In this work, the seed clusters are selected based the consolidation of multiple criteria from multiple users. Then the local clustering algorithm is applied on each seed clusters. This idea appears to be particularly suited for the proposed use case.

4. Conclusion and research directions

There exist several algorithms for hypergraph clustering. They differ in terms of both problem complexity and objectives. Each algorithm has its own limitation and shortcomings. Therefore, improvement can be made and there are scenarios where existing algorithms cannot offer satisfactory solutions.

This article proposes an approach where hidden knowledge may be extracted by an attempt to cluster the Parkinson Disease map (i.e. the hypergraph). The approach comprises the following characteristics :

1. Meta-heuristic-based approach: Due to the NP -hard nature of clustering, meta-heuristic is a suitable approach to tackle the complexity and the large size of the Parkinson Disease map.
2. Node-based and edge-based knowledge extraction: The result of the clustering of the hypergraph of the Parkinson Disease map ought to enable knowledge extraction hidden in the nodes and edges.
3. Multi-objective approach: Assessing the clustering quality is a multi-objective problem which implies a hierarchical approach, a combination or a pareto-based approach for the various objectives and metrics.
4. Expert-driven search. As many possibilities of clusters as well as traversal paths may exist, expert knowledge has to be incorporated in order to find satisfactory solutions

We intend to design such algorithm as upcoming work and collaborate the assessment with biologists.

References

[1] C. Pizzuti and S. E. Rombo (2014). Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods. Bioinformatics. 30(10), 1343-1352.

[2] S. Parthasarathy, S. Tatikonda and D. Ucar (2010). A Survey of Graph Mining Techniques for Biological Datasets. Managing and Mining Graph Data, 40, 547-580.

[3] OpenBEL Wiki. (n.d.). Retrieved May 4, 2016, from http://wiki.openbel.org/display/home/Home

[4] K. A. Fujita, M. Ostaszewski, Y. Matsuoka, S. Ghosh, E. Glaab, C. Trefois, I. Crespo, T. M. Perumal, W. Jurkowski, P. M. A. Antony, N. Diederich, M. Buttini, A. Kodama, V. P. Satagopam, S. Eifes, A. del Sol, R. Schneider, H. Kitano, R. Balling (2013) Integrating pathways of Parkinson's disease in a molecular interaction map. Molecular Neurobiology, 49, 88–102.

[5] S. E. Schaeffer (2007). Survey: Graph clustering. Computer Science Review, 1(1), 27-64.

[6] K. Y. Chiang, C. J. Hsieh, N. Natarajan, I. S. Dhillon and A. Tewari (2014). Prediction and Clustering in Signed Networks: A Local to Global Perspective. Journal of Machine Learning Research, 15, 1177-1213

[7] A. Trifunović (2006). Parallel Algorithms for Hypergraph Partitioning (Doctoral Dissertation). Retrieved from https://www.doc.ic.ac.uk/~wjk/publications/trifunovic-2006.pdf

[8] Y. R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram and A. Kelliher (2009). MetaFac: community discovery via relational hypergraph factorization. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 527-536.

[9] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008, 10008.

[10] M. Ovelgonne and A. Geyer-Schulz (2010). Cluster Core and Modularity Maximization. 2010 IEEE International Conference on Data Mining Workshops, 1204-1213.

[11] U. Brandes, M. Gaertler and D. Wagner (2003). Experiment on Graph Clustering Algorithms. Algorithms – ESA 2003, 2832, 568-579.

[12] G. Palla, I. Derényi, I. Farkas and T. Vicsek (2005). Nature, 435, 814-818.

[13] Y. Tian, R. A. Hankins and J. M. Patel (2008). Efficient Aggregation for Graph Summarization. Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 567-580.

[14] Y. Zhou, H. Cheng and J. X. Yu (2009). Graph Clustering Based on Structural/Attribute Similarities. Proceedings of the VLDB Endowment, 2(1), 718-729.

[15] A. A. Hagberg, D. A. Schult and P. J. Swart (2008). Exploring network structure, dynamics, and function using NetworkX. Proceedings of the 7th Python in Science Conference, 11–15.

[16] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, A. C. Ponce Leon F. de Carvalho (2009). A Survey of Evolutionary Algorithms for Clustering. IEEE Transactions on Systems, Man, and Cybernetics, 39(2), 133-155.

[17] M. M. Moshizi, V. K. Bardsiri and E. Heydarabadipour (2015). The application of Meta-Heuristic based Clustering techniques in Wireless Sensor Network. International Journal of Control and Automation, 8(3), 319-328.

[18] G. Bello-Orgaz, H. D. Menéndez and D. Camacho (2012). Adaptive K-Means Algorithm for Overlapped Graph Clustering. International Journal of Neural Systems, 22(5), 1250018.

[19] J. Kohout and R. Neruda (2013). Two-Phase Genetic Algorithm for Social Network Graphs Clustering. 27th International Conference on Advanced Information Networking and Applications Workshops, 197-202.

[20] J. Cao, S. Wang, F. Qiao, H. Wang, F. Wang, P. S. Yu (2016). User-Guided Large Attributed Graph Clustering with Multiple Sparse Annotations. Advances in Knowledge Discovery and Data Mining, 9651, 127-138.

Nature inspired computing for complex industrial problems

Particle Swarm Optimization for the Location Routing Problem

Sohaib Afifi[1], Laure Brisoux[2], Corinne Lucet[2], Aziz Moukrim[3]

[1] Laboratoire de Génie Informatique et d'Automatique de l'Artois EA 3926, Université d'Artois, Béthune, France
sohaib.afifi@univ-artois.fr

[2] Laboratoire de Modélisation, Information & Systèmes EA 4290, Université de Picardie Jules Verne, Amiens, France
{laure.brisoux, corinne.lucet}@u-picardie.fr

[3] Laboratoire HeuDiaSyC UMR 7253, Sorbonne universités, Université de Technologie de Compiègne, Compiègne, France
aziz.moukrim@hds.utc.fr

1 Introduction

The Location Routing Problem (LRP) is a logistic problem often seen as the combination of two difficult decision problems, the well known Vehicle Routing Problem (VRP) [1] and the Facility Location Problem (FLP) [2]. LRP is a very studied combinatorial optimization problem in which the aim is to minimize the total cost by simultaneously selecting the potential depots and designing optimal tours for a set of vehicles serving a set of customers or clients geographically distributed and respecting some constrains. LRP NP-Hard. Readers are encouraged to refer to [3] for a recent survey.

Motivated by the potential applications and by the challenge of computational time and solution quality, we propose a Particle Swarm Optimization [4] algorithm. PSO algorithms use a swarm of candidate solutions (particles) that are iteratively moved on the search space, according to an individual and modular velocity.

2 Problem formulation

In this work, we consider a variant of LRP with one single route per capacitated depot. Thus a single uncapacitated vehicle is associated with each opened depot. It starts and ends at this point. The demands of the customers must all be satisfied where each one is served by exactly one vehicle. The problem can be modeled by a directed graph $G = (V \cup D, A)$, where $V = \{1, \ldots, n\}$ is the set of customers, $D = \{d_1, \ldots, d_m\}$ is the set of depots and $A = (V \times V) \cup (V \times D) \cup (D \times V)$ is the set of arcs. There are no loop and no arc between depots. A travel time T_{ij} is associated to each arc (i, j). Each depot $d_i \in D$ has a limited capacity Q_i and an opening cost O_i. Each customer $i \in V$ has a non-negative demand q_i which should be satisfied. In this problem we consider that the constraints of depot capacities could be violated, with a fixed penalty cost P_{capa}.The objective is to minimize the cost of traveling and of opening depot, but also penalties induced by the capacity overflows.

A valid solution for LRP can be represented by a collection r of disjoint tours $r_1, r_2, \ldots r_k$ with $k \leq m$, such that exactly one depot is assigned per tour and exactly one tour is assigned per customer. We denote $d_{\alpha(l)}$ the depot assigned to the tour r_l and $s^l_1, s^l_2, \ldots, s^l_{|r_l|}$ the customers served by tour r_l with the respected visit order. We note $r_l = s^l_1 s^l_2 \ldots s^l_r d_{\alpha(l)}$ such tour. Then the objective cost associated to r_l and to $r = r_1, r_2, \ldots r_k$ can be written as follows (see equation 1):

$$C(r_l) = \sum_{(i,j) \in r_l} T_{ij} + O_{\alpha(l)} + MAX\{0, (\sum_{i \in r_l} q_i) - Q_{\alpha(l)}\} * P_{capa} \quad \text{and} \quad C(r) = \sum_{l=1}^{l=k} C(r_l) \qquad (1)$$

3 PSO Algorithm for LRP

Our PSO works with a swarm of P particles. A typical iteration consists of updating the position x_i^{t+1} and velocity v_i^{t+1} of each individual i of the swarm according to x_i^t (current position), x_i^{lbest} (best known position of the particle), global best known solution x^{gbest}, and current velocity v_i^t as

in standard PSO strategy [4]. Each particle has a p_{LS} probability to be improved through a local search process. The algorithm is stopped after *itermax* consecutive position updates have failed to give a new local best.

3.1 Position of particle and evaluation

More precisely, the *position* x_i of particle i, is a giant tour of all accessible customers in V, i.e. a solution for the traveling salesman problem (TSP). A giant tour is a permutation of the n customers. This one can be transformed into LRP solution $r = r_1, r_2, \dots r_k$ by splitting it into $k \leq m$ tours and assigning one depot to each of them. Adapting the splitting process of Duhamel *et al.* [5] to our LRP problem, allows us to compute from a giant tour x_i the corresponding LRP solution $r(x_i)$ that is the best solution that respects the order of customers, described by the giant tour. The cost of the giant tour x_i is equal to the cost of this $r(x_i)$ solution : $C(x_i) = C(r(x_i))$. At the beginning of the process, P particles (i.e. P permutations of the n customers) are randomly generated. Then, we improve 10% of the particles by a local search method.

3.2 Particle movement: a crossover operator

The updating of the position can be interpreted as a recombination of three positions representing the inertia, cognitive and social parameters of the PSO. There are various ways to combine positions. In our approach, we apply the operator of movement presented in [6] that is comparable to a genetic crossover. The crossover principle is to extract subsequences of respectively nb_1, nb_2 and nb_3 customers from x_i, x_i^{lbest} and x^{gbest} such that $n = nb_1 + nb_2 + nb_3$ and to juxtapose them to update position x_i. The procedure of extraction is applied to x_i, x_i^{lbest} and x^{gbest} in a random order, with respect to the three PSO parameters w, c_1 and c_2, used to calculate nb_1, nb_2 and nb_3 as follows:

$$nb_1 = w.n,\ nb_2 = (1-w).n.\frac{c_1.\pi_1}{c_1.\pi_1 + c_2.\pi_2} \text{ and } nb_3 = (1-w).n.\frac{c_2.\pi_2}{c_1.\pi_1 + c_2.\pi_2}$$

π_1 and π_2 are real random numbers in $[0, 1]$. nb_1, nb_2 and nb_3 are rounded with the nearest integer. The three subsequences thus obtained are juxtaposed in a random order.

3.3 Intensification operator

In our PSO algorithm we choose to apply intensification process at each iteration and on each updated position x_i of particle i with a probability p_{LS}. Our intensification operator is a local search which contains four neighborhood functions: $2opt^*$ and $OrOpt$, the classical movements for all routing problems as TSP; $Depot\_Swap$, movement that explores all exchanging of tours between depots; $Destruction/Repair$ movement that randomly chooses a set of d customers, removes them, and reinserts with a variant of the Best Insertion Algorithm (BIA) [7]. These functions aim at the improvement of the LRP solutions and thus work directly on $r(x_i) = r_1, \dots r_k$. To every application of the intensification operator, the four neighborhood functions are called successively and iteratively as long as improvements are found. The resulting LRP solution $r(x_i)$ is transformed into a giant tour x_i by concatenation.

4 Experimental results

Our PSO algorithm was implemented in C++ language. The program is compiled with GNU GCC in a Linux environment and all experiments were conducted on an Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz.

In order to assess its efficiency we compared our results (the best solution) with those of the VNS algorithm presented in [8]. They proposed an iterative local search (ILS) algorithm embedded in a Genetic scheme in the first work and then a variable neighborhood search (VNS) in the second. The tested instances are first the 36 Tuzun and Burke instances [9], with 100 to 200 customers and 10 to 20 depots, and secondly the 30 Prins *et al.* instances [3] with 20 to 200 customers and 5 to 10 depots. Tables 1 and 2 report our results and compare them with the best results of the literature. Column **Best** shows the average of the best known solution collected from all methods

n	m	Best	Literature			PSO		
			Obj	Cpu	%gap	Obj	Cpu	%gap
100	10	610.7	611	12.2	0	610.7	6.7	0
100	20	604.2	604.2	10.1	0	604.2	8.5	0
150	10	772.2	773.5	68.2	0.2	772.2	44.6	0
150	20	781	781.2	73.5	0	781	47.1	0
200	10	688.2	688.2	39.2	0	688.2	15.2	0
200	20	672.5	673	42	0.1	672.5	21	0

Table 1. Average results on Tuzun and Burke's instances

n	m	Best	Literature			PSO		
			Obj	Cpu	%gap	Obj	Cpu	%gap
20	5	15582.3	17039	0	13.7	15582.3	0.4	0
50	5	18237.9	18238	3.9	0	18237.9	6.5	0
100	5	104738.5	105631.5	22.7	0.7	104738.5	47.1	0
100	10	144442	144444.5	34.2	0	144442	50.6	0
200	10	249592.3	249592.3	51.6	0	249592.3	50.7	0

Table 2. Average results on Prins *et al.*'s instances

(including ours) for each class of instances. Columns **n** and **m** represent respectively the number of customers and depots. Columns **Obj** and **Cpu** report the average of the best solution found by each method and the computational time. **%gap** reports the average relative gap to the best solutions. For the first family of instances, our algorithm obtains a better solution for 7 instances of 36 and the second one for 6 instances of 30 within a comparative computation time.

5 conclusion

We have presented a PSO algorithm to solve the Location Routing Problem restricted to one vehicle per tour. We use a solution representation that allows to consider a subset of the LRP solution set and extracts, thanks to a special splitting process, the corresponding optimal solution. We also used local search to improve each solution with a given probability. The results obtained on a set of benchmarks showed the efficiency of this approach.

Acknowledgment This work is carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program Investments for the future managed by the National Agency for Research (Reference ANR-11-IDEX- 0004-02). It is also partially supported by the Regional Council of Picardie and the European Regional Development Fund (ERDF), under PRIMA project and TCDU project (Collaborative Transportation in Urban Distribution, ANR-14-CE22-0017).

References

1. Toth, P., Vigo, D.: The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (2002)
2. Holmberg, K., Rönnqvist, M., Yuan, D.: An exact algorithm for the capacitated facility location problems with single sourcing. EJOR **113** (1999) 544–559
3. Prodhon, C., Prins, C.: A survey of recent research on location-routing problems. EJOR **238** (2014) 1–17
4. Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Volume 4 (1995) 1942–1948
5. Duhamel, C., Lacomme, P., Prins, C., Prodhon, C.: A GRASP×ELS approach for the capacitated location-routing problem. Computers & Operations Research **37** (2010) 1912–1923
6. Dang, D.C., Guibadj, R.N., Moukrim, A.: A PSO-Based Memetic Algorithm for the Team Orienteering Problem. In: EvoApplications 2011, Proceedings, Part II. (2011) 471–480
7. Bouly, H., Dang, D.C., Moukrim, A.: A memetic algorithm for the team orienteering problem. 4OR **8** (2009) 49–70
8. Jarboui, B., Derbel, H., Hanafi, S., Mladenović, N.: Variable neighborhood search for location routing. Computers & Operations Research **40** (2013) 47–57
9. Tuzun, D., Burke, L.: A two-phase tabu search approach to the location routing problem. EJOR **116** (1999) 87–99

Optimization methods for the design of production systems

Optimized Positioning of a Robot Arm boring Regularly Distributed holes on a metallic sheet using Simulated Annealing Heuristic

Dj. Bellala[1] L. Ourlis[2] and H. Smadi[3]

1. *C. E. M. L., UNIVERSITY OF BATNA 2, 05000 Algeria*
bellala_djamel@yahoo.co.uk

2. INDUSTRIAL ENGINEERING Dept., UNIVERSITY OF BATNA 2, 05000 Algeria
ourlisl@gmail.com

3. *IHSI, UNIVERSITY OF BATNA 2, 05000 Algeria*
h.smadi@hotmail.fr

Keywords: random variable, continuous uniform distribution, probability density function, expected value, quadratic function, simulated annealing.

1 Introduction

For use in cooling and air conditioning installations, a robot arm has to perform a set of circular holes over a rectangular metallic sheet. These holes are accurately calculated in order to meet the cooling system requirements. The metallic sheet is long enough that the robot arm has to move nearby to it and perform its task as quickly as possible. The robot arm instantaneous location points have to be optimally determined so that it can perform the overall task in a minimum period of time. The model of the robot arm displacement along the metallic sheet is obtained from its comings and goings while performing the holes' boring by analyzing the average distance between each robot arm position and every hole to be made. Since the holes should be regularly and uniformly distributed, consequently their respective positions are taken as continuous random variables and calculated according to some probability laws. Once the positioning model has been obtained, which quadratic in our case, we show in how the robot arm should be positioned along the edge of the metallic sheet in order to minimize its total displacement. To solve such a problem the choice is made upon the simulated annealing heuristic, a useful numerical technique used for solving large and difficult problems where global optima are required.

References

[1] L. George, P. John 2000. "*Numerical Methods using Matlab*", Prentice Hall, Upper Saddle River, New Jersey 07458.

[2] B. Jan, T. Vladimir 2005. "*Optimization: Insights and Applications*", Princeton University Press, Princeton Series in Applied Mathematics,.

[3] T.C. Martins, A.K. Sato and all, August, 2012. "*Simulated Annealing – Advances, Applications and Hybridizations*", Marcos de Sales Guerra Tsuzuki, Janeza Trdine 9, 51000 Rijeka, Croatia.

[4] Z. Michalewicz, David B. Fogel 2OO4. "*How to Solve It: Modern Heuristics*", 2nd edition, Springer-Verlag Berlin Heidelberg New York.

[5] F. S. Hillier, G. J. Lieberman, "Introduction to Operations Research", McGraw-Hill international, ninth edition 2010.

[6] Singiresu S. Rao 2009. "*Engineering optimization: theory and practice*", 4th ed., John Wiley & Sons, Inc., Hoboken, New Jersey.

[7] R. K. Arora 2015. "*Optimization Algorithms and applications*", CRC Press, Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742.

Constructive and Enhanced Heuristics for a Welding Process Optimization of a Central Girder of a Semi Trailer

Dj. Bellala[1] H. Smadi[2] A. Medjghou[3] and L. Ourlis[4]

1. *C. E. M. L., UNIVERSITY OF BATNA 2, 05000 Algeria*
bellala_djamel@yahoo.co.uk

2. IHSI, UNIVERSITY OF BATNA 2, 05000 Algeria
h.smadi@hotmail.fr

3.4. INDUSTRIAL ENGINEERING Dept., UNIVERSITY OF BATNA 2, 05000 Algeria
amedjghou@yahoo.fr
ourlisl@gmail.com

Keywords: Optimization, nearest-neighbor algorithm, subtour-reversal heuristic, welding process, robot arm, TSP.

1 Introduction

Many industries carry out metal assemblies of elements through the welding process. In order to increase the welding quality as well as the industrial productivity, robots are introduced. In addition to their flexibility, these robots are used to automate the welding process and achieving their tasks as rapidly as possible. Nowadays, the use of robotized welding technologies became very common due to their various advantages like the good precision, the decrease in the expenditure of the end product especially in terms of labor, stocking and expedition costs. In addition to that, these robots can operate in difficult conditions as at high temperature and hostile environments. The objective of this paper is to optimize the welding process of a central girder using two heuristics: the nearest-neighbor algorithm and the subtour-reversal heuristic. The first algorithm, which is characterized by being easy to implement, is used as a home base for the second one, which requires more computations. The second algorithm, which usually yields better results, is used to considerably improve the outcome of the first algorithm. Since the working area is modeled by an array representing the actual traveling distance between each pair of the welding points and that the robot arm can move back and forth without obstacle, then the mathematical programming formulation of the problem is that of the TSP, where the objective is to minimize the total task time of the welding process and satisfying the constraint which ensures that every welding point is visited by the robot arm only once. In this paper the two algorithms are combined together into one heuristic, in which the output of the nearest-neighbor algorithm is used as an input for the subtour-reversal one. This work is based on a case study within the TIRSAM company specialized in manufacturing semi trailers.

References

[1] H. A. Taha, 2007 "*Operations Research an Introduction*", eighth edition, Pearson International Edition Upper Saddle River, New Jersey 07458.

[2] L. George, P. John 2000 "*Numerical Methods using Matlab*", second edition, Prentice Hall, Upper Saddle River, New Jersey 07458.

[3] F. S. Hillier G. J. Lieberman, 2010 "*Introduction to operations research*", ninth edition, McGraw-Hill international edition.

[4] Singiresu S. Rao 2009. "*Engineering optimization: theory and practice*", 4th ed., John Wiley & Sons, Inc., Hoboken, New Jersey.

[5] R. Bronson G. Naadimuthu, 1997 "*Operations Research*", second edition, Schaum's outline series, McGraw-Hill.

[6] D. B. Marghitu, 2009 "*Mechanisms and Robots Analysis with MATLAB*" British Library Cataloguing in Publication Data, Springer-Verlag London Limited.

[7] F. Greco, 2008 "*Traveling Salesman Problem*" In-Teh is Croatian branch of I-Tech Education and Publishing KG, Vienna, Austria.

[8] B. I. Kazem, And all 2008 "*Motion Planning for a Robot Arm by Using Genetic Algorithm*" Jordan Journal of Mechanical and Industrial Engineering, Vol. 2, Num. 3, ISSN 1995-6665 Pages 131 – 136.

sciencesconf.org:imeta2010:109930

Minimizing the maximum lead time in a 3-stage supply chain scheduling problem

W. Besbes[1], J. Teghem[2] and T. Loukil[3]

[1] CIT College, Taif University, KSA
walid.besbes@gmail.com
[2] MathRO Laboratory, University of Mons, Belgium
Jacques.Teghem@umons.ac.be
[3] Tabuk University, KSA
loukilt@gmail.com

1 Introduction

Supply Chain Management (SCM) is a relative new subject in Operations Research which is receiving now an extensive attention (see for instance [1]). The aim of SCM is ambitious: to integrate and coordinate all the actors of an industrial process from the initial suppliers of raw materials or sub-products till the final customers through different levels of production. Often the three main actors considered are the suppliers (S), the manufacturers (M) and the customers (C). It is the case in this study in which, we analyse a supply chain scheduling (SCS) model that can be modelized by a 3-stage HFS. In our problem, three types of constraints are considered: (i) **Eligibility constraints**: particular suppliers and manufacturers are eligible to produce each job or product for a certain customer. In our model, we will suppose that the choice has been realized, i.e., in other words, only one path supplier-manufacturer-customer, noted (s(j)-m(j)-c(j)) is dedicated to each job j. (ii) **Transportation constraints**: each supplier makes deliveries to several manufacturers m, which also make deliveries to different customers c. There exists a transportation time between each pair (s-m) and (m-c). (iii) **Batch delivery constraints**: at each supplier s and each manufacturer m, the jobs are formed into batches of a fixed capacity. Each batch is delivered to the next stage; it is either m for s or c from m, into a single shipment. Due to the complexity of this problem two approached methods are proposed to minimize the maximum lead time in addition to a new lower bound to evaluate the effectiveness of both approaches. The outline of this paper is structured as follows. Section 2 goes over the related literature. In section 3, we formally describe the problem in detail. Section 4 elaborates the new proposed heuristic whereas the proposed GA is presented in section 5. The new lower bound developed for this problem is detailed in section 6. Section 7 gives a detailed computational evaluation of the proposed heuristic and the GA. Finally, some conclusions on this work are given.

2 Related works

The supply chain scheduling (SCS) problem has recently gained a growing importance, especially during the last few years. Nevertheless, we remark that most of works dealing with this problem consider only two integrated stages SCS models: either material supply and production or production arrangement and finished-goods delivery. For example, the reader is referred to the works of [3], [4], [5], [6], [7] and [8]. More recently, Zegordi et al. [9] considered a two-stage supply chain system. The first stage is composed of several suppliers with different production speeds. The second one is a single manufacturer. The products delivery from suppliers to manufacturer is ensured by several vehicles with different speeds and different transportation capacities. They proposed a mixed integer programming model and a "gendered" genetic algorithm, that employs two different chromosome structures, to minimize the total completion time. Barzoki and Hejazi [10] considered the production arrangement and finished-goods delivery issue while studying a two-stage supply chain system. In fact, there is a single machine to process the jobs that will be delivered in batches to several customers. Their objective is to minimize the sum of the total weighted number of tardy jobs, the total due date assignment costs and the total batch delivery costs. As said above, several works studying the SCS problem have been proposed. For an extensive literature review of the

problem, we refer the reader to the recent survey paper of Chen [11]. To the best of our knowledge, only a handful of papers have considered the integrated 3-stage supply chain problem. Hall and Potts [2] studied the scheduling models integrating the supply of raw materials, the production scheduling and finished-goods delivery. They aimed to minimize the overall scheduling and delivery cost. In their models, there are only one supplier and several manufacturers and customers. However they did not consider in their problem the jobs transportation times. They proposed dynamic programming algorithms for some problems, proving the intractability of such problems. Li and Ou [12] studied a particular case of two-stage supply chain that can be considered as a three-stage supply chain model. Formally, there are a single machine and a single warehouse. Initially, the jobs are located at the warehouse. Then, they will be transported to the factory (single machine) for processing. Once the processing has been finished, the jobs will be delivered back to the warehouse. They assumed that there is one vehicle for delivery activities with a limited capacity of transportation. After analyzing the computational complexity of the problem, they discussed some particular cases of the problem that can be solved to optimality and then, proposed a heuristic for the general problem while analyzing its worst-case performance. Wang and Chen [13] considered the 3-stage supply chain makespan minimization problem. Each one of the three stages is located at a different place. They assumed that there are only one supplier, one manufacturer and also one customer in their problem, contrary to our model. They considered one capacitated vehicle between supplier and manufacturer and also one another capacitated vehicle between manufacturer and customer. In the same year, Sawik [14] studied a three-stage supply chain model with suppliers, manufacturers (producers) and customers. However, he assumed that the transportation times are identical for all shipping lot. Moreover, he did not really focus on the batch delivery issue, subject of the present work. More recently, Soltani and Sadjadi [15] considered a 3-stage supply chain composed of several suppliers, a distribution center and several customers. Nevertheless, they only focused on the transportation facilities optimization and so, they did not consider the production scheduling issue. They proposed two hybrid meta-heuristics based on a simulated annealing and a variable neighbourhood search to minimize the total flow time. Note that in all these works, the authors did not consider in their models the case of several suppliers, manufacturers and customers. Motivated by this remark, we consider in this paper an integrated three-stage supply chain scheduling problem with several suppliers, manufacturers and customers, located at different positions.

3 Problem definition

In the considered supply chain, there are several suppliers (S), manufacturers (M) and customers (C). Initially, a set of orders (n jobs) must be sent to customers. Each job j (j=1,2,...,n) requires to be processed by a single supplier $s(j)$ to produce a semi-finished good and, by a single manufacturer $m(j)$ to produce finished goods. We assume that job preemption is forbidden. Note that, for each job, the supplier and the manufacturer which will process it are known beforehand. Each job j ($j = 1, 2, ..., n$) has a processing times p_{1j} on the first stage (S) and p_{2j} on the second stage (M). Initially, the unprocessed jobs, as raw materials, are located at suppliers and need to be processed at this level before being delivered as semi-finished goods to manufacturers for further process. Then, after finishing their process on stage 2 (M), the jobs need to be transported to the customers, too. We also assume that for each job, we know beforehand the $c(j)$ to which, it will be delivered. Hence, each job is finally assigned to a triplet (s(j)-m(j)-c(j)). At each supplier, a subset of an unlimited number of vehicles is available to transport semi-finished goods to manufacturers. We assume that the vehicles have non-identical capacities of loading, which depends on the supplier s from which they will transport batches of k^s jobs. Additionally, each vehicle needs $t_{sm} > 0$ units of time to travel from a supplier s to a manufacturer m. We do not consider the return time from m to s since we assume that there is an unlimited number of vehicles. In the same way, there is a subset of an unlimited number of vehicles at each manufacturer to deliver finished goods to customers. We also assume that the vehicles have non-identical capacities of loading, which depends on the manufacturer from which they will deliver batches of k'^m jobs. In other words, all vehicles located at a given manufacturer m, have the same loading capacity k'^m, i.e. the size of the batch. However, this capacity may be different if we consider another manufacturer. Moreover, each vehicle needs $t'_{mc} > 0$ units of time to transport jobs from a manufacturer m to a customer c. Therefore, since each job j is assigned to a triplet (s-m-c), its transportation times from s to m ($t_{s(j)m(j)}$) and

from m to c ($t'_{m(j)c(j)}$) are known beforehand. We should note that the loading and unloading process of jobs are included in the transportation time. So, each job is available for processing at the corresponding supplier at time $t = 0$ and is available for delivery to the corresponding manufacturer once its processing is finished. Additionally, we assume that the cardinality of each subset of jobs assigned to a given supplier s (n_s) (resp. manufacturer m (n_m)), is a multiple number of the corresponding batch size k^s (resp. k'^m). Mathematically, the non-negative integer b_s (resp. b'_m) satisfying $n_s = b_s.k^s$ (resp. $n_m = b'_m.k'^m$) is the exact number of delivery batches from s to the manufacturers (resp. from m to the customers).
The logistical issues of our problem are to determine:

- the starting times of processing each job on both stages s and m.
- the departure time of each batch from s to m and from m to c.
- which jobs will be transported in each delivery batch.

The objective is to minimize the last lead time, i.e., the arrival time of the last delivered job to a customer or the arrival time of the last delivered batch to a customer. For convenience and readability, we summarize below notations that we will use in the remaining parts of this paper.

- J_s: the subset of jobs to be processed by supplier s, $s = 1, 2, \ldots, S$ with $|J_s| = n_s$, the number of jobs processed by supplier s;
- J_m: the subset of jobs to be processed by manufacturer m, $m = 1, 2, \ldots, M$ with $|J_m| = n_m$, the number of jobs processed by manufacturer m;
- J_c: the subset of jobs to be delivered to customer c, $c = 1, 2, \ldots, C$ with $|J_c| = n_c$, the number of jobs delivered to customer c;
- n_{mc}: the number of jobs delivered from manufacturer m to customer c, $m = 1, \ldots, M$; $c = 1, \ldots, C$;
- $s(j)$: the job j is processed by supplier $s(j)$, $j \in J$;
- $m(j)$: the job j is processed by manufacturer $m(j)$, $j \in J$;
- $c(j)$: the job j is delivered to customer $c(j)$, $j \in J$;
- p_{1j}: the processing time of job j at supplier $s(j)$, $j \in J$;
- p_{2j}: the processing time of job j at manufacturer $m(j)$, $j \in J$;
- t_{sm}: the transportation time of any batch from supplier s to manufacturer m, $s = 1, 2, \ldots, S, m = 1, 2, \ldots, M$;
- t'_{mc}: the transportation time of any batch from manufacturer m to customer c, $m = 1, 2, \ldots, M, c = 1, 2, \ldots, C$;
- k^s: the size of each batch delivered from supplier s, $s = 1, 2, \ldots, S$;
- k'^m: the size of each batch delivered from manufacturer m, $m = 1, 2, \ldots, M$;
- b_s: the total number of batches delivered from supplier s, $s = 1, 2, \ldots, S$;
- b'_m: the total number of batches delivered from manufacturer m, $m = 1, 2, \ldots, M$.
- b_{sm}: the total number of batches delivered from supplier s to manufacturer m, $s = 1, 2, \ldots, S$; $m = 1, 2, \ldots, M$;
- b'_{mc}: the total number of batches delivered from manufacturer m to customer c, $m = 1, 2, \ldots, M$; $c = 1, 2, \ldots, C$.

We recall that each job j is dedicated to one path (s-m-c). Thus for a specific path s-m-c, we have n_{smc} the total number of jobs following this path. Based on the well-known three-field $\alpha|\beta|\gamma$ notation of [16], its extended version proposed by [17], the recent five-field notation of [11], we denote the considered model as follows: $FH3|\,|S(j)| = |M(j)| = 1, transport|V(\infty, k^s/k'^m), direct|C|D_{max}$ where:
FH3: a three-stage hybrid flow shop;
$|S(j)| = |M(j)| = 1$: machines are dedicated. In fact, each job j is dedicated to one path (s-m-c);
transport: the transportation constraint from suppliers to manufacturers;
$V(\infty, k^s/k'^m)$: it is related to vehicle characteristics and delivery methods.(∞) specifies that there is a sufficient number of vehicles available either from S to M or from M to C. Therefore, the vehicle availability is not a constraint in our case. (k^s/k'^m): means each shipment from supplier s (resp. m) can deliver up to k^s (resp. k'^m) jobs. In our case, it is exactly k^s (resp. k'^m) jobs.
direct: *batch delivery by direct shipping*, i.e., only orders going to the same manufacturer (resp. customer) can be delivered together in the same shipment.
C: in our work, we consider the model with multiple customers ($C \geq 2$);
D_{max}: the considered measure of performance represents in our problem the customer service level.

4 The heuristic Hb

We propose a new heuristic, noted Hb, which is based on two steps, related respectively to the stages one and two. In the first one, we proceed similarly to the Johnson's algorithm for each supplier s. After building both subsets J_{s1} and J_{s2} (see algorithm 1), we obtain a preliminary sequence σ_s for each supplier s ($s = 1, \ldots, S$). According to the order of jobs in this sequence σ_s, we will gather the jobs into b_s batches (where each one of them contains k^s jobs); obviously, each batch contains jobs which must be delivered to the same manufacturer. Regarding the obtained b_s batches at each supplier s, we will compute for each one of them two values p_{1b} and p_{2b}. Indeed, for each batch b, p_{1b} (resp. p_{2b}) is equal to the sum of the processing times on stage S (resp. M) of all jobs forming this batch b and the transportation time t_{sm} (resp. t'_{mc}) between the current supplier s (resp. manufacturer m) and the manufacturer m (resp. customer c) to which this batch b will be transported. Then, for each supplier s, two sub-sequence of batches will be built: B_{s1} and B_{s2}. B_{s1} contains the batches having a p_{1b} lower than or equal to p_{2b}, sorted in ascending order of their p_{1b}. Therefore, the second sub-sequence B_{s2} will be composed of the remaining batches sorted in a non-ascending order of their p_{2b}. Finally, each job $j \in J_s$ will be processed on s following the order of the sequence $B_s = B_{s1} \cup B_{s2}$. Concerning the second step of this heuristic, we will essentially apply

Algorithm 1 The step 1 of Hb pseudo-code

for ($s := 1$ to S) **do**
  $\mathbf{J}_{s1} = \{j \in J_s | (p_{1j} + t_{sm(j)}) \leq (p_{2j} + t'_{m(j)c(j)})\}$
  Order the jobs of J_{s1}in an ascending order of their $(p_{1j} + t_{sm(j)})$
  $J_{s2} = \{j \in J_s | (p_{1j} + t_{sm(j)}) > (p_{2j} + t'_{m(j)c(j)})\}$
  Order the jobs of J_{s2} in a non-ascending order of their $(p_{2j} + t'_{m(j)c(j)}))$
  Build the sequence $\sigma_s = J_{s1} \cup J_{s2}$
  for ($j := 1$ to n_s) **do**
    (in the order of σ_s)
    for ($b := 1$ to b_s) **do**
      Assign j to the corresponding batch b (going to the same manufactuer)
    end for
  end for
  for ($b := 1$ to b_s) **do**
    (if the batch b is going to the manufactuer m)
    $p_{1b} = \sum_{j \in b} p_{1j} + t_{sm}$
    $p_{2b} = \sum_{j \in b} (p_{2j} + t'_{mc(j)})$
  end for
  $B_{s1} = \{b \in B_s | (p_{1b} \leq p_{2b}\}$
  Order the batches of B_{s1} in ascending order of their p_{1b}
  $B_{s2} = \{b \in B_s | (p_{1b} > p_{2b}\}$
  Order the batches of B_{s2} in a non-ascending order of their p_{2b}
  Build the sequence $B_s = B_{s1} \cup B_{s2}$
end for

the FIFO dispatching rule at each manufacturer m. Nevertheless, if we have several available jobs when the corresponding machine m becomes also available - and this appears frequently because the b_s jobs of a batch arrive simultaneously - we tie-break the rule by choosing the job j that has the largest $t'_{mc(j)}$.
We should precise at this level that three other tie-breaking rules have been tested: (i) the largest p_{2j}, (ii)the largest $(p_{2j} + t'_{mc(j)})$ and a job j to assign to the batch, not yet achieved, already containing the greatest number of jobs. The conducted preliminary comparison tests showed that the best rule to apply, in terms of solution quality is the largest transportation time $t'_{mc(j)}$.

5 The Genetic Algorithm (GA)

The proposed genetic algorithm is in fact a new adaptation of our GA, initially developed in [18]. Motivated by the encouraging results already obtained by the previous versions of our GA, developed in [18] and [19], to solve the Hybrid Flow shop Scheduling (HFS) problem (we note that the general model of SCS can be designed as a HFS), we propose a new adaptation of this GA to deal with this considered problem. The main steps of the GA are briefly described below.

5.1 Encoding and decoding of solutions

The proposed new adaptation of our GA concerns the decoding algorithm used to obtain a full schedule. Indeed, as it is previously proposed in the genetic algorithm [18], the solutions returned by the GA represent the jobs sequence before their processing on the first stage (here, the set of suppliers). So, to obtain a full schedule, we have to decode these solutions. In fact, we will gather the jobs into s-sequence according to the supplier s which will process each one of them following the order of the initial jobs sequence returned by the GA. Thereafter, based on the new order of the jobs sub-sequence at each supplier, we group these jobs into batches according to the manufacturer m to which the batches will be delivered. Concerning the second stage (manufacturer), we will proceed in the same way as in step 2 of the heuristic Hb.

5.2 Generation of the population and selection

The initial population is randomly generated with POP as its size. The roulette wheel method is used to select the potential parents from the current population for the crossover. Then, the elitism selection is applied to choose the POP fittest individuals among all individuals from the current population and also from the new individuals issued from the crossover and mutation. The main aim of this selection manner is to preserve good individuals and eliminate the bad ones from one generation to another [19].

5.3 Crossover and mutation

The proposed crossover operator (SJ-XO) is a new combination of the crossover operators based on previous studies [20] and [21]. The SJ-XO starts with the first step of the SJOX crossover operator proposed in [21]. In fact, at the beginning, they examine each couple of parents selected for crossover gene by gene and the identical jobs occupying the same positions of both parents are directly copied at the same positions to the offspring. Then, an iterative procedure is applied. It begins by copying the first available job (i.e., job not yet assigned to offspring) of parent 1 (parent 2, respectively) to offspring 1 (offspring 2, respectively). In algorithm 2, a general description of the SJ-XO crossover operator to generate the first offspring is proposed.
Concerning the mutation, the shift mutation operator is applied in this GA. It consists on a random selection of a gene from the chromosome and its insertion at another random position in the chromosome.

5.4 Tuning the GA's parameters

For this problem, we have initially conducted several tests to choose the better parameters values of our GA. The following values have been tested:

- population size: 40, 60, 80, 100, 150 and 200;
- crossover rate: 0.5, 0.6, 0.7 and 0.8;
- mutation rate: 0.05, 0.1, 0.15 and 0.2;
- the stopping criterion: 100, 250, 500, 1000, 2000, 3000, 4000 and 5000 runs without any improvement of the D_{max}.

Finally, the following parameters values for the GA have been choosen: population size = 100, crossover rate = 0.7, mutation rate = 0.1 and stopping criterion = 3000 iterations without any improvement of the D_{max}.

Algorithm 2 The crossover operator (SJ-XO) to generate OFF1

Require: P1: the first parent; P2: the second parent.

Copy the identical jobs occupying the same positions at both parents directly to the offspring (OFF1 and OFF2).
Select the first available gene of P1, record it to the first available gene of OFF1 and name it: $s - gn$.
repeat
  Name the successor gene of $s - gn$ in P1: $n - gn1$
  Name the successor gene of $s - gn$ in P2: $n - gn2$.
  if both $n - gn1$ and $n - gn2$ are available **then**
    select the gene (job) for the $OFF1$ which has the minimum total processing time and name it $s - gn$.
    if $n - gn1$ and $n - gn2$ have the same total processing time **then**
      Select randomly one of them.
    end if
  else if $n - gn1$ is available and $n - gn2$ is not available **then**
    Assign $n - gn1$ to OFF1 and name it $s - gn$.
  else if $n - gn1$ is not available and $n - gn2$ is available **then**
    Assign $n - gn2$ to OFF1 and name it $s - gn$.
  else
    Find the first available gene of P1 and P2, starting from the position of $n - gn1$ and $n - gn2$, respectively.
    if the last gene in P1 or P2 is reached **then**
      Return to the first gene in the sequence of P1 (respectively P2)
    end if
    Select the gene for the OFF1 which has the minimum total processing times and name it $s - gn$.
  end if
until OFF1 is achieved

6 The lower bound

To evaluate the proposed heuristic and GA, we develop a new lower bound for this problem. Indeed, the computed LB corresponds to the maximal stage-based lower bound value referred to LB(S) for stage 1 (suppliers) and LB(M) for stage 2 (manufacturers).

$$LB = \max\{LB(S), LB(M)\} \tag{1}$$

Concerning the first stage S, the related *LB(S)* is equal to the maximum processor-based bounds $LB(J_s)$ of each supplier s.

$$LB(S) = \max_{s}\{LB(J_s)\}, s = 1, \ldots, S \tag{2}$$

At this level, we have to compute for each supplier s its related processor-based bound $LB(J_s)$. This bound is obtained by the sum of four terms.
(i) Obviously, the completion time of the last job j processed by the supplier s is equal to the sum of the processing times of all jobs executed by this supplier s.
(ii) Thereafter, this last job will be delivered to its corresponding manufacturer m into a batch and it needs t_{sm} units of time.
(iii) Then, we have to minimize the sum of the total processing times of the jobs, forming a batch delivered by s to this manufacturer m.
(iv) Finally, we have to consider the minimal transportation time of one of these jobs i.e., t'_{mc}.
The sum of the three last terms is minimized on the different manufacturers.
Therefore, to obtain the corresponding $LB(J_s)$, we have to compute:

$$LB(J_s) = \sum_{j \in J_s} p_{1j} + \min_{m | J_s \cap J_m \neq \varnothing} \left\{ t_{sm} + \min_{\{j_1, \ldots, j_{b_s}\} \subset J_s \cap J_m} \left\{ \sum_{b=1}^{b_s} p_{2j_b} + \min_{c \in \left\{c_{(j_1)}, \ldots, c_{j_{(b_s)}}\right\}} t'_{mc} \right\} \right\} s = 1, \ldots, S \tag{3}$$

Concerning the second stage M, the related *LB(M)* is equal to the maximum processor-based bounds $LB(J_m)$ of each manufacturer m.

$$LB(M) = \max_{m}\{LB(J_m)\}, m = 1, \ldots, M \tag{4}$$

In fact, we know that the starting time of the first job j processed by a manufacturer m can be no less than the minimal value of the sum of the total processing times of all the jobs belonging to the same batch delivered from a supplier s plus its delivery time from s to m. This term is minimized on the different suppliers. If there is no idle time on the manufacturer m, the completion time of the last processed job on this machine corresponds to the sum of the latter result and the total processing times of all jobs belonging to J_m, i.e., $\sum_{j \in J_m} p_{2j}$. Finally, to be delivered to its customer c, the last processed job j on machine m needs a minimal value of transportation time t'_{mc}. The computation processor-based bound of each manufacturer m is performed as follows.

$$LB(J_m) = \min_{s|J_s \cap J_m \neq \varnothing} \left\{ \min_{\{j_1,\ldots,j_{b_s}\} \subset J_s \cap J_m} \sum_{b=1}^{b_s} p_{1j_b} + t_{sm} \right\} + \sum_{j \in J_m} p_{2j} + \min_{c|J_m \cap J_c \neq \varnothing} t'_{mc}; m = 1, \ldots, M \tag{5}$$

7 The computational study

7.1 Experimental design

Since we do not find instances of the considered problem in the literature, we conduct all computational experiments based on randomly generated instances. For this computational study, nine job sizes are considered, n = 20, 30, 40, 50, 60, 70, 80, 90 and 100 where the processing times p_{ij} and transportation times (t_{sm} and t'_{mc}) of jobs are uniformly distributed with the range of [1, 99]. We choose to fix the number of customers to four for all the considered instances, whereas, the number of suppliers and manufacturers varies from 2 to 6. In fact, five different configurations of suppliers and manufacturers are considered in this study (S-M): (2-2), (2-3), (3-3), (2-6) and (6-2). Concerning the additional constraint considered in this problem - the batch delivery - we generate four different configurations of batches capacities: 2, 2-3, 2-5 and 5 with thus two cases where the capacity is the same for all the batches which will be delivered from suppliers or manufacturers (2 and 5). In the remaining two cases (2-3 and 2-5), we may have batches with capacities 2 or 3 (resp. 2 or 5) either on stage S or stage M. Nevertheless, we assume that for each supplier (resp. manufacturer), we do not allow to have batches with different capacities, it will be the same. Finally, we should recall that once an instance is generated, a checking procedure is triggered to ensure that the number of jobs assigned to each supplier s (n_s) (resp. manufacturer m (n_m)), is a multiple number of the corresponding batch size. As a consequence, 9*5*4 = 180 instances are - a priori - generated. Nevertheless and due to the latter assumption noted above, we have to eliminate some of these instances since they become dummies with certain configurations of the generated data. For instance, if we consider (6-2-4) as respectively the number of suppliers (6), manufacturers (2) and customers (4) with 20 jobs and a capacity of batch, equal to 5 for all suppliers and manufacturers, two suppliers among the six will not process any jobs. Therefore, we finally keep 169 instances instead of 180.
In the remaining part of this section, we will computationally analyse the performance of the two proposed approaches (Hb and GA) in addition to the lower bound to solve the NP-hard considered problem. In total, we perform (2 (Hb and LB)+5(runs of the GA)) * 169 (instances) = 1183 evaluations of the D_{max}.

7.2 Computational results

We have three goals through this computational study. The first one is to analyse the tightness of the proposed lower bound by identifying the instances where the GA or Hb are able to reach this LB. Our second goal is to evaluate both algorithms in terms of solution quality, whereas the final goal is to compare both approaches in terms of CPU time.
Analysis of the LB: if we consider the whole benchmark, the LB is proved as an optimal solution in 44 out of 169 instances, representing approximately 26,036% of cases. Nevertheless, for the remaining 73,964% of cases, it's true that neither the GA nor the Hb did not meet the LB but that does not mean that both methods did not find the optimal solution. Table 1 indicates the cases where the LB is the optimal solution according to the different configurations of S-M-C generated.

The second remark that we will point out is the fact that the LB is proved to be optimal in 31 out of the whole 44 cases found out, in the subset of instances with 6-2 as the number of suppliers (6)and manufacturers (2).

Table 1. Analysis of the LB according to the numbers of s and m

S-M-C	Total number of instances	optimal instances	%opt-LB
2-2-4	36	3	8,33
2-3-4	33	0	0
3-3-4	31	3	9,68
2-6-4	34	7	20,59
6-2-4	35	31	88,57
$\sum$	169	44	

So, through this table, it is clear that the proposed LB is much more efficient with a greater number of suppliers compared to the number of manufacturers. In such case, we have normally LB = LB(M) since there are more jobs to process at the manufacturers than, at the suppliers. Further, is can be explained by the fact that with a greater number of suppliers, the idle times of machines on the manufacturer stage - which is supposed empty after the processing of the first job in the computation of $LB(J_m)$ - will be probably minimized since each manufacturer will receive an enough number of batches reducing thereby its idle time, leading to optimal or near-optimal solutions in the majority of cases. We also remark that, among the four other configurations of table 1, the LB is more efficient in case of 2 suppliers and 6 manufacturers that in the three other configurations with similar number of suppliers and manufacturers. In such case, we have normally LB = LB(S) since there are more jobs to process at the suppliers than at the manufacturers.
Evaluation of both algorithms: this section concerns the two latter goals outlined above. The average percentage deviation from the lower bound for both algorithms is computed as follows:

$$\%deviation = \frac{D_{max} - LB}{LB} * 100 \tag{6}$$

First of all, we should precise that we have performed five runs of the GA for each instance to evaluate its robustness. We began by conducting the experimentation tests according to the number of jobs. Table 2 shows the % deviation obtained with the heuristic Hb and the minimal, mean and maximal % deviation obtained by the GA in each instance among the five runs. It appears from this table that the GA performs better than Hb since, in mean, it provides smaller % deviation. Nevertheless, this table gives no information concerning the performance of both algorithms in function of the configuration (S-M-C) of the instances. Therefore, the comparison of Hb and GA has been finally conducted according to the five configurations of the numbers of suppliers and manufacturers similarly to the LB analysis. The results of CPU times for both methods and also the percentage of solved problems to optimality according to the number of suppliers and manufacturers are summarized in table 3. The first column of this table represents the configurations of the numbers of suppliers, manufacturers and customers. The next four columns concern the percentage of problems for which the LB is obtained and thus, optimality is proved for respectively, the heuristic (denoted in the table as H), the GA_{min}, GA_{mean} and GA_{max}. Similarly, the columns 6, 7, 8 and 9 illustrate the average results relative to the average percentage deviation from the LB. Finally, the two last columns are related to the computation times (CPU), in seconds. We should precise that we compute an average CPU for the whole runs of the GA. It is denoted as $GAavg$ in the last column of the table. The results of this table corroborate the remark highlighted in the previous sub-section concerning the performance of the LB according to the numbers of suppliers and manufacturers. In fact, we can observe through this table that both algorithms are much more effective when the number of suppliers is greater than the number of manufacturers in terms of percentage of solved problems to optimality and average percentage deviation from the lower bound. As instance, the heuristic Hb reaches the optimal solution in two problems out of 36 for the first configuration 2-2-4 of $S - M - C$ representing 2,77% whereas this percentage becomes equal to 77,142% for the last configuration 6-2-4 of $S - M - C$. The same observation is also verified for the GA with its three values: GA_{min}, GA_{mean} and GA_{max}.

Table 2. The comparison of Hb and GA according to the number of jobs in terms of %dev

Number of jobs	GA			Hb
	GA_{min}	GA_{mean}	GA_{max}	
20	7,467	7,467	7,467	14,803
30	8,374	9,290	10,245	14,924
40	4,313	5,513	6,454	8,768
50	4,260	4,930	5,897	11,879
60	4,253	5,154	6,447	13,890
70	3,325	4,895	6,523	14,125
80	2,504	3,628	4,988	7,391
90	3,880	5,069	6,851	11,722
100	3,245	4,278	5,692	10,097

Table 3. Summary of the evaluation of both algorithms

S-M-C	% LB is obtained				%dev				CPU(s)	
	H	GA_{min}	GA_{mean}	GA_{max}	H	GA_{min}	GA_{mean}	GA_{max}	H	GA_{avg}
2-2-4	2,77	8,33	5,55	5,55	13,335	3,829	4,913	6,189	0,056	25,637
2-3-4	0	0	0	0	8,008	4,836	5,741	6,801	0,054	63,016
3-3-4	3,225	9,677	0	0	14,011	4,663	5,702	7,033	0,021	29,364
2-6-4	5,882	17,647	5,882	5,882	3,591	2,970	3,332	3,934	0,053	13,537
6-2-4	77,142	54,285	48,571	48,571	0,502	1,279	1,281	1,291	0,015	7,028

Concerning the average percentage deviation from the lower bound, we can observe a greater robustness of the GA when the numbers of suppliers and manufacturers are different, especially when S > M. Additionally, the results show that the GA outperforms the heuristic in the four first subsets of instances (2-2-4; 2-3-4; 3-3-4 and 2-6-4). However, we remark that the heuristic is more efficient than the GA for the last subset of instances (6-2-4) with an average percentage deviation from the LB equal to 0,502% against 1,279% for the GA_{min}. The main reasons of this fact relatively similar to those described above to explain the tightness of the LB in this case. Nevertheless, the results of GA remain satisfying varying from 1,279% as the better average percentage deviation from the lower bound, obtained by the GA_{min} in the subset of instances with the configuration (6-2-4) to 7,033%, obtained by the GA_{max} in the subset of instances with the configuration (3-3-4). Hence, we may also deduce that generally the problem is much more complex to solve with an equilibrated number of suppliers and manufacturers. Finally, table 3 shows the interesting obtained CPU time concerning the heuristic Hb (see the last two columns of this table). In fact, we can clearly observe its quickness with an average computational time equal to 0,056(s) in the worst case (2-2-4).

8 Conclusion

In this paper, we considered the batch delivery constraints in addition to the eligibility and the transportation times constraints. To solve this NP-hard problem, we developed a new heuristic Hb, a GA and also a new lower bound. The conducted computational study showed the performance of both proposed algorithms, especially in terms of solution quality. It also allowed us to highlight some particular conditions for which we demonstrated that especially, the proposed LB and the heuristic are really well fitting to tackle some particular cases of this problem. There are still many interesting research topics in connection with the supply chain scheduling problem. We project in the future to extend the model by considering (i) the machine availability constraints in our supply chain model, (ii) additional actors in the SCS, like the warehouses, the retailers, etc. and (iii)

additional constraints like, the vehicles availability constraint, etc. with the particular characteristic of our model i.e. the two stage processing of jobs.

References

1. T. P. Harrison and H. L. Lee and J. J. Neale, *The practice of supply chain management: where theory and application converge*, kluwer academic publishers, 2003.
2. N. G. Hall and C. N. Potts, *Supply chain scheduling: batch and delivery*, Operations Research, Vol. 51, pp. 566-584, 2003.
3. Y. C. Chang and C. Y. Lee, *Machine scheduling with job delivery coordination*, European Journal of Operational Research, Vol. 158, pp. 470-487, 2004.
4. Z. L. Chen and G. Vairaktarakis, *Integrated scheduling of production and distribution operations*, Management Science, Vol. 51, pp. 614-628, 2005.
5. G. Pundoor and Z. L. Chen, *Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and distribution cost*, Naval Research Logistics, Vol. 52, pp. 571-589, 2005.
6. X. Wang and T. C. E. Cheng, *Machine scheduling with an availability constraint and job delivery coordination*, Naval Research Logistics, Vol. 54, pp. 11-20, 2007.
7. L. Tang and H. Gong, *A hybrid two-stage transportation and batch scheduling problem*, Applied Mathematical Modelling, Vol. 32, pp. 2467-2479, 2008.
8. M. M. Mazdeh and M. Sarhadi and K. S. Hindi, *A branch-and-bound algorithm for single-machine scheduling with batch delivery and job release times*, Computers & Operations Research, Vol. 35, pp. 1009-1111, 2008.
9. S. H. Zegordi and I. N. K. Abadi and M. A. B. Nia, *A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain*, Computers & Industrial Engineering, Vol. 58, pp. 373-381, 2012.
10. M. R. Barzoki and S. R. Hejazi, *Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains*, European Journal of Operational Research, Vol. 228, pp. 345-357, 2013.
11. Z. L. Chen, *Integrated Production and Outbound Distribution Scheduling: Review and Extensions*, Operations Research, Vol. 58, Vol. 58, pp. 130-148, 2010.
12. C. L. Li and J. W. Ou, *Machine scheduling with pickup and delivery*, Naval research Logistics, Vol. 52, pp. 1-14, 2005.
13. X. Wang and T. C. E. Cheng, *Production scheduling with supply and delivery considerations to minimize the makespan*, European Journal of Operational Research, Vol. 194, pp. 743-752, 2009.
14. T. Sawik, *Coordinated supply chain scheduling*, International Journal of Production Economics, Vol. 120, pp. 437-451, 2009.
15. R. Soltani and S. J. Sadjadi, *Scheduling trucks in cross-docking systems: A robust meta-heuristics approach*, Transportation Research Part E: Logistics and Transportation Review, Vol. 46, pp. 650-666, 2010.
16. R. L. Graham and E. L. Lawler and J. K. Lenstra and A. H. G Rinnooy Kan, *Optimization and approximation in deterministic sequencing and scheduling: A survey*, Annals of Discrete Mathematics, Vol. 5, pp. 287-326, 1979.
17. A. Vignier and J. C. Billaut and C. Proust, *Les problemes d'ordonnancement de type flow-shop hybride : etat de l'art*, RAIRO RO, Vol. 33, pp. 177-183, 1999.
18. W. Besbes and J. Teghem and T. Loukil, *A Genetic algorithm to minimize the makespan in the k-stage hybrid flow shop scheduling problem*, Foundations of Computing and Decision Sciences, Vol. 35, pp. 3-24, 2010.
19. W. Besbes and J. Teghem and T. Loukil, *Scheduling hybrid flow shop problem with non-fixed availability constraints*, European Journal of Industrial Engineering, Vol. 4, pp. 413-433, 2010.
20. C. Oguz and M. F. Ercan, *A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks*, Journal of Scheduling, Vol. 8, pp. 323-351, 2005.
21. R. Ruiz and C. Maroto, *A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility*, European Journal of Operational Research, Vol. 169, pp. 781-800, 2006.

An exact approach for solving multi-capacitated location problem based on a set partitioning formulation

Mohammed EL AMRANI
Team ROL
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes - ENSIAS -
Mohammed V University in Rabat
Madinat Al Irfane, BP 713, Agdal
Rabat, MOROCCO
xamrani@gmail.com

Youssef BENADADA
Team ROL
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes - ENSIAS -
Mohammed V University in Rabat
Madinat Al Irfane, BP 713, Agdal
Rabat, MOROCCO
yss.benadada@gmail.com

Bernard GENDRON
Interuniversity Research Center on Enterprise Networks, Logistics and Transportation - CIRRELT -
Université de Montréal
CP 6128, Succursale Centre-ville
Montréal QC H3C 3J7 CANADA
bernard.gendron@cirrelt.ca

ABSTRACT

The budget constraint multi-capacitated location problem **(MCLP)** is a generalization of the famous capacitated p-median location problem **(CPMP)**; it is characterized by allowing each facility to be open with several different capacity level. We consider **n** customers, **m** facilities and **l** capacity levels, we observe that the solution form of MCLP can be shown as a set of disjoint clusters, each one is made up of one facility and a set of customers, the capacity level used is selected according to the customers' demands. In this paper, we present the set partitioning formulation of the MCLP problem (called **MCLP-SP**), then we suggest an adapted exact solving method, which we call **NFF** (Nearest Facility First). Computational results are presented at the end using instances that we have created under some criteria of difficulties or adapted from those of p-median problems available in literature. The NFF method provides very good results for low and medium difficulty instances, but it is less effective for the more complex ones. To remedy this problem, the method will be supplemented by column generation approach.

KEYWORDS

Location, p-median, set partitioning, exact approach, column generation.

1. INTRODUCTION

Locating facilities is one of the main problems when it comes to making strategic or tactical decisions. The objective of this kind of problems is usually to minimize a cost function that can include the cost for the assignment as well as the opening cost of facilities. In many location variants, the facilities' opening cost is initially fixed by a budget constraint, in such case only the assignment cost is minimized.

Facility location has been the subject of a large number of publications in the fields of supply chain optimization and operational research. The p-median is the most famous location problem that we can find hugely in literature, the CPMP is a variant of this well-known problem subject on the capacity constraints.

Given a bipartite graph G(V, U, E) where V and U are respectively the sets of customers and facilities' nodes and E is the set of edges. V and U are each independent sets such that every edge connects a vertex in U to one in V.

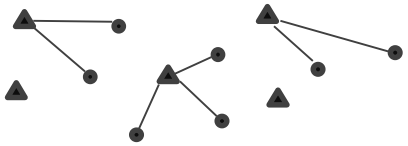


Figure 1: graphical representation of location problem

The location problem graph is composed of many connected sub-graphs, each one is also made up of either one facility solely (triangle) or a facility with a partition of customers (points).

Let E = {ij : $i \in$ V, j $\in$ U } and c_{ij} the assignment cost of the customer i to the facility j, the customer demand d_i and the facility capacity u_j are associated respectively with node $i \in$ V and node j $\in$ U.

Let x_{ij} be the binary variable associated with the edge $ij \in$ E (x_{ij} = 1 if customer i is assigned to facility j, 0 otherwise) and let y_j be the binary variable associated with the median node $j \in$E (y_j = 1 if facility j is used, 0 otherwise). The integer linear mathematical formulation is :

$$Min \sum_{i \in N} \sum_{j \in M} d_i c_{ij} x_{ij} \qquad (1)$$

$$\sum_{j \in M} x_{ij} = 1 , \quad i \in N \qquad (2)$$

$$\sum_{i \in N} d_i x_{ij} \leq u_j y_j \ , \quad j \in M \qquad (3)$$

$$x_{ij} \leq y_j , \quad i \in N , j \in M \qquad (4)$$

$$\sum_{j \in M} y_j = \quad p \qquad (5)$$

$$x_{ij} \in \{0,1\}, i \in N, \mathrm{j} \in M \qquad (6)$$

$$y_j \in \{0,1\}, \ \mathrm{j} \in M \qquad (7)$$

The objective (1) is only to minimize the assignment costs, which may occur as transport costs by a unit of distance, in this variant we are not looking to minimize the opening costs of the medians. The demand constraints (2) force each customer to be assigned to one and only one median. Constraints (3) impose that the capacity of a median must not be exceeded. Constraints (4) are used to restrict customers to be assigned to a closed facility. It can be noted that in the capacitated p-median location problem, this constraint becomes redundant with the capacity constraints; however, the experience shows that this it represents a valid high inequality to reduce the execution time. Constraint (5) is to specify that the number of medians must be equal to ***p***. (6) and (7) are the integrality constraints.

The search for p-median nodes in a network or graph is a classical location problem. The purpose is to locate p facilities (medians) in order to minimize the assignment cost of customers to facilities. The CPMP defines for each median one capacity which characterizes the service provided. The total demands for all customers assigned to one facility cannot exceed its service capacity.

In various industry field the service costs increase with its capacity used, the application of any single CPMP problem present a waste in terms of resources, the capacity of the service presented can hugely exceed the customer demands. In order to generalize the CPMP for more complex situations faced in industry, MCLP (CPMP generalization) appeared for the first time in (EL AMRANI, BENADADA, & GENDRON, 2016), defines several capacity levels for each facility. Thus a facility can be open at only one capacity level. The total demands of the assigned customers define the capacity level to use. Each opened level have a corresponding cost.

Several applications in the industry use the MCLP concept such as telecommunications, energy management, and many others. This can explain the fact that this is one of the most important known problems which can have impact on the strategic decisions.

The aim of this problem is to optimize the related transport cost of assigning customers to facilities. Each customer has a fixed demand served by a single open facility. The facilities can be used in one of many preset levels of capacities, no one can be open for more than one level at the same time. By assigning customers to facilities, we have to check that the total demands of customers served by each facility is less than its level capacity used. The sum of facility opening costs is bounded by a limited budget.

Unlike the p-median problem, widely discussed in the literature, the MCLP is a new problem that we did not find any existing study, only in (EL AMRANI, BENADADA, & GENDRON, 2016). MCLP problem is NP-complete problem because it represents a generalization of CPMP (Garey MR, 1979). Variants of the latter appeared in (Roberto Baldacci, 2001), (Luiz AN Lorena, 2003) (Edson LF Senne, 2004) (Maurizio Boccia, 2007) and (Sittipong Dantrakula, 2014). To solve the CPMP several approaches have been proposed: (Maurizio Boccia, 2007) used a cutting planes algorithm based on Fenchel cuts (Edson LF Senne, 2004) (Alberto Ceselli, 2005) and (Ceselli, 2003) proposed the application of Branch & Price and branch & Bound methods based on Lagrangian relaxation and (Luiz aN Lorena, 2003) proposed a resolution with column generation.

In this study of proposing new formulation of MCLP based on set partitioning, we will test the new model by using resolution methods, namely the Branch and Cut used by CPLEX solver and a new exact resolution approach based on column generation called NFF.

The NFF proposed approach to solve MCLP consists on assigning each customer to the nearest possible facility, to do that we will sort all facilities by the assignment cost. At the first iteration, all customers are assigned to the nearest facility. Then we compare the sum of the demands assigned to each facility by its capacity level used. If this iteration provides one solution, so the solution found is optimal, otherwise we allow customers to be assigned to the next nearest facilities until arriving at a feasible solution.

This paper is organized as follows. In section two, we discuss the new formulation of the MCLP based on set partitioning, the third one is devoted to the solving methods, namely the Branch and cut used by CPLEX and a new suitable exact approach, called NFF. Computational results are presented in the second last section before the conclusion.

2. FORMULATION

The MCLP problem is a location problem with capacity where facilities can be used at several levels, each level is characterized by a certain capacity to respect and the facility can open only one level at once. In the mathematical formulation, we will need to create additional variables and notations and modify the constraints of the previous problem such that each facility must respect the maximum capacity of the selected level.

The mathematical formulation is as follows:

$$Min \sum_{i \in N} \sum_{j \in M} d_i c_{ij} x_{ij} \qquad (8)$$

$$\sum_{j \in M} x_{ij} = 1, \quad i \in N \qquad (9)$$

$$\sum_{i \in N} d_i x_{ij} \leq \sum_{k \in K} u^k y_j^k \ , \quad j \in M \qquad (10)$$

$$\sum_{k \in K} y_j^k \leq 1, \quad j \in M \qquad (11)$$

$$x_{ij} \leq \sum_{k \in K} y_j^k, \quad i \in N, j \in M \qquad (12)$$

$$\sum_{j \in M} \sum_{k \in K} f^k y_j^k \leq B \qquad (13)$$

$$x_{ij} \in \{0,1\}, \quad \in N, \quad \text{j} \in M$$

$$y_j{}^k \in \{0,1\}, \text{j} \in M, k \in K$$

Where $\boldsymbol{K}$ denotes the set of levels, $\boldsymbol{u^k}$ represent capacity of level $\boldsymbol{k}$, $\boldsymbol{f^k}$ is the opening cost associated with level $\boldsymbol{k}$, **B** is the limit budget on the total opening costs.
$\boldsymbol{y_j^k}$ is decision binary variable that is one if and only if the facility $\boldsymbol{j}$ is opened and used at the level $\boldsymbol{k}$.

This model generalizes the capacitated p-median problem including the capacity levels concept. This new data appears in the capacity constraints (10) and at the budget constraint (13), because the facilities have different opening costs. (12) are also additional constraints which represent valid inequalities that cut the feasible region. (10) are capacity constraints and (11) force the facility to be open at one level at most.

In this section, we choose to re-formulate our MCLP as a set partitioning problem, this new formulation is more suitable to well effective solving method, namely the NFF exact method described above. Given a bipartite graph G(V, U, E) where V and U are respectively the sets of customers and facilities' nodes and E is the set of edges. V and U are independent sets such that every edge connects a vertex in U to one in V.

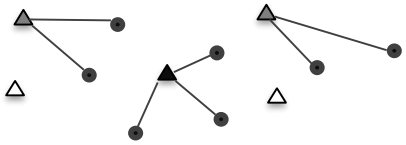


Figure 2: graphical representation of MCLP problem

The MCLP problem graph is composed of many connected sub-graphs, the facility (triangle) can be operated in many levels: white triangle for closed facility, grey one for the level one or black for a level with the biggest capacity, the capacity level is chosen according to customers' demands. We can easily see that the final solution will be in form of a set of pair (facility, clusters subset), this pair will be called "cluster". All of these clusters will build the feasible region of this new model. This above new formulation will be called MCLP-SP

$$Min \sum_{j=1}^{m} \sum_{p=1}^{l} c_p^j x_p^j \qquad (\mathbf{14})$$

$$\sum_{j=1}^{m} \sum_{p=1}^{l} a_{ip} x_p^j = 1, \quad i \in N \qquad (\mathbf{15})$$

$$\sum_{p=1}^{l} b_{jp} x_p^j \leq 1, \quad j \in M \qquad (\mathbf{16})$$

$$\sum_{j=1}^{m} \sum_{p=1}^{l} f_p x_p^j \leq B \qquad (\mathbf{17})$$

$$x_p^j \in \{0,1\}, p \in \{1,2,..,l\}, j \in M$$

$S = \{S_1, S_2, \ldots, S_l\}$ indicates the set of subset of N;
$S \subseteq \mathcal{P}(N)$.
$\boldsymbol{c_p^j}$ is the total assignment cost of the customer subset $\boldsymbol{S_p}$to the facility $\boldsymbol{j}$.

$$A = [\boldsymbol{a_{ip}}]_{n \times l} \; with \; \boldsymbol{a_{ip}} = \begin{cases} 1 & si \; \boldsymbol{i} \in S_p \\ 0 & sinon \end{cases}$$

Satisfying $\sum_{i \in N} \boldsymbol{d_i a_{ip}} \leq Max_k\{\boldsymbol{u^k}\}$

$$B = [\boldsymbol{b_{jp}}]_{m \times l} \; with \; \boldsymbol{b_{jp}} = \begin{cases} 1 \; if \; \boldsymbol{j} \; is \; assigned \; to \; S_p \\ 0 \quad otherwise \end{cases}$$

$\boldsymbol{f_p} = Min_k\{\boldsymbol{f^k} | \sum_{i \in N} \boldsymbol{d_i a_{ip}} \leq \boldsymbol{u^k}\}$ and $\boldsymbol{c_p^j} = \sum_{i \in S_p} \boldsymbol{d_i c_{ij}}$
$\boldsymbol{x_p^j}$ is a decision binary variable that is equal to one if and only if cluster build by facility **j** and customer subset $\boldsymbol{S_p}$ belongs to the solution.

It may be noted that in this formulation the constraints (8), (9), (13) are respectively equivalent to (14), (15), (17) and the constraints (9), (10), (11) appeared explicitly in the initial formulation become implicitly considered. Constraint (16), which is not declared in the first model explicitly, prevents us to have, in the feasible solution, the same facility in more than one cluster.

If **S** is the set of all subsets of **N**, the formulation can give an optimal solution to the MCLP-SP. However, the number of subsets may be very huge, and an exact resolution with a solver such as CPLEX becomes impossible for a reasonable time. Therefore, we must think of a way to solve this problem with a reduced computational time. To circumvent this situation, we propose in the next section a method to prohibit the assignment of customers to facilities supposed too far, limiting the number of produced clusters.

3. SOLVING METHODS

To proceed with solving the problem, we will suggest to test the new formulation with an exact method well known, that of Branch and Cut, although this method presents difficulties for large instanced, it will serve to validate our new modeling. The adapted method for the present formulation will be details afterwards.

3.1. Branch and cut

The Branch and cut is a combination of two algorithm into one, namely the Branch and Bound and cutting plans.

Algorithm

1- At each top of the resolution tree, a linear relaxation is solved by the method of cuts.
2- If the solution is feasible (or if it is above a known solution), the summit declared sterile.
3- If all pending summits are sterile, STOP, the best solution found is optimal.

4- Otherwise, choose a non-sterile pendant summits, select a fractional variable xi, and consider two subproblems by fixing xi at 1 and xi at 0 (Branching)
5- Solving every problem by generating new unsatisfied constraints (cutting). Go to step 3.

The branch and cut algorithm is more efficient because it is reliable and more fast than the branch and bound.

In this section, we present the resolution method, which consists of banning the creation of clusters with a too high total assignment cost.

The first step in solving this problem is the generation of clusters. In the first place, we classify, for each customer, the facilities in ascending order of assignment costs. Then we assign to each customer the least expensive facility. After this initial assignment, and for all iterations that follow, we compare for each facility the sum of customer demands assigned with the different capacity levels. If the total of these demands exceeds the highest level, then the cluster is rejected by unfeasibility. Otherwise, we create the clusters with the lowest level satisfying all demands. Thus, the opening cost of the cluster is that of the selected level, and the assignment cost of the cluster is the sum of all assignment.

The solution found in this step, if it exists, can only be optimal, because it presents an ideal case, where each customer is assigned to the nearest facility, which means the least costly. Otherwise, we regenerate the clusters by adding the second closest facility for a customer in terms of assignment cost, and so on until achieve a feasible solution or arrive at a number of clusters considered too high for reasonable computational time. If necessary, we use the column generation method.

To explain more the logic of the method, we detail the steps in the following algorithm:

Algorithm

Generate_ clusters

If the set L does not exceed a maximum number (5000 clusters)

1- Search all assignments already made or clusters already created, each facility with its customers associated.
2- Let n the number of customers,for each pair (facility, part of customers).

If the sum of all demands exceeds the maximum capacity

a- The cluster with ***n*** customers is rejected by unfeasibility.
b- If ***n>0***we create ***n*** clusters each one with ***n-1*** customers. Otherwise STOP.
c- Comme back to (2-)

Otherwise

i = 0 ;

a- If the sum of demands does not exceed the capacity (***i***), we add this cluster in the set L with the capacity (***i***) and the corresponding opening cost and total assignment costs.
b- Otherwise ***i←i+1***, come back to (a-) ;
c- Remove duplicates and STOP.

EndIf

EndGenerate_ clusters

0- Initially there are a set N of customers, a set M of facilities and a empty set L.
1- For each customer, sort the facilities in ascending order in terms of assignment.
2- Assign to each customer the nearest facility.
Repeat |N|*(|M|-1) times
3- Generate theclusers.
4- If solution is found, it can only be, STOP. (thealgorithmestops by optimaly).
5- Otherwise, add the best assignment (customers, facility) and come back to (4-).
EndRepeat
(if no solution found)
Repeat |L| times, and let ***p*** = 1
6- Develop the cluster ***p*** with all possible combinations.
7- If one solution is found, STOP. Otherwise, ***p←p+1***, come back to(6-)

Otherwise we move to a resolution with columns Generation.

EndAlgorithm

This algorithm always finishes by find the solution, if such a solution exists.

The following diagram describe the sequence of the algorithm:

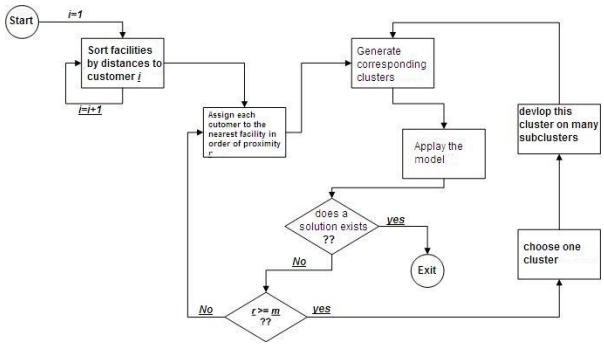


Figure 3: Descriptive diagram of the algorithm

4. COMPUTATIONAL RESULTS

We did not find any reference in the literature on MCLP. This model therefore has, as we know, no existing instances for testing and comparison. For this, we will in the context of this work create instances using semi-random values based on a justified and appropriate choice. We will also use some p-median instances to supplement the computational tests, since the MCLP is a generalization of the latter.

We turn the two algorithms using the CPLEX solver. We use the version 7 of Java and version 12 of Cplex and we run the program on a machine i7-2600 CPU @ 3.40 GHz.

Our data set consists of five instance classes which have five levels of difficulty (easy, medium, difficult, very difficult and complex). The difficulty of these instances is based on the size of the problem, which is often measured by the number of customers. In contrast, the number of facilities and the number of levels used have a low impact on the size of the problem. Each difficulty level contains several test instances. These instances also represent difficulty sub-levels. They are characterized by the dispersion of the points (customers compared to plants) and also by the amount of available resources. Experience shows that the difficulty of the problem varies proportionally to the variance of customer-facility distances and customer demands. At the same time, it varies inversely with the budget allocated to the opening of facilities, and to their capacity levels. Thus increasing the difficulty, while keeping the feasibility, we multiply the number of iterations needed to find the optimal solution.

This algorithm is composed of two parts: the first concerns the reformulation of the mathematical model and the adaptation of data in the form of clusters by manipulation with JAVA. In the second we solved the new model using the CPLEX solver.

The parameters of these instances are described in the following table:

DL : Difficulty level (E : easy, M : medium, D : difficult et VD : very difficult)
NC : Number of Customer
NF : Number of Facility
NL : Number of capacity Levels
NCL : Number of Clusters used to find solution.

For simplification reasons for large size problems, we add at each iteration, a set of clusters instead of adding one. This means that the number of clusters with which we find the feasible solution is not minimum, and the first solution found when running is not necessarily optimal, but very near the optimal one, so we have to wait in such case finishing the calculation.

The following table lists the various instances used and the computational results found:

Instance	DL	NC	NF	NL	Branch & Cut		NFF			
					Obj	CPU (s)	NCL	Obj	CPU (s)	GAP
F1	Easy	**10**	3	2	103	0,04	14	103	0.15	0%
F2		**10**	5	2	586	1,26	473	586	0.20	0%
F3		**20**	5	3	387	3,04	489	387	0.15	0%
F4		**30**	8	3	334	4,75	734	334	0.16	0%
M1	Medium	**50**	4	3	8826	0,07	69	8826	0.55	0%
M2		**50**	6	4	2612	2,40	1014	2612	1.52	0%
M3		**70**	6	4	7079	10,55	1116	7079	2.43	0%
D1	Difficult	**100**	10	5	12618	0,124	55	12618	0.45	0%
D2		**100**	15	5	1587	0,38	259	1587	0.52	0%
D3		**200**	15	8	90312	0,592	-	-	-	-
TD1	Very Difficult	**300**	25	10	34175	0.842	1214	34584	1.14	1.18%
TD2		**300**	30	10	25037	0.967	2034	25146	1.26	0.43%
TD3		**402**	30	12	42233	0.827	3248	43011	2.35	1.80%
TD4		**402**	40	12	39805	1.872	-	-	-	-
C1	Complex	**500**	50	4	25452	11,87	4365	26314	5.28	3.27%
C2		**1000**	100	4	46719	143,53	4813	47624	12.69	1.90%
C3		**3038**	600	10	-	-	3128	68421	52.2	-
C4		**3038**	700	10	-	-	-	-	-	-
C5		**3038**	1000	10	-	-	-	-	-	-

According to the computational results, the NFF approach is based on the concept of attacking the simplest first; we start by assigning each customer to the nearest facility, if a solution is found it will be optimal, if necessary we test the next facility. And so on until arrive at the solution. We notice from the table of results that the approach works very well with a very reasonable execution time for easy instances in terms of dispersion points and in terms of resource availability, but it is less effective for difficult instances. It is for this reason that we will harness the power of our method in such instances and we will complete the approach by the column generation method, if the method is not successful after the generation of 5000 clusters.

In these results we have defined the Gab for only difficult instances, the should normally always equal zero, because the method always gives optimal solutions. However, for reasons to reduce the execution time, we sometimes accept results close to the optimal. That said, the loss of optimality can occur under two scenarios: either skipping a few iterations, or if we fail to generate all possible clusters because of a technical

problem, on memory, that we plan to correct in the next version.

5. CONCLUSION

In this paper we introduced the multi-capacitated location problem with budget constraint. We have proposed a mathematical formulations based on set partitioning. We propose Branch and Cut to validate the new formulation, and finally the NFF method that gives the optimal solution for all instances in which the difficulty level is medium or difficult. The resolution method is less effective for more complex instances. An alternative for such instances would be to use the column generation method.

6. BIBLIOGRAPHY

Alberto Ceselli, G. R. (2005). A Branch-and-Price Algorithm for the Capacitated p-Median Problem.

Ceselli, A. (2003). Two exact algorithms for capacitated p-median problem.

Edson L.F. Senne, L. A. (2004). A branch-and-price approach to p-median location poblems.

EL AMRANI, M., BENADADA, Y., & GENDRON, B. (2016). Generalization of capacitated p-median location problem : modeling and resolution. *International IEEE Conference of Logistics Operations Managment* (pp. 1-6). FES: IEEE.

Garey MR, J. D. (1979). Computers and intractability : a guide to the theory of NP completeness.

Luiz A.N. Lorena, E. L. (2003). A column generation approach to capacitated p-median problems. http://www.lac.inpe.br/~lorena/instancias.html.

Maurizio Boccia, A. S. (2007). A Cut and Branch Approach for the Capacitated p-Median Problem Based on Fenchel Cutting Planes.

Roberto Baldacci, E. H. (2001). A new method for solving capacitated location problems based on a set partiotioning approach.

Sittipong Dantrakula, C. L. (2014). Applied p-median and p-center algorithms for facility location problems.

A memetic algorithm with population management for the problem of production, maintenance and quality integration

M. Nourelfath[1], H. Beheshti-Fakher[1] and M. Gendreau[2]

1. Université Laval, CIRRELT, Canada

2. École polytechnique de Montréal, CIRRELT, Canada

Abstract.
In this paper, we develop a memetic algorithm with population management to solve the integrated planning of production, maintenance, and quality in multi-product, multi-period imperfect systems. The proposed memetic algorithm employs the Nelder-Mead algorithm along with an optimization package for exact determination of the values of several decision variables in each chromosome evolution. The method extracts not only the positive knowledge in good solutions, but also the negative knowledge in poor individuals to determine the algorithm transitions. The method is compared in terms of the solution time and quality to several heuristic methods.

Keywords: combinatorial optimisation, Integrated model, Metaheuristics

1 Introduction

Existing literature on integration of production, maintenance and quality, highlights the significance of improvements achieved by the joint models [1], [2]. This approach is a beneficiary of the whole system [3], [4], and improves the productivity [5] or the profitability [6], [7].
However, such joint models are generally non-linear and difficult to be solved in a reasonable time or with rational computational effort. Exploiting the benefits of integrated models necessitates the development of efficient solution methods.
In [9] the authors compared the efficiency and accuracy of exact and heuristic algorithms and concluded that the solution time of exact methods exponentially increases by the problem size, whereas the solution time in heuristic methods is almost constant and considerably low. They proposed heuristic methods for large problems.
The evolutionary algorithms (EAs) have been widely and successfully used in solving difficult and large problems. Among the existing EAs, the memetic algorithms (Mas) have been increasingly employed in solving problems in operations research and computer sciences [11]. Memetic algorithms exploit the benefits of integrating population-based search and local improvement methods. Such a combination considerably improved the power of EA [12]. Efficiency and robustness of MAs in solving complex planning and scheduling problems and their capability in balancing between exploitation and exploration are our motivation in adopting a memetic algorithm to solve the integrated production-maintenance problem.
The literature review first, illuminates the value of joint scheduling and shows that just a few papers have addressed the integration of production planning and maintenance scheduling in imperfect systems, and second, underlines the gap between the theory and application of joint models. To the best of our knowledge, very few implementations of joint models have been reported yet.
We develop a solution method based on memetic algorithms for an integrated problem from the literature. In the meantime, we implement innovative strategies to improve the algorithm efficiency.

For the first time in similar problem, we not only make use the properties of good solutions (known as positive knowledge) but also, the characteristics of poor solutions (negative knowledge) in algorithm transitions. The proposed method uses population management to maintain the diversity during the solution process. Hybridization of genetic algorithm with local search methods is carried out in two points. First, using CPLEX to solve a part of the model that is a linear program, and then, we use Nelder-Mead or tabu-search algorithms to enhance its performance.
The rest of this paper is as follows. In section 2, the problem definition is presented. Integrated planning model is explained in section 3. Section 4 presents the solution method and search management tools and section 5 provides a comprehensive examination of the algorithm's performance and comparisons. Section 6 includes our concluding remarks.

2 Problem definition and solution method

In this section, the joint production and maintenance model in imperfect systems and a solution method are presented.

2.1 Joint scheduling model

A capacitated lot-scheduling problem in a multi-machine, multi-product, multi-period context is considered. Demand for the products in each period is known and unsatisfied orders will charge the backorder cost to the system. Over production imposes the inventory holding cost and switching to a new product type on a machine is accompanied with a setup charge. Defective rates and expected availability of machines depend on the preventive maintenance (PM) and quality inspection plans. The total cost of the system to be minimized is the sum of production, maintenance, and quality costs. At the beginning of each period the machines are in normal state, but by progressing the time, the machines deteriorate and so, the failure probability increases. The imperfect preventive maintenance with multiple PM options for each machine is considered and it is assumed. We assume a maintenance opportunity at the beginning of each period and the solution of the model determines the optimal PM level for each machine in each period.
The three possible states of each machine are (1) the normal (or in-control) state with negligible defective rate, (2) the shifted (or out-of-control) state with higher nonconformity, and (3) failed state. Machine failures are self-announcing and a PM team will perform a minimal repair that brings the machine in operation without influencing on its age. During each period, several error-free process inspections are considered to detect the machine's state. By detecting a shifted state (1) the machine will be corrected that imposes an adjustment cost, and (2) the sub-lots produced in that interval will be quality checked to separate the defective items. Such nonconforming products will be reworked before sending to a customer. The optimal number of quality inspections depends on the PM plans and the machine. Given the number of process inspections, the length of intervals will be determined such that the integrated hazard over all intervals is constant [12].

2.2 Solution method and its features

Memetic algorithm [13] (also called *hybrid genetic algorithms* or *genetic local search*) is a population-based heuristic that combines GA with local search methods. A literature review of memetic algorithms can be found in [14]. The literature also shows the impact of preserving the population diversity on efficiency of heuristics [15]. An impressive increase in the algorithm's efficiency is demonstrated by implementation of population management strategies in population-based heuristics [16]. The authors proposed a mechanism that addresses both the objective value and its contribution to population diversity in the evaluation of individuals.
In this paper, a memetic algorithm with population management for the integrated production-maintenance problem is proposed. The algorithm employs CPLEX optimization package to solve the linear part of the problem, and in the meantime, it exploits Nelder-Mead method to improve the

performance. Population management strategies are used to well-organize the solution process and both positive and negative knowledge (information extracted from better and worse solutions) are used in the solution process. Intensification of good solutions, diversification of the population, adaptive control of the algorithm parameters, and survivor selection based on the contribution of individuals in the population heterogeneity are the implemented strategies. Performance indicators (the solution time and quality, and the algorithm robustness) are used to compare it to other heuristics or different configurations of MAPM.

3 Integrated planning model

3.1 Cost of the production system

Total cost of the production system (TC_{PS}) is the sum of processing cost, setup cost, inventory holding costs, and backorder cost for all combinations of products and machines. We have:

$$TC_{PS} = \sum_{t\in T}\left(\sum_{p\in P}\sum_{m\in M}\left(x_{pmt}\pi_{pm} + S_{pmt}s_{pm}\right) + \sum_{p\in P}\left(I_{pt}h_p + B_{pt}b_p\right)\right), \quad (3.1)$$

where p and P are the index and number of products, m and M are the index and number of machines, t and T are the index and number of periods, x_{pmt} is the production level of product p on machine m in period t, π_{pm} is the processing cost of product p on machine m. Also, S_{pmt} is the binary variable of setup of product p on machine m in period t, s_{pm} is the setup cost (product p on machine m), I_{pt} is inventory level (product p in period t), h_p is the unit holding cost (product p), B_{pt} is the backorder level of product p in period t, and b_p is the backorder cost.

Binary variable of setup () and production level (x_{pmt}) are linked together, so $x_{pmt} \leq S_{pmt}g_{pm}$.

The total lot-size of product p in period t (LS_{pt}) is given by $LS_{pt} = \sum_{m\in M} x_{pmt}$ and the balance equation between production, inventory, backorder, and demand is given by $I_{pt} - B_{pt} = I_{pt-1} - B_{pt-1} + LS_{pt} - d_{pt}$ (d_{pt} is the demand for product p in period t).

The available production time (APT_{mt}) of machine m in period t is the remaining time in period after minimal repairs, so $APT_{mt} = L - NF_{mt}TMR_m$, where L is the length of period, $NFmt$ is the average number of failures (machine m in period t), and TMR_m is the time of minimal repair. Total production on a machine is constrained to the available production time; $\sum_{p\in P} x_{pmt} \leq APT_{mt}\, g_{pm}$.

3.2 Cost of the maintenance system

Let us assume that the time-to-shift function follows a Weibull distribution with parameters λ_m and φ_m i.e. $s_m(t) = \lambda_m \cdot \varphi_m \cdot t^{\varphi_m - 1} \cdot e^{-\lambda_m t^{\varphi_m}}$ and its cumulative function is $S_m(t) = 1 - e^{-\lambda_m t^{\varphi_m}}$, so the conditional probability of a shift at age t given the machine was in control at age t_0 is $s_m(t|t_0) = \frac{s(t)}{1-S(t_0)} = \lambda_m \cdot \varphi_m \cdot t^{\varphi_m - 1} \cdot e^{-\lambda_m\left(t^{\varphi_m} - t_0^{\varphi_m}\right)}, t > t_0$. Also, the time-to-failure function is also a Weibull function with parameters θ_m and ρ_m, i.e. $f_m(t) = \theta_m \cdot \rho_m \cdot t^{\rho_m - 1} \cdot e^{-t^{\rho_m}}$ and its cumulative distribution is $F_m(t) = 1 - e^{-t^{\rho_m}}$. Considering the Nakagawa's imperfect maintenance model [16] and linear link between maintenance cost and improvement in the machine conditions, the total cost of the maintenance system is $TC_{MS} = \sum_{m\in M}(PMC_t + MRC_t)$. In this equation, $PMC_t = \sum_{m\in M} CPM(PM_{mt})$ is the cost of PM of level PM_{mt} (machine m in period t), and $MRC_t = \sum_{m\in M} NF_{mt} \cdot CMR_m$ is the corrective maintenance cost with CMR_m as the cost of a minimal repair. Maintenance times are calculated such that the integrated hazard over maintenance-inspection intervals is constant. The maintenance budget constraint indicates that the total

maintenance budget assigned to PM in each period is limited to the available PM budget in that period, i.e. $\Sigma CPM(PM_{mt}) \leq PMB_{mt}$. For more details of calculation of the age of machine in periods and intervals, one can refer to [8].

3.3 Cost of the quality system

The total cost of the quality system is the sum of inspection cost, quality checking cost, reworking cost, and the process adjustment cost. The inspection cost for machine m in period t is $IC_{mt} = NI_{mt} \cdot v_m$, where NI_{mt} is the number of quality inspections of machine m in period t and v_m is the cost of an inspection. With PS_{mt} the probability of a quality shift on machine m in period t, β_p the cost of quality checking of product p, and y_{mjt} as the age of machine m at the end of interval j in period t, the expected quality checking cost for product p over all intervals is $QC_{mt} = \left(1 - (1 - PS_{mt})^{1/NI_{mt}}\right) \cdot \sum_{p \in P} x_{pmt} \cdot \beta_p$. If a shift occurs at age τ in interval j, from this instant of time until the end of the interval, the machine will work in a shifted state. Given the number of process inspections (NI_{mt}), the expected shifted state time that machine m operates in out-of-control conditions in period t, is $ESST_{mt} = \left(1 - (1 - PS_{mt})^{1/NI_{mt}}\right) \cdot \int_{W_{mt}}^{Y_{mt}} (Y_{mt} - \tau) \cdot s_m(\tau | W_{mt}) d\tau$. Hence, the total expected reworking cost for machine m is given by $TRC_{mt} = \frac{ESST_{mt}}{APT_{mt}} \cdot \sum_{p \in P} x_{pmt} \cdot \alpha_{pm} \cdot R_p$, where α_{pm} is the defective rate of product p in shifted state of machine m and R_p is the reworking cost of product p. Finally, considering the probability of shift in intervals, total expected cost of machine adjustments in period t is $TAC_{mt} = NI_{mt} \cdot AC_m \cdot \left(1 - (1 - PS_{mt})^{1/NI_{mt}}\right)$. Therefore, the cost of the quality system is $TC_{QS} = \sum_{t \in T} \sum_{m \in M} (IC_{mt} + QC_{mt} + TRC_{mt} + TAC_{mt})$.

3.4 Integrated model and the search space

The integrated model minimizes the total cost of the system ($Min\, TC_{PS} + TC_{MS} + TC_{QS}$). The model constraints are (a) the link between setups and production levels, (b) the balance equation between lot-sizes, inventories, backorders, and demands, (c) the production capacity by the available production times, and (d) the maintenance budget constraint.

The integrated model presented in this paper is a nonlinear problem with complicated evaluations. The sources of its difficulty are originated from the challenging interactions between maintenance and production from the one hand, and between maintenance and quality from the other hand. Also, determining the optimal number of process inspections minimizing the quality cost (eq. 4.25) is very hard. The number of possible scenarios for maintenance is $\prod_{m=1}^{M} (Q_m)^T$, where, Q_m is the number of PM levels for machine m, and T is the number of periods. With decision variables related to production and quality systems, size of the solution space is very huge, which justifies the need for efficient solution methods.

4 Solution methods

4.1 MAPM algorithm

The process flow of MAPM is presented in Fig. 1. In the following sections, the operators and ingredients of the algorithm are explained.

- Parameter calibration
- Generate the initial population
- Do (Solution process)
  - Evaluate the population diversity
  - Update the dynamic parameters
  - Do (Generation of the next population)
    - Generate a child by
      - Crossover
      - Intensification of top-half solutions of the population
      - Diversification of the whole population
    - Perform local search on the child
    - Add the child to the population
  - Loop until the size of the population is doubled
  - Perform survivor selection
- Loop until the stop condition is satisfied
- Return the best solution

Fig. 1: Process flow of MAPM.

4.2 Solution encoding and population initialization

A vector of integers representing the PM level for each machine in each period is considered as the solution encoding approach. Given such a vector, the maintenance cost can be evaluated and the lot-sizing model with quality inspections reduces to a linear mixed-integer problem. We use Cplex to solve this MIP model. The population is determined in the calibration process (between 10 and 50 and the initial population is generated according to a uniform random process.

4.3 Population diversity and distance measurement method

To prevent premature convergence of the algorithm, the population diversity is controlled to stay above a minimum allowed range. Each time it falls below the limit, certain parameters of the algorithm are modified such that it generates more diverse solutions. In this paper, Shannon entropy and the general form of the distance function; *Minkowski distance method* are utilized to quantify the population diversity and the distance between chromosomes [8]. The distance between solution *A* and population *P* is defined as the normalized minimum distance between *A* and the population members, i.e. $D(A,P) = \min_{p \in P} \Delta(A,p)$.

4.4 Neighborhood structure, parent selection and crossover

Solution *B* is a neighbor of solution A if they are in the neighborhood circle with radius $\Delta(A,B) = \Delta \cdot L^{-1/r}$. The selection process takes the advantage of *tournament method* with selection pressure ϕ and picks the parents by a uniform probability function. Selection is based on the fitness of individuals. Then, using the uniform crossover, the two parents are combined to generate a child.

4.5 Intensification and diversification strategies

The objectives of intensification and diversification are to ensure that the best solution of the current search space is found and also, to force the search process to investigate not-visited areas. Intensification uses the data of top-half solutions, whereas diversification considers the whole population. In case of intensification, the solution with the minimum distance, and in case of

diversification, the solution with the maximum distance is selected. The intensification-diversification power; ζ is the number of random solutions that compete in the selection process.

4.6 Local search and Nelder-Mead methods

The proposed MAPM comprises the integration of GA and local search methods in two phases of the algorithm, first; the evaluation of each chromosome is comprised of solving a mixed integer problem with the existing methods; and second, we make use of the Nelder-Mead algorithm to improve the performance of the EA. The exact values of variables concerning the lot-sizing problem are evaluated using the Cplex optimization package that solves the related MIP model. Fig 2 shows the algorithm flowchart. The objective in NM is to exploit the information from several neighbor solutions in transitions toward the local optima. In this method, we consider not only better solutions (positive knowledge), but also the worse solutions (negative knowledge) to introduce three candidate moves. Then, the best answer among them is considered as the algorithm transition.

Given a solution p to be improved:

1. Repeat
   - 1.1. Generate the sub-population P (N neighbor points of p) and evaluate them
   - 1.2. Determine the improvement (reflection) vector D
     - For each gene position $j \in \{1,..., L\}$
     - For each solution $\hat{p} \in P$
     - If $\hat{p}$ is better than p then $d_j = \hat{p}_j - p_j$ ($\hat{p}_j$, p_j, and d_j are respectively j^{th} elements of $\hat{p}$, p, and D)
     - Else $d_j = p_j - \hat{p}_j$
   - 1.3. Determine the reflected point; X_R, the expanded point; X_E, and the contracted point; X_C and modify them if needed
     - i. XR = p + D (extended point)
     - ii. $XE = p + 2 \times D$ (expanded point)
     - iii. XC = best solution in P (contracted point)
   - 1.4. Move to the best solution among the candidates (p = max $\{X_R, X_E, X_C\}$)
2. Loop until the *Stop Condition* is satisfied (number of attempts that fail to improve the solution)

Fig. 2: Nelder-Mead algorithm.

4.7 Survivor selection and algorithm calibration

Let us consider that $R_f(p)$ and $R_d(p)$ are respectively the rank of chromosome p in the population, based on its fitness and its diversity contribution. The biased fitness of p is $B(p) = \gamma \times R_f(p) + (1- \gamma) \times R_d(p)$, where γ is the bias parameter ($0 \leq \gamma \leq 1$). Selection of survivors is based on the biased fitness and the parameter γ increases during the algorithm execution such that, at the final phases of the solution process, higher priorities are yielded to the better solutions.
The values of the algorithm parameters (λ, Φ, ζ, It_{NI}, and r) are selected using the well-known meta-GA calibration method.

5 Computational results

This section illustrates the performance of the proposed algorithm and compares it to existing heuristics. Also, we show the advantage of population management in MAPM. The algorithm is

implemented in VB.net and the numerical tests are performed on an Intel core i7 – 3.4 GHz with 16 GB of RAM.

5.1 Sample Problem

Let us first consider a problem with M = 3 machines, P = 2 products, T = 6 periods, and the length of periods L = 1. Each machine has Q_m = 4 PM levels and the maintenance budget limitation for all periods is PMB_t = 1000. The problem data are given in Table 1.

Table 1: Problem data.

		s_{pm}		α_{pm}		CMR_m	TMR_m	v_m	λ	Φ	θ	ρ	W_{m0}	AC_m	CPM (Thousands)
	Product	1	2	1	2										
Machine	1	40	-	0.6	-	800	0.02	50	1	2.5	1	2.5	2	40	3000,500,200,0
	2	30	10	0.4	0.5	700	0.01	30	0.177	2.5	0.177	2.5	2	20	5000,500,200,0
	3	-	35	-	0.8	900	0.015	40	0.064	2.5	0.064	2.5	2	30	4000,600,300,0

		g_{pm}		π_{pm}		p	h_p	b_p	β_p	R_p	d_{pt}					
	Product	1	2	1	2						1	2	3	4	5	6
Machine	1	2500	-	6	-	1	2	25	1	2	3500	4000	1500	2500	1000	5000
	2	1000	1500	8	9	2	3	40	2	3	2500	2000	1500	1500	3500	3500
	3	-	3000	-	10											

Moreover, a larger version *P2* (with T = 12 periods) is considered. *P2* is the replication of the data of *P1* (demand of period 7 is the same as period 1, etc.). A set of 50 random problems is also generated to compare the performance and robustness of the algorithm. These samples are created by uniform randomization, where $T \in (1\ldots12)$, (M, P, and Q) $\in (1\ldots5)$, other data between 0.5 to 1.5 times the average of data given for the first product, first machine, in the first period.
All the test problems and size of the search spaces are summarized in Table 2.

Table 2: Sample problems.

Problem	Description	Problem size
P1	The sample problem given in the chapter III (Section 3.5)	6.9×10^{10}
P2	Larger version of *P1* with 12 periods	4.7×10^{21}
P3	Set of 50 random problems	$<8.7\times10^{41}$

5.2 Parameter settings for the joint scheduling problem

The values of the algorithm parameters are set using the meta-calibration method of section 4.3.11. With slight modifications, the parameter values; (λ, Φ, ζ, It_{NI}, r) related to the chromosome lengths ($T\times M$) are:

- (10, 3, 5, 10, 2) for $T\times M \leq 10$
- (15, 3, 5, 20, 2) for $10 < T\times M \leq 20$
- (20, 5, 10, 30, 2) for $20 < T\times M \leq 30$
- (40, 6, 25, 50, 2) for $30 < T\times M \leq 50$

The size of the tabu list in TS is set to the chromosome length, mutation probability in MA is p_m = 0.05, and if not indicated, the considered solution time is $5\times T\times M\times Q$.

5.3 The best solution of the test problems with different algorithms

The solutions found by the MAPM-NM for all the test problems are the best solutions. The methods developed in this section are able to solve various types of the integrated problems.

5.4 Robustness of the algorithm

Capability of the algorithm and the solution quality is also investigated in solving the set of 50 random problems (P5) using MAPM-NM, MAPM-TS, TS, and MA algorithms. For each problem, the gap to the best solution is calculated and the average gaps are shown in Table 3.

Table 3: Average gaps to the best solution in 50 random problems.

	MAPM-NM	MAPM-TS	TS	MA
Success rate	73.3%	26.7%	6.7%	20.0%

Accordingly, MAPM-NM yields the smallest gap among four algorithms and so, it can be conveniently used to solve the joint problem. Only in 8 out of 30 problems, the solution found by MAPM-NM was very close but not the best solution. By increasing the solution time, the gap decreases, but almost in all cases, this algorithm yields the better solutions. The results of 30 replications of MAPM-NM, MAPM-TS, and TS for P1 with two different solution times are shown in Fig 3 and 4. In both cases, MAPM-NM outperforms MAPM-TS. The two variants of MAPM perform better than tabu-search algorithm and as expected, increasing the solution time has resulted in the improvement of the final solution (smaller total costs). The minimum, average, maximum and variance of the solutions are listed in Table 4. Similar results were obtained in different solution times and problems, therefore, we remark that the proposed algorithm is robust and, integration of genetic algorithm with a Nelder-Mead method has resulted in a better performance.

Table 4: Solution data in different execution times (30 replications; *P1*).

	Run time = 50 Sec.			Run time = 200 Sec.		
	MAPM-NM	MAPM-TS	TS	MAPM-NM	MAPM-TS	TS
Min	436480	440946	449824	428502	428502	430416
Max	447647	461061	475252	434518	442487	461695
Average	442881	451009	462878	431614	435668	446816
Average gap to the best solution	1.5%	3.3%	6.0%	0.7%	1.7%	4.3%
Standard dev.	3199	5635	8417	1679	3990	9916

The standard deviation values indicate the better performance of MAPM-NM, and small standard deviation compared to the average value shows the robustness of the algorithm. The average gap to the best solution in MAPM-NM is also smaller and in 5 out of 30 replications (in 50 Sec results); the solution of MAPM-NM was very close to but not the best solution.

5.5 Evolution of the best solution and effect of the population management

Fig. 5 and 6 illustrates the evolution of the best solution in memetic algorithm with and without population management. Population diversity in MAPM smoothly reduces with the solution time, whereas in MA without population management, the heterogeneity is quickly lost (mutation probability is $p_m = 0.05$).

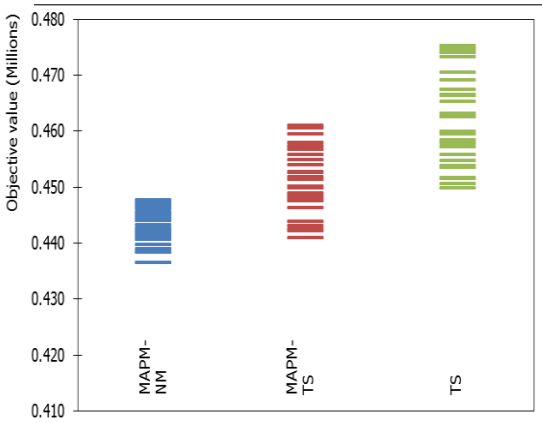


Fig. 3: Results of 30 replications in 100 seconds.

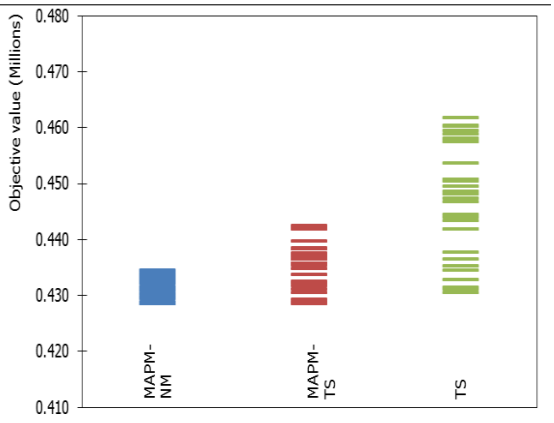


Fig. 4: Results of 30 replications in 360 seconds.

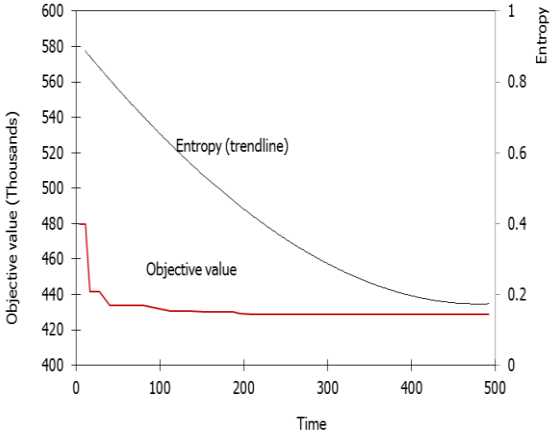


Fig. 5: Evolution of the best solution and entropy variations in MAPM.

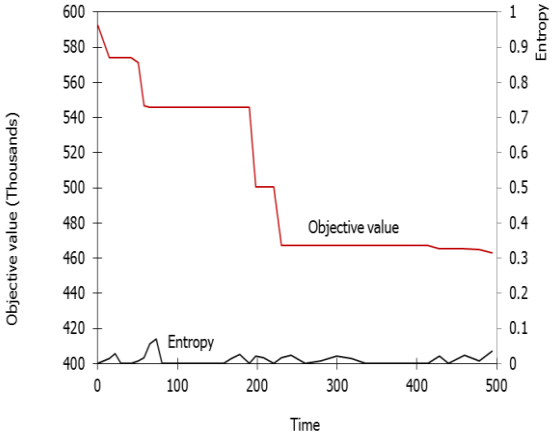


Fig. 6: Evolution of the best solution and entropy variations in MA.

Using intensification and diversification strategies and maintaining the population diversity in MAPM has resulted in obtaining better solutions in each instant of time. In 30 replications of the two algorithms (for 50 Sec.), in 78% of cases, MAPM has outperformed MA, and a statistical analysis (F-test) indicates a significant difference between the means of the two algorithms (F = 28.3 compared to $F_{CRITICAL}$ = 3.9).

6 Conclusions

We proposed a memetic algorithm with population management (MAPM) to figure out an integrated, production-maintenance scheduling problem with regard to quality aspects. This paper is the first one addressing a MAPM to solve joint problems in imperfect processes. The goal of population management is to balance between exploration and exploitation capabilities of population-based evolutionary algorithms. We adopted several strategies, including the diversity contribution of individuals in survival selection, inserting intensified or diversified solutions to maintain the heterogeneity, and adaptive control of diversity related parameters during the execution of the algorithm. Nelder-Mead algorithm is proposed for local search while, the algorithm uses Cplex to solve the linear part of the model. We also introduced an approach to exploit both positive and negative knowledge of the neighbor points. The results show that the MAPM-NM algorithm outperforms MA, TS and other configurations proposed in this paper. The proposed algorithm is efficient in terms of solution time and quality, and it can solve different instances of the problem.

References

[1] Y. Xiang (2013). Joint optimization of X control chart and preventive maintenance policies: A discrete-time Markov chain approach. European Journal of Operations Research, 229, 382–390.

[2] C.R. Cassady, R.O. Bowden, L. Liew & E.A. Pohl (2000). Combining preventive maintenance and statistical process control: A preliminary investigation. IIE Transactions, 32 (6), 471–478.

[3] L.A. Hadidi, U.M. Al-Turki & M.A. Rahim (2012). Joint job scheduling and preventive maintenance on a single machine. International Journal of Operations Research, 13 (2), 174–184.

[4] M. Nourelfath, N. Nahas & M. Ben-Daya (2016). Integrated preventive maintenance and production decisions for imperfect processes. Reliability Engineering & System Safety, 148, 21–31.

[5] J.P. Kenné, A. Gharbi & M. Beit (2007). Age-dependent production planning and maintenance strategies in unreliable manufacturing systems with lost sale. European Journal of Operations Research, 178, 408–420.

[6] Y.C. Chen (2013). An optimal production and inspection strategy with preventive maintenance error and rework. Journal of Manufacturing Systems, 32, 99–106.

[7] H. Beheshti Fakher, M. Nourelfath & M. Gendreau (2014). Profit maximization by integrating production planning, maintenance scheduling, quality aspects and sale decisions. Proceedings of the International Conference on Industrial Engineering and Operations Management, Bali, Indonesia, 1281–1292.

[8] H. Beheshti Fakher, M. Nourelfath & M. Gendreau (2016). A cost minimization model for joint production and maintenance planning under quality constraints, International Journal of Production Research (DOI:10.1080/00207543.2016.1201605).

[9] K.S. Moghaddam & J.S. Usher (2011). Sensitivity analysis and comparison of algorithms in preventive maintenance and replacement scheduling optimization models. Computers & Industrial Engineering, 61, 64–75.

[10] W.E. Hart, N. Krasnogor & J.E. Smith (2004), Recent advances in memetic algorithms. Heidelberg, New York: Springer.

[11] P. Merz (2000). Memetic algorithms for combinatorial optimization problems: fitness landscapes and effective search strategies. PhD dissertation, University of Siegen, Germany.

[12] P.K. Banerjee & M.A. Rahim (1988). Economic design of x-chart under Weibull shock models. Technometrics, 30, 407–414.

[13] P. Moscato & M. Norman (1989). A competitive and cooperative approach to complex combinatorial search. Technical Report 790, Caltech Concurrent Computation Program.

[14] F. Neri, & C. Cotta (2012). Memetic algorithms and memetic computing optimization: A literature review. Swarm and Evolutionary Computation, 2, 1–14.

[15] M. Laguna, R. Marti, & V. Campos (1999). Intensification and diversification with elite tabu-search solutions for the linear ordering problem. Computers and Operations Research, 26, 1217–1230.

[16] T. Vidal, T.G. Crainic, M. Gendreu, & C. Prins (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers & Operations Research, 40, 475–489.

[17] T. Nakagawa (1980). A summary of imperfect preventive maintenance policies with minimal repair. *Operations Research*, 14 (3), 249–255.

Nature-Inspired Algorithms: Design, Analysis and Application

Solving Flexible Job Shop Scheduling Problem Using A Global Updating Pheromone Rule In Knowledge Based Algorithm

H. Lehbab[1] and S.A. Rahal[2]

[1] University of Science and Technology of Oran,
Mohamed Boudiaf, Algeria
hayet.lehbab@univ-usto.dz
[2] University of Science and Technology of Oran,
Mohamed Boudiaf, Algeria
rahalsa2001@yahoo.fr

Abstract. Flexibility and complexity of production systems represent the main feature that cause difficulties to resolve scheduling problem. We have based our research on Ant Colony Optimization algorithm to solve flexible job-shop scheduling problem. Many approaches have been considered to improve this algorithm, whether to enhance the speed of finding the solution or the quality of this latter. In this paper, we present result of using another global updating pheromone strategy in a knowledge based ant colony optimization algorithm.

Mots-Clefs. Knowledge based algorithm, pheromone updating strategy, flexible job-shop scheduling problem, optimization , scheduling, ant colony optimization

1 Introduction

Flexible job shop scheduling problem - *FJSP* - is an extension of the classical job shop scheduling problem - *JSP* - where its aim is to be too close as possible to the real manufacturing systems. This extension is demonstrated in the fact that an operation can be performed by more than one machine. At this level, the size and the complexity of the problem are taking into account to choose which algorithm is the best to reach the best solution In case of considerable problem, heuristics and meta-heuristics are the more efficient means to reach optimal solutions. Among these methods, ant colony optimization is a class of metaheuristics, inspired from the behavior of real ant, know as very communicative with their environment [2] . We are interested in this work at the quality of the solution and the speed of the algorithm to lead it. we present at first the problem formulation and mention some work that have been done in this purpose. After that, we will present the knowledge based ant colony optimization and the integration of the other update pheromone strategy. finally we will discuss the result obtained and our future work.

2 Problem formulation

The FJSSP, Similar to the classical JSP, can be formulated as follows [2]:

- A set of independent job J= { J_1, J_2, ... , J_n}
- Each job is a sequence of preordered operation to be executed consecutively.
- A set of machine U= { M_1, M_2, ... , M_m}
- Each operation O_{ij} can be executed on any machine in a part of the set of machine U
- d_{ijk} denote the processing time of an operationO_{ij} on a machine M_k

We consider the assumptions listed as follows:

1. All the machines are available at the beginning.
2. All the jobs are available at time 0.
3. A machine can only process one operation at a time
4. An operation can be executed on a machine without interruption.
5. No more than one operation of the same job can be executed at a time.

Table 1. An instance of FJSSP

Job	*Operationnumber*	$M1$	$M2$	$M3$	$M4$	$M5$	$M6$
Job1	O11	5	-	4	-	-	-
	O12	-	1	5	-	3	-
	O13	-	-	4	-	-	2
Job2	O21	-	6	-	-	-	-
	O22	-	-	1	-	-	-
	O23	2	-	-	-	-	-
	O24	-	6	-	6	-	-
	O25	1	6	-	-	-	5
Job3	O31	-	6	-	-	-	-
	O32	-	-	4	-	-	2

Table 1 shows an instance of a flexible job shop scheduling problem. this instance consist of 3 jobs and machines. for example, the job J_1 is composed of three operations O_{11}, O_{12}, O_{13}; where the first operation O_{11} can be processed in machine M_1 or machine M_3 in 5 and 4 unit of time consecutively.

The main objective is to terminate all operations the earliest possible. This completion time is called makespan time. Hence, the schedule must balance between the precedence relations of operations in the same job and the dependence of machines. In other terms, the objective is the determination of optimal or near optimal schedules for the flexible job shop problem by assigning the operations to one of its alternative machines and sequencing the operations on the machines for the objective of minimization of makespan time, given the processing time of operations on all its alternative routes and the precedence relationship between the operations.

3 Related works

Several researchers have studied the flexible job-shop scheduling problem, we mention below some works that used the ACO to solve this problem.

The work [6] presented a modified scheme named local search ant colony optimization algorithm on the basis of alternative ant colony optimization algorithm for solving flow shop scheduling problems. This work integrates the local search mechanism into ant colony optimization algorithm for solving flow shop scheduling problem to improve the quality of solutions. Simulation results demonstrate that the appliedrandom order state transition rule used in ACO with local search integrated is an effective scheme for the flow shop scheduling problems.

[7] offered an overview of some bio-inspired approaches, especially ant colony. The authors analyze the efficiency of using these methods combined with multi-agents system to solve flexible job-shop problem.

In [8], a two-stage ant colony optimization (ACO) algorithm is implemented in a multi-agent system (MAS) to accomplish integrated process planning and scheduling (IPPS) in the job shop type flexible manufacturing environments. An innovative two-stage ACO algorithm is introduced where in the first stage of the algorithm, ants are directed to deposit pheromones at the nodes to select a set of more favorable processes. In the second stage, the set of nodes not selected in the first stage will be ignored, and pheromones will be deposited along the graph edges while the ants traverse the paths connecting the selected set of nodes.

[9] An adaptation of memetic algorithm combined with a new procedure of local search is proposed to solve job-shop scheduling problem considering reconfiguration. minimization of makespan and average tardiness time is considered to evaluate approach performance.

An improvement of the ant colony algorithm is presented in [5], where searchers combine the particle swarm method for scheduling problem of a flexible multi-objective job shop. The proposed algorithm is divided in two parts: the first uses of the PSO algorithm convergence speed to set the initial positions of the ants, and the second part uses the algorithm of ant colonies improved with local search algorithm, to find a global and optimal scheduling.

In [1], an application of the ant colony optimization is presented, an algorithm is introduced to optimize the computing time using a new pheromone update strategy that gave good results according to the authors with the application of the SPU rule to decrease the complexity of the algorithm.

The work presented in [3] proposed an algorithm based on ant colony that simultaneously performs scheduling, correspondence and linear placement of the spots. Their heuristic solutions built gradually and search the best around them, separating non-promising areas. They showed how to consider the partial dynamic reconfiguration for such problems and demonstrated that their approach was more general and robust compared to other conventional heuristics used to this kind of problem.

KBACO or Knowledge Based Ant Colony Optimization [4] is an integration of knowledge model with the optimization model of ant colonies ACO. The idea is to guide the ant colony optimization taking into account the knowledge gained from previous iterations of the ACO algorithm. In KBACO algorithm, the performance of the ant colony algorithm has been greatly improved by integrating the ACO model with the model of knowledge. Some knowledge available was learned from the optimization of the ACO by the knowledge model. At the same time, existing knowledge was applied to guide the current ACO heuristic search. The experimental results indicate that the proposed algorithm outperforms some other methods published in the quality scheduling.

Authors[10]used a neighbor structure in a local search combined with genetic algorithm which improve results compared to the result obtained of both algorithm separated.

According to this study, we notice that the idea of keeping a memory to guide the algorithm provide a lower makespan. We pay attention also to the speed of the KBACO algorithm, therefore we are presenting in this work an Improved Knowledge Based Ant Colony Optimization Algorithm described below.

4 Knowledge based algorithm

Inspired from KBACO algorithm proposed in [4], this algorithm which its framework is presented in the figure below is an instantiation of the Knowledge based Search Heuristic Architecture KB-SHA.
KBACO Algorithm can be summarized in the following steps[4]:

1. Initialization of a knowledge genotype which is composed of several set, the ESK: Elite Solution Knowledge, OAMK : Operation Assignment Machine knowledge, OAPK : Operation Assignment Priority knowledge.
2. Build a set of feasible solution using the ant colony optimization.
3. Update the knowledge genotype.

These steps are repeated until stopping criterion is reached.

In this work, we are interested to improve the quality of the solution obtained, which is the makespan and at the same time optimize the time of this algorithm to reach such a solution. We have implemented the same steps in the proposed approach [4], we are interested in the phase of updating the knowledge.

4.1 Pheromone updating Strategy

At this stage, the solution is constructed. Thus, the pheromone deposit rule is applied in the case where the makespan is improved, and the evaporation rule is applied for each iteration.
We consider following idea:

– Apply another update pheromone strategy named Global Updating rule based on the proposed approach in [1] where it has been proved that is the best update pheromone strategy between all of the strategies published.

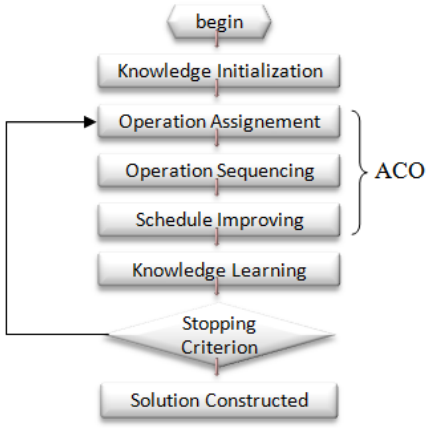


Fig. 1. Knowledge based optimization architecture

This rule is applied in our work on the Operation Assignment Machine Knowledge $OAMK$ and Operation Assignment Position Knowledge $OAPK$ defined in [4] :

$$OAMK(i,j,k) = OAMK(i,j,k) + (1-\rho) * \frac{q}{L} \tag{1}$$

$$OAPK(i,j,k) = OAPK(i,j,k) + (1-\rho) * \frac{q}{L} \tag{2}$$

where :
- p is evaporation rate
- q is a value in [0,1]
- L represents the length of the path which is in our case the makespan.

This rule is inversely proportional with respect to the path length; therefore, a larger value of pheromone is added to the most optimal makespan. In this strategy the rule is applied to a predetermined number of iterations, then for the next iteration, it is applied on another number of iterations. This process will be repeated until the stopping criterion.

5 Implementation and results

5.1 Data definition

The first set denoted BRdata, is composed of ten instances taken from Brandimarte (1993), which the number of jobs is between 10 and 20, the number of machine is between 4 and 15 and the total number of operation for each job is between 5 and 15.

5.2 Stopping criterion

After several tests of performance, we chose to stop the algorithm if the best results (the set of best solution) stabilized in a predefined number of iterations that we set at 100 stable iterations. Our goal is to compare the speed of the algorithm, and this criterion indicates the rate of change of the results.
If the algorithm stops quickly with a high makespan, this means that it is slow because it stabilizes quickly and so it needs a large number of iterations to achieve optimal makespan.

5.3 Evaluation criterion

We compared the obtained results with two criterion, the aim is to minimize the makespan, which is constitute the first criterion. In the other hand, we introduced the speed which is determined by the number of iterations divided by the execution time.
This latter constitute the second criterion since it gives the number of iterations per second.

5.4 Obtained Results

In Table 2, we present the best obtained results after 20 trial for each instance of problem where :

- N denotes the number of jobs
- m denotes the number of machines
- T flexibility rate (average number of operation per machine)
- Total is the total number of operation
- speed, the speed of the algorithm defined in the previous section

Table 2. Application of KBACO and IKBACO on Brandimarte dataset

File Name	Total	N x M	T	KBACO		IKBACO	
				Makespan	*Speed*	*Makespan*	*Speed*
Mk01	55	10x6	2	50	9,04	46	10,34
Mk02	58	10x6	3,5	47	5,83	44	5,85
Mk03	150	15x8	3	265	1,10	220	0,99
Mk04	90	15x8	2	90	3,56	78	3,65
Mk05	106	15x4	1,5	215	5,68	203	6,01
Mk06	150	10x15	3	105	0,10	93	0,38
Mk07	100	20x5	3	213	3,72	181	4,14
Mk08	225	20x10	1,5	615	0,63	554	0,65
Mk09	240	20x10	3	464	0,31	430	0,31
Mk10	240	20x15	3	362	0,16	350	0,15

The obtained results are illustrated on the following graph Fig3. with the flexibility of each problem.
we notice that the execution speed is clearly independent from the flexibility. For example, in the first and forth problem where the flexibility is equal, the speed remain different, since this result could be explained by the total number of operation which is 55 in the first problem and 90 in the fourth one.

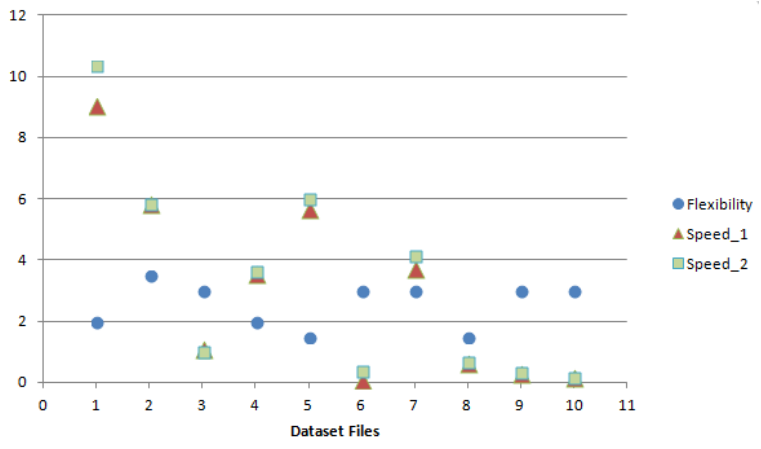


Fig. 2. Graphic showing the results of both approaches according to the flexibility

We applied both algorithms on the Hurink's dataset, which consist of problems with a larger number of jobs. The result are shown in Table 3 :

Table 3. Application of KBACO and IKBACO on Hurink's dataset

File Name	Total	N x M	T	KBACO		IKBACO	
				Makespan	*Speed*	*Makespan*	*Speed*
la26	200	20x10	1,15	1359	1,39	1365	1,37
la27	200	20x10	1,15	1397	1,37	1416	1,41
la28	200	20x10	1,15	1361	1,42	1380	1,39
la29	200	20x10	1,15	1349	1,28	1370	1,27
la30	200	20x10	1,15	1439	1,30	1389	1,33
la31	300	30x10	1,15	1794	0,77	1708	0,82
la32	300	30x10	1,15	1885	0,83	1928	0,80
la33	300	30x10	1,15	1765	0,76	1762	0,77
la34	300	30x10	1,15	1849	0,76	1799	0,77
la35	300	30x10	1,15	2017	0,71	1962	0,70

From Table 2 and Table 3,We notice that applying the new pheromone update rule gives:

- a better speed in 60% cases.
- a better Makespan in 75% cases.

5.5 Discussion

The speed of the algorithm after applying the update pheromone SPU strategy has slightly increased in 12 out of 20 problems which represent 60%. This results can be explained by the fact that in this Pheromone update strategy does not occur in all iterations thereby increasing the speed of the algorithm.
However, the results show the minimization of the makespan in 75 % cases in 20 trials.

6 Conclusion

We based this work on KBACO approach [4] which gave significant results in minimizing makespan according to comparisons made by the authors on their approach. In this paper the speed of the algorithm slightly increased by applying the global updating pheromone inspired by the approach presented in [1], giving a higher speed in 60% of problems tested and in the same way minimize makespan.

Finally, we suggest to develop other interesting aspects in the algorithm :
- Compare the speed of our approach with other meta-heuristics
- Assess the makespan obtained after the integration of our improvements.
- Varying the preset values in the algorithm, as the predetermined number of solutions in each iteration, as they may give different results.

References

1. Anitha, J. and M. Karpagam (2013). Ant Colony Optimization using Pheromone Updating Strategy to Solve Job Shop Scheduling, Proceedings of 71th International Conference on Intelligent Systems and Control, 367-372.
2. Dorigo, M. and T. Sttzle, (2003). Handbook of Metaheuristics, volume 57 of International series in operations research and management science, chapter The Ant Colony Optimization Metaheuristics : Algorithms, Applications and Advances. Kluwer Academic Publishers, Boston Hardbound.
3. Ferrandi, F., P.L. Lanzi, C. Pilato and D. Sciuto, Ant Colony Optimization for Mapping, Scheduling And Placing in Reconfigurable Systems NASA/ESA Conference on Adaptive Hardware and Systems, (AHS-2013) 47-54
4. Li-Ning, X., C. Ying-Wu,W. Peng, Z. Qing-Song, X. Jian, (2010). a Knowledge-Based Ant Colony Optimization for Job Shop Scheduling Problem, Applied soft computing, p. 888- 896, 2010.
5. Li, L., W. Keqi and Z. Chunnan An Improved Ant Colony Algorithm Combined with Particle Swarm Optimization Algorithm for Multi-objective Flexible Job Shop Scheduling Problem , IEEE Computer Society: International Conference on Machine Vision and Human-machine Interface, (2010) 88-91.
6. Chen, Ruey-Maw, et al. "An effective ant colony optimization-based algorithm for flow shop scheduling." Soft Computing in Industrial Applications, 2008. SMCia'08. IEEE Conference on. IEEE, 2008.
7. Leito, Paulo, Jos Barbosa, and Damien Trentesaux. "Bio-inspired multi-agent systems for reconfigurable manufacturing systems." Engineering Applications of Artificial Intelligence 25.5 (2012): 934-944.
8. Wong, T. N., et al. "Integrated process planning and schedulingmulti-agent system with two-stage ant colony optimisation algorithm." International Journal of Production Research 50.21 (2012): 6188-6201.
9. Zhang, Zhiqiang, and Zhang Jing. "An Improved Ant Colony Optimization Algorithm for Permutation Flowshop Scheduling to Minimize Makespan." 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies. IEEE, 2012.
10. Gonzlez Fernndez, Miguel ngel, Mara del Camino Rodrguez Vela, and Jos Ramiro Varela Arias. "An efficient memetic algorithm for the flexible job shop with setup times." Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling. Association for the Advancement of Artificial Intelligence (AAAI), 2013.

Alignment of Multiple Sequences by the adaptation of the fireworks algorithm

K. Khelifi[1] ,L. Djerou[2], and N.khelil[3]

[1]. *University of Biskra, Algeria*
khaoula.kh21@gmail.com

[2]. *LESIA Laboratory, University of Biskra, Algeria*
ldjerou@yahoo.fr

[3]. *AM Laboratory, University of Biskra, Algeria,*
khelilna@yahoo.fr

Keywords: Fireworks algorithm, Multiple sequence alignment, Neighborhood generation, Intensification/Diversification strategies.

1 Introduction

Multiple sequence alignment (MSA) is an important tool in biological analysis and one of the most challenging and active ongoing research problem fields of computational molecular biology. Alignment of sequences can be an important tool to measure the similarity of two or more sequences of DNA, RNA, or Protein. It can be used to: determine evolutionary distances between organisms and infer phylogenetic relationships, discover conserved motifs that might be important at the levels of transcription, translation, and/or structure, and improve our understanding and prediction of molecular structures.

The problem of sequence alignment is to obtain the maximum alignment of a set of n genomic sequences, which is denoted as $S = \{s_1, s_2, \ldots, s_n\}$; each sequence of this set is formed by the alphabet $\sum = \{A,C,G,T\}$. The solution of this problem is represented by $S' = \text{Max}(S)$, which denotes a set $S' = \{s'_1, s'_2, \ldots, s'_n\}$ with the alphabet $\sum' = \sum \cup \{-\}$, S' represents the optimal alignment of S.

Finding the optimal alignment of a set of sequences is known as a NP-complete problem [3], due to its exponential time complexity when the number of sequences and their lengths increase.

Several iterative methods were proposed in the literature to solve MSA problem. The basic idea is to start by an initial alignment and iteratively refines it through a series of suitable refinements called iterations. The process is reiterated until satisfaction of some criteria. Iterative methods can be deterministic or stochastic, depending on the strategy used to improve the alignment. In this way, several metaheuristics have been designed to obtain suboptimal alignments, for example, Simulated Annealing, Tabu Search, Ant Colony Algorithm, Genetic Algorithms, among others. The disadvantage is that metaheuristics do not guarantee optimal solutions, but solutions generated can be very close to optimal solution in a reasonable processing time.

In this work, we use a novel swarm intelligence optimization metaheuristic “the fireworks algorithm (FWA)”, developed by Ying Tan in 2010 [4], to obtain optimal results of multiple sequence alignment. The fireworks algorithm (FWA) is inspired by the fireworks explosion in the night sky, and it is based on four operators: the explosion operator, the mutation operator, the mapping rule, and selection strategy. The effect of the explosion operator is to generate sparks around fireworks. The number and amplitude of the sparks are governed by the explosion operator. After that, some sparks are produced by mutation operator. The mutation operator utilizes Gaussian operator to produce sparks in Gaussian distribution. Under the effect of the two operators, if the produced spark is not in the feasible region, the mapping rule will map the new generated sparks into the feasible region. To select the sparks for next generation, the selection strategy is used at last. Fireworks algorithm runs iteratively until reaches the termination conditions.

Since its advent in 2010, FWA has attracted researcher's attention from diverse domains of knowledge. Much effort has been devoted to analyze the conventional FWA and many applications are reported, where FWA is successfully utilized to resolve diverse problems [5].

Our proposed FWS Alignment algorithm is the adaptation of FWA algorithm to MAS problem resolution that can be summarized as following:

In initialization, FWS Alignment algorithm generates N fireworks (solutions = alignments) as follows: generate an initial alignment using ClustalW algorithm which is based on the progressive method of Feng and Doolittle [1]. Then the rest of the population is generated by using a set of move mechanisms in the positions of gaps and residues in the initial alignment.

The COFFEE (Consistency-based Objective Function For alignmEnt Evaluation) function [2] is used as an objective function. It works by first generating the pairwise library of the sequences in the alignment and then it calculates the level of identity between the current multiple alignment and the pairwise library. The global score measuring the quality of the alignment A is computed by the following formula.

$$f(A) = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} W_{ij} * Score(A_{ij})}{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} W_{ij} * Len} \qquad (1)$$

where N is the number of sequences; Len is the length of the multiple alignment; W_{ij} is the percent identity between the two aligned sequences S_i and S_j; A_{ij} is the pairwise projection of sequences S_i and S_j obtained from the multiple alignment; and $Score\ (A_{ij})$ is the overall level of identity between A_{ij} and the corresponding pairwise alignment. When COFFEE is used as objective function, an optimal MSA solution is one which achieves the maximum COFFEE score.

Each alignment A_i, of this initial population, generates sparks (alignments) by explosion operation that include the explosion strength, explosion amplitude and displacement operation. The explosion strength calculates the number of spark alignments Ns_i and the amplitude Am_i of the current alignment A_i using the following equations:

$$Ns_i = m * \frac{f(A_i) - y_{min} + \varepsilon}{\sum_{j=1}^{N}\left(f(A_j) - y_{min}\right) + \varepsilon} \qquad (2)$$

the limitation of number of sparks $\widehat{Ns_i}$ is as follws:

$$\widehat{Ns_i} = \begin{cases} round(a.m), & if\ Ns_i < am, \\ round(b.m), & if\ Ns_i > bm \quad , a < b < 1 \\ round(a.m), & otherwise \end{cases} \qquad (3)$$

The amplitude of the fireworks with better fitness values gradually reducing, On the contrary, the fireworks with worse fitness values will explore the optima through large amplitude. The explosion amplitude is defined below:

$$Am_i = SA * \frac{y_{max} - f(A_i) + \varepsilon}{\sum_{j=1}^{N}\left(y_{max} - f(A_j)\right) + \varepsilon} \qquad (4)$$

Where a and b are constant, round () is the rounding function, SA is a constant as the sum of all amplitudes, while Y_{min} means the fitness value of the worst individual among the N individuals, Y_{max} is the fitness value of the best individual among the N individuals, while the parameter ε is used to prevent the denominator from becoming zero.

FWA generates different random displacements within each amplitude to ensure the diversity of population. The amplitude is determined by the maximum possible distance between the current alignment and the spark alignments which will be generated. Displacement operation, in FWS Alignment algorithm, is used to identify where the change in positions of gaps and residues will be applied.

To further improve the diversity of a population and increase the fitness of alignments, FWS Alignment algorithm applies a mutation on the alignments which have a fitness value between the fitness of the best alignment and another alignment chosen randomly. This mutation replaces two random sequences in the multiple sequence alignment by the corresponding sequences in their pairwise alignment.

To ensure that all spark alignments are placed inside the amplitude of their firework alignment source, the hamming distance is used, in our approach, to calculate the distance between every firework alignment and its spark alignments, if a distance was found superior of the amplitude the spark alignment will regenerated until its distance will be lower than the amplitude.

To generate the next population of fireworks, first of all, the selected strategy, in our method, keeps the best alignment in the population; the rest is chosen using the hamming distance where the alignment that is farther from other alignments has more chance to be selected. The last element necessary for FWS Alignment algorithm FWA is termination criteria. In our study, the search can be stopped when certain number of iterations is completed.

To evaluate our approach, we have used a sequence data sets provided by BAliIBASE benchmark. BAliBASE offers a baliscore program in C, which uses two different scoring systems to estimate the quality of an alignment compared to a reference alignment: SPS (Sum-of-Pairs Score) and CS (Column score). The following table presents the comparison of our method with the four well-known MSA methods: clustalx, mafft, muscle and clustalO. The results encourage us to continue in improving this research.

	clustalx		mafft		muscle		clustalO		**FW Alignment**	
	SPS	CS	SPS	CS	SPS	CS	SPS	CS	SPS	CS
1csy_ref1	0.861	0.742	0.846	0.730	0.825	0.719	0.836	0.764	**0.863**	**0.742**
1ycc_ref1	0.907	0.821	0.899	0.821	0.857	0.738	0.792	0.583	**0.907**	**0.821**
3grs_ref2	0.800	0.079	0.820	0.213	0.843	0.241	0.825	0.167	**0.800**	**0.079**
1idy_ref3	0.624	0.250	0.633	0.313	0.757	0.188	0.392	0.000	**0.587**	**0.000**
1pysA_ref4	0.558	0.177	0.633	0.287	0.670	0.354	0.730	0.478	**0.501**	**0.177**

Table.1. comparison using computed BAliIBASE scores

References

[1] D. Feng and R. Doolittle (1987). Progressive Sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of molecular Evolution 25*, 351-360.

[2] C. Notredame, L. Holm, and D. G. Higgins (1998). COFFEE: an Objective Function for Multiple Sequence Alignments. *Bioinformatics,* Vol. 14, pp. 407-422.

[3] L.Wang and T. Jiang (1994). *The* complexity of multiple sequence alignment. *Journal of Computational Biology*, vol. 1, no. 4, pp. 337–348.

[4] Y.Tan and Y. Zhu (2010). Fireworks algorithm for optimization. *Advances in Swarm Intelligence. Springer*, pp. 355–364.

[5] Y.Tan (2015). Algorithm; a Novel Swarm Intelligence Optimization Method. *ISBN: 978-3-662-46352-9* (Print) *978-3-662-46353-6* (Online).

Work in Progress on Autoconstructive Evolution (Extended Abstract)

Lee Spector

Cognitive Science
Hampshire College
Amherst, MA 01002 USA
lspector@hampshire.edu

Biological evolution is itself subject to evolutionary change. The mechanisms by which organisms survive, select mates, and produce variation, themselves vary and are subject to selection over evolutionary time. This ability of evolution to itself evolve is arguably crucial for its adaptive power; while the evolutionary mechanisms of early Earth were not capable of producing human intelligence and culture without further development, they eventually evolved into mechanisms that could do so, billions of years later.

The fact that evolution evolves has long been recognized within the field of evolutionary computation, and researchers have expressed eagerness to incorporate self-adaptive elements into evolutionary computation systems. One motivation for this interest is the fact that there is significant evidence, both theoretical and empirical, that the human programmers and users of evolutionary computation systems make a variety of choices that matter, and that these choices could probably be improved. For example, the details of the design and parameters of the genetic operators that produce variation can have profound effects [11], and several promising recent developments involve new genetic operators [12]. We have no reason to believe that we have yet found the best possible genetic operators or that the same operators will be best across different problems or even across different runs on the same problem.

For these reasons, several "self-adaptive" evolutionary computation approaches have been proposed. However, in most of the prior work, the ways in which the evolutionary process could change over time were quite limited, often to the changing of mutation rates [1] or the parameters or configuration of hand-written genetic operators [2, 4]. In work on "meta-genetic programming" [3], populations of genetic operators co-evolve with populations of problem solving systems, but neither this approach nor any of the other prior work in this area has produced systems that are convincingly more successful or adaptive than their non-self-adaptive counterparts on problems involving the synthesis of programs or systems for complex environments.

The prior work that my own research group has conducted in this area, under the description of "autoconstructive evolution," might be considered among the more ambitious projects in this tradition, even if it has produced no greater successes than others until, perhaps, recently. In autoconstructive evolution the algorithms for the production of offspring, and hence for significant aspects of the overall evolutionary process, are encoded within the individuals in evolving populations, as they are in biological systems. They are thereby subject to variation, selection, and evolution.

In a sequence of papers over approximately the last decade and a half, we have explored fundamental issues related to autoconstructive evolution [13–15, 5, 6]. The systems demonstrated in these papers could only solve relatively simple problems, but they demonstrated and allowed for exploration of the evolutionary dynamics that result when methods of variation can themselves vary, as they can and do in biology. These systems generally solved problems no better than standard genetic programming systems, but this was not surprising because they had to evolve their own mechanisms of variation even as they evolved solutions to problems. Standard genetic programming systems, on the other hand, use "hand-designed" genetic operators that work reasonably well even though they cannot adapt. This prior work allowed us to study principles of autoconstructive evolution, but it did not produce particularly useful problem-solving systems.

Two recent innovations, however, improve the prospects for autoconstructive evolution as a method that may outperform existing evolutionary algorithms. The first is a new parent selection algorithm, called "lexicase selection," that promotes and maintains diversity in ways that significantly enhance the problem-solving power of genetic programming in many domains [16, 7, 8]. In an autoconstructive evolution system, lexicase selection has the additional benefit of preventing

individuals with maladaptive reproduction strategies from dominating a population. The second recent innovation is the development of "diversification" constraints on reproduction in autoconstructive evolution, which ensure that individuals are capable of producing offspring that differ in their reproductive methods. These two innovations, in combination, have allowed autoconstructive evolution to solve significantly more difficult problems than it has solved in the past.

The aim of our current work on this project is to systematically investigate and exploit the power of autoconstructive evolution to produce intelligent and robust systems that solve difficult problems. Our current system, called AutoDoG (for "Autoconstructive Diversification of Genomes"), uses linear genomes for hierarchically structured programs [9], lexicase selection, and diversification constraints on individuals in the population. We have observed that AutoDoG is capable of finding general solutions to the "Replace Space with Newline" software synthesis problem [10], we have analyzed the evolutionary dynamics of AutoDoG relative to non-autoconstructive runs of our system, and we have presented data showing that the processes of variation do indeed vary and evolve over the course of AutoDoG runs [17].

We have not yet determined which features of AutoDoG are responsible for its new-found problem-solving power. In the next phase of our research we will investigate this question, in part by systematically disabling elements of the system, to see which are crucial for solving problems. We then plan to focus on testing and attempting to enhance the effectiveness of our linear genome representations, lexicase selection, and diversification constraints. We will also investigate the nature of the reproductive methods that evolve in successful runs of autoconstructive evolution. We expect to have preliminary results on these experiments to share, along with an overview of the overall project of autoconstructive evolution, at the META'2016 conference.

Applications in this phase of the project will focus on a set of general program synthesis benchmarks recently developed in our group, which generally exceed that capabilities of existing genetic programming systems (and of program synthesis systems of all other types as well) [10]. In this context, we hope eventually to produce autoconstructive evolution systems with problem-solving power that rivals or exceeds that of genetic programming systems with hand-designed genetic operators. We also plan to apply autoconstructive evolution to a variety of application areas that are particularly challenging with respect to intelligence and robustness, including the evolution of control strategies for agents in complex virtual worlds.

Autoconstructive evolution should be of interest to the broader community of metaheuristics researchers both because it is itself a metaheuristic method with significant potential, and because it provides a case study regarding several issues that can arise in self-adaptive evolutionary algorithms. This presentation will review prior work on autoconstructive evolution, describe the current state-of-the-art in detail, present the result of recent experimental work, and outline topics for continued research.

References

1. Peter J. Angeline. Adaptive and self-adaptive evolutionary computations. In *Computational Intelligence: A Dynamic Systems Perspective*, pages 152–163. IEEE Press, 1995.
2. Peter J. Angeline. Two self-adaptive crossover operators for genetic programming. In Peter J. Angeline and K. E. Kinnear, Jr., editors, *Advances in Genetic Programming 2*, chapter 5, pages 89–110. MIT Press, Cambridge, MA, USA, 1996.
3. Bruce Edmonds. Meta-genetic programming: Co-evolving the operators of variation. *Elektrik*, 9(1):13–29, May 2001. Turkish Journal Electrical Engineering and Computer Sciences.
4. Brian W. Goldman and Daniel R. Tauritz. Self-configuring crossover. In Gisele L. Pappa, Alex A. Freitas, Jerry Swan, and John Woodward, editors, *GECCO 2011 1st workshop on evolutionary computation for designing generic algorithms*, pages 575–582, Dublin, Ireland, 12-16 July 2011. ACM.
5. K. Harrington, E. Tosch, L. Spector, and J. Pollack. Compositional autoconstructive dynamics. In *Unifying Themes in Complex Systems Volume VIII: Proceedings of the Eighth International Conference on Complex Systems*, New England Complex Systems Institute Series on Complexity, pages 856–870. NECSI Knowledge Press, 2011.
6. Kyle I. Harrington, Lee Spector, Jordan B. Pollack, and Una-May O'Reilly. Autoconstructive evolution for structural problems. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, GECCO Companion '12, pages 75–82, New York, NY, USA, 2012. ACM.

7. T. Helmuth, L. Spector, and J. Matheson. Solving uncompromising problems with lexicase selection. *Evolutionary Computation, IEEE Transactions on*, 19(5):630–643, Oct 2015.
8. T. M. Helmuth. *General Program Synthesis from Examples Using Genetic Programming with Parent Selection Based on Random Lexicographic Orderings of Test Cases*. PhD thesis, College of Information and Computer Sciences, University of Massachusetts, Amherst, 2015.
9. Thomas Helmuth, Nicholas Freitag McPhee, and Lee Spector. Plush: Linear genomes for structured push programs. In *Genetic Programming Theory and Practice XIV*, Genetic and Evolutionary Computation. Springer, 2016.
10. Thomas Helmuth and Lee Spector. General program synthesis benchmark suite. In Sara Silva, Anna I Esparcia-Alcazar, Manuel Lopez-Ibanez, Sanaz Mostaghim, Jon Timmis, Christine Zarges, Luis Correia, Terence Soule, Mario Giacobini, Ryan Urbanowicz, Youhei Akimoto, Tobias Glasmachers, Francisco Fernandez de Vega, Amy Hoover, Pedro Larranaga, Marta Soto, Carlos Cotta, Francisco B. Pereira, Julia Handl, Jan Koutnik, Antonio Gaspar-Cunha, Heike Trautmann, Jean-Baptiste Mouret, Sebastian Risi, Ernesto Costa, Oliver Schuetze, Krzysztof Krawiec, Alberto Moraglio, Julian F. Miller, Pawel Widera, Stefano Cagnoni, JJ Merelo, Emma Hart, Leonardo Trujillo, Marouane Kessentini, Gabriela Ochoa, Francisco Chicano, and Carola Doerr, editors, *GECCO '15: Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, pages 1039–1046, Madrid, Spain, 11-15 July 2015. ACM.
11. S. Luke and L. Spector. A revised comparison of crossover and mutation in genetic programming. In *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 208–213, University of Wisconsin, Madison, Wisconsin, USA, 22-25 July 1998. Morgan Kaufmann.
12. Tomasz P. Pawlak, Bartosz Wieloch, and Krzysztof Krawiec. Review and comparative analysis of geometric semantic crossovers. *Genetic Programming and Evolvable Machines*, 16(3):351–386, September 2015.
13. L. Spector. Autoconstructive evolution: Push, pushGP, and pushpop. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 137–146, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.
14. L. Spector and A. Robinson. Genetic programming and autoconstructive evolution with the push programming language. *Genetic Programming and Evolvable Machines*, 3(1):7–40, March 2002.
15. Lee Spector. Towards practical autoconstructive evolution: Self-evolution of problem-solving genetic programming systems. In Rick Riolo, Trent McConaghy, and Ekaterina Vladislavleva, editors, *Genetic Programming Theory and Practice VIII*, volume 8 of *Genetic and Evolutionary Computation*, pages 17–33. Springer New York, 2011.
16. Lee Spector. Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, GECCO Companion '12, pages 401–408, New York, NY, USA, 2012. ACM.
17. Lee Spector, Nicholas Freitag McPhee, Thomas Helmuth, Maggie M. Casale, and Julian Oks. Evolution evolves with autoconstruction. In *Companion Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, page in press. ACM, 2016.

mapGA: A Genetic Algorithm for Mapping Large Ontologies

Guefrouchi Ryma
MISC Laboratory, Abdelhamid Mehri University Constantine
Algeria
Email: ryma_doc@yahoo.fr

Pr. Kholladi Mohamed-Khireddine
El Oued University
Algeria
Email: kholladi@yahoo.fr

Abstract. Ontology mapping is one solution to the ongoing problem of ontology heterogeneity. Finding such a solution can be seen as an assignment problem of concepts between the source and target ontologies, using some similarity metrics. To overcome the complexity of the task of computing similarities between concepts, this paper adapts the genetic algorithm mapGA for ontology mapping. A concept-correspondence matrix is searched using the genetic algorithm. The latter provides a binary matrix of correspondences between concepts, and three rules are proposed to help in reducing the number of needed-similarities calculations between concepts.

1 Introduction

Nowadays, a great interest is given to the ontologies because of their power in the representation, treatment and knowledge management. Several ontologies in different areas have been developed according to different standards, in several languages and by several organizations. In order to utilize, exploit and take advantage of the knowledge stored in these heterogeneous ontologies and operate on them in related domains, it is necessary to proceed by performing a set of operations including comparisons as well as mapping the composing concepts. The most cited definition of the mapping is that of Noy, in which it is considered as a process that specifies a semantic convergence between different ontologies in order to extract the correspondences between certain entities [1], [2], this process includes three phases: the discovery, the representation and reasoning with mappings [2].
This paper is interested in the phase of the discovery of ontology mapping which is subject to the problem of terminological heterogeneity that occurs due to variations in names when referring to the same concepts in different ontologies [3]. This terminological heterogeneity is solved through the measurement of the terminological similarity between the concepts of ontologies and many methods have been proposed in this context. These methods are characterized by the comparison between all the

couples of concepts belonging to each of the ontologies to map. The main drawback among all these methods occurs while dealing with large-sized ontologies where they lose their effectiveness due to the significant time and memory consuming.
In this paper, the proposed is the use of mapGA to decrease the processing time as well as to obtain the optimal mapping, starting from a set of random mappings and taking advantage of the genetic operators.
The reminder of this paper is organized as follows. Section 2 provides a review of the literature related works in the field of mapping large-sized ontologies. Section 3 describes and details our proposed approach for the in-hand problematic.
Section 4 summarizes the obtained results and shows the performance of the proposed algorithm. Finally, a conclusion and research perspectives are drawn up in Section 5.

2 Contribution

Genetic algorithms are population-based computational models inspired by genetics and evolution [9]. The paper in hand presents the use of mapGA a genetic algorithm for the mapping of large ontologies. Finding the optimal mapping between ontologies is considered as assignment problem. In which the concepts of the two ontologies being mapped are matched using a correspondence matrix. mapGA is used for ontology mapping to accelerate the process of computing similarities specially when it comes large ontologies. Fig.1 illustrates the steps of our algorithm:

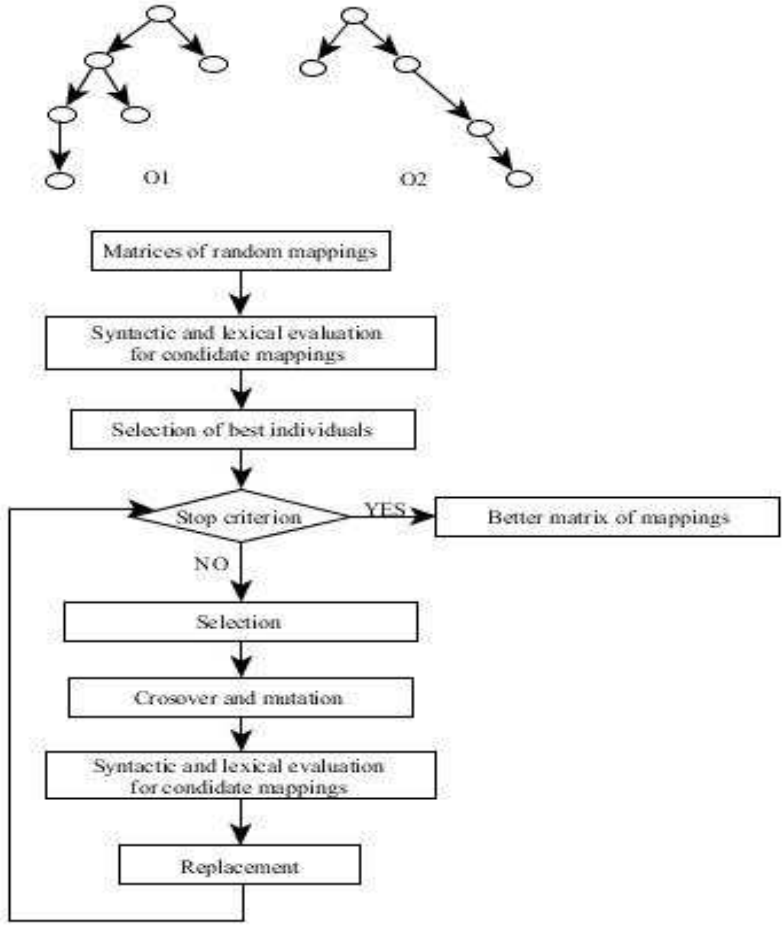


Fig. 1. The steps of the proposed genetic algorithm.

2.1 Problem formulation

To treat the ontology mapping problem as an optimisation problem two elements are to be defined: representation of the solution and the objective function. The search process is also presented.

2.2 Representation of the solution

For each concept in an ontology it mapped zero or one concept in the second ontology. For these raison, a solution represented as a binary matrix in which the lines are the concepts of the first ontology and columns are the concepts of the second, a 1

represents the existence of correspondence between concepts, and a 0 represents its absence; with only one 1 in each line and column because only 1 to 1 or 1 to 0 mappings are considered.

2.3 Objective function

The used objective function, to be maximised, is the number of correct mappings between the concepts in a solution matrix. More precisely, a 1 in a matrix is considered to be a correct mapping if either the two involved concepts are syntactically or lexically similar. A lexical similarity exists when the two concepts have at least one common synonym, while a syntactic similarity requires a 90 % overlapping between them. We use Levenstein distance generally used for the syntactic mapping, though for the OAEI[1]. [18].Some minor pre-processing is done on the concepts labels for capital letters, camels, and underscores. Lexical measure S_LEX (C1, C2) between two concepts C1 and C2 is given by the formula (1):

$$S\_LEX(C1, C2) = \text{Cardinality}\ (\text{syno}(E1) \cap \text{syno}(E2)). \quad \textbf{(1)}$$

Where Syno (C1) and syno (C2) are the vectors of synonyms of both concepts C1 and C2. We use the famous WordNet dictionary for enrichment of the two vectors.
In Order to reduce the calculations number, three rules are proposed in each iteration:

- Similarity measures are computed only for candidate mappings in the binary matrices.
- For Correct discovered mappings, the columns and lines will be eliminated in the next iterations for all the individuals in the population.
- For Incorrect discovered mappings, only these mappings will be eliminated in the next iterations for all the individuals in the population.

2.4 The search process

As seen in chart of figure, the flowing steps are to be applied:

- Selection: Better individuals are selected for the mutation and cross over operators.
- Mutation operator: it consists on a permutation between two columns for discovering new individuals. Fig.3 illustrates this operator.
- Cross Over operator: it is a Single Point crossover, it is a combination between two old individuals for given new once. The first part of the individual is that of its parent and the second is constitute in putting the one in the order of appearance in the second parent. Fig.4 illustrates this operator.

[1]OAEI (Ontology Alignment Evaluation Initiative) organizes evaluation campaigns aiming at evaluating ontology matching technologies. http://oaei.ontologymatching.org/.

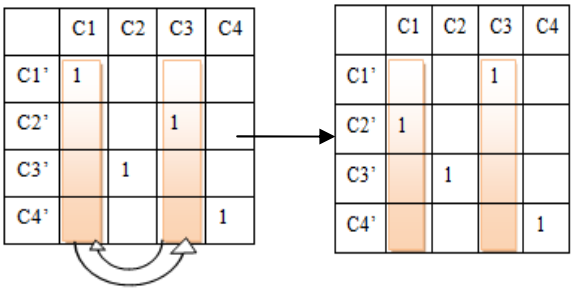
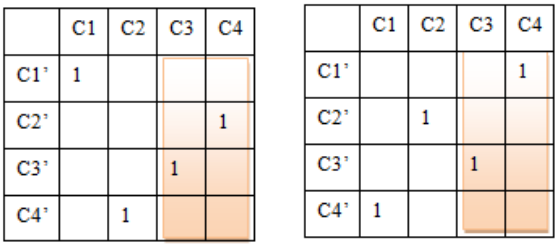


Fig.2. Mutation operator

Parents



Children

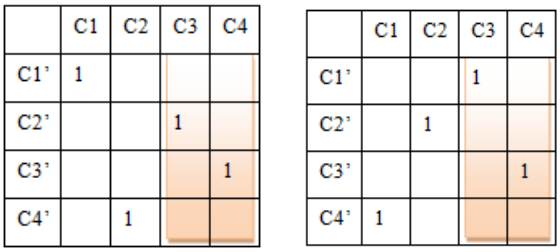


Fig.3. Cross Over operator

- Evaluation and replacement: Fitness functions are updated, and the replacement is elitist, where better individuals replace bad once.
- Stop criterion: mapGA is stopped after a number of generations, or when fitness function closes to the optimal.

3 Related works

This section presents a brief review of the related works based on optimization technique for ontology mapping. We start with GA which is used in two different contexts: optimising the quality of similarity measures, or optimising directly the global ontologies mapping. For the first context GAOL [13], is a genetic based

algorithm for determining the optimal weight configuration for a weighted average aggregation of various bases matchers. The algorithm uses a one dimension array of matcher weights for presenting an individual, and its fitness function is one of the famous parameters recall, precision, and F-measure, the main inconvenient of GAOL is that it needs a reference alignment in order to evaluate its fitness function.

Near to our mapGA, and in the context of optimizing the global ontologies mapping, GAOM is introduced, the two algorithms are similar in their genetic operators, and fitness functions those not require any reference alignments. In addition to this, for the two algorithms, the stop criterion is a number of generations, or fitness function close to the optimal. But distinguished differences between the two algorithms are summarized in Table.1:

Table.1. Comparison between GAOM and mapGA

	GAOM	mapGA
Individual presentation	One-dimensional array V with n1 integer elements that may take values between 1 and n2: V [n1]= n2 , mean that concept number n1 is mapped to number n2.	Matrix M [n1, n2]
	Only 1:1 correspondences are considered.	1:1 and 1:0 correspondences are considered.
	The size of the array is fixed through the iterations and the algorithm cannot trait the cases where the not mapped concepts.	The size the matrix is reduced through the iterations and the cases of not mapped concepts are discovered.
Function fitness	The use of extensional and intentional features of ontological concepts to evaluate the mappings.	The use of similarity measures(syntactic, lexical, …etc) of ontological concepts
	All the mappings are computed for evaluating any global mapping.	Only candidate mappings are computed for evaluating any global mapping in order to reducing the number of needed calculations.

As seen in the table 1. GAOM is limited when there is only a partial overlap of ontologies, and when it consists of large ontologies. For our proposal, the individuals are represented in a binary matrix for booking the correct mappings throughout the iterations, both size of mapping matrices and number of needed calculations are reduced through the iterations to ensure the convergence towards the best global mapping.

Other optimization algorithm for ontology mapping are proposed, including maPSO and MapEVO [15], they are two systems based respectively on DPSO (Discrete Particle Swarm optimization), and Evolutionary programming. For the first algorithm an individual is a swarm particle, whereas for the second an individual is an evolving specie. The function fitness for the two algorithms is the same fitness values based on local and global fitness of each single correspondence in the alignment. The main difference between these algorithms and mapGA is the number of needed

calculation to evaluate a global mapping where in the case of mapPSO and mapEVO all correspondences must be computed in each iteration.

4 Experiments and results

To show the effectiveness of mapGA, it was tested on a PC with the following characteristics: Intel Core 2 Duo 2.33GHz and 4GB RAM. It was also several choices for the parameters of GA, and benchmarks of ontologies to be mapped.

4.1 mapGA Parameters:

- A population of 10 individuals
- 0.90 for crossover probability
- 0.08 for mutation probability
- Number of generations less than the size of the mapped ontologies.

4.2 Benchmarks

For the evaluation of our algorithm, like most of works in ontology mapping, we used both measurement precision and Recall [16]. The two measurements are given by the following formulas:

$$\text{Precision} = \frac{\text{relevant mapping} \cap \text{retrieved mapping}}{\text{relevant mapping}}$$

$$\text{Recall} = \frac{\text{relevant mapping} \cap \text{retrieved mapping}}{\text{releretrieved mapping}}$$

In our evaluation, we choose pairs of ontologies of OAEI 2011 benchmarks, The reference ontology is that of test #101which is matched with variation of ontologies grouped according to their levels of complexity [17], table 2 shows the obtain results of mapEVO, mapPDO, and mapGA :

	mapEVO		mapPSO		mapGA	
	Precision	Recall	Precision	Recall	Precision	Recall
1xx	0.99	1.00	0.99	0.92	1	1
2xx	0.54	0.21	0.63	0.62	0.50	0.47

Table2. Recall and precision for mapEVO, mapPSO, and mapGA.

For comparing time execution, we have used the same reference ontology matched with three other ontologies:

- Biblio: it concerns bibliographic references and it contains 33 named classes.
- Ekaw: from conference track, it contains 74 classes and 33 object properties.
- Finance: based on the Finance ontology, which contains 322 classes.

	Bilbio	Ekaw	Finance
mapGA	4 mn	5 mn	45 mn
mapPSO	3.09h	3.72h	85.98h
mapEVO	7.44	9.96	1.25h

Table3. Time comparaison between mapGA, GAOM, mao PSO and mapEVO.

As show in table2, in simple tests (1xx) the three systems have similar-high performance, because these tests consist on comparing the ontology with itself or with another irrelevant ontology, for second groupe of tests (2xx), the two systems mapPSO an mapEVO are slightly better than our algorithm, simply because they introduce structural measures in their evaluation of a correspondence. Looking Table.3, is clear that our algorithm is most better comparing the time execution, so it is most performer when it consist of large ontologies, this is because we are focus our work on reducing number of needed calculations for discovering correspondences.

5 Conclusion and Perspectives

This paper addresses the problem of high similarities computing time between concepts of ontologies within the discovery phase of mapping. In this context, this paper uses mapGA for discovering the mappings between concepts of ontologies before executing all the similarity computations. The proposed algorithm uses a matrix representation, syntactic with lexical metrics for computing function fitness of a solution, and propose three rules for reducing the number of needed calculations. The obtain results show the effectiveness of the proposal for mapping large ontologies. As a future work, it aims to introduce in the fitness function structural measures for including the relationships between concepts in the evaluation of a global mapping. It aims also to participate at the OAEI platforms evaluation.

Acknowledgments

I would like to thank Mr Draa Amer for thoughtful and helpful comments on earlier drafts at this paper.

References

1. B. Faqihi, N. Daoudi, and R. Ajhoun. From the Collaboration of Companies to the Interoperability of Information Systems (Concepts and Perspectives). International Journal of Information and Education Technology, Vol. 5, No. 3, March 2015.
2. N.Noy. Semantic integration: a survey of ontology-based approaches. SIGMOD Rec., Vol. 33, No. 4, 2004.
3. JérômeEuzenat, Pavel Shvaiko Ontology Matching Second Edition, Springer-Verlag, Berlin Heidelberg (DE), 2013.
4. Faycal Hamdi, Brigitte Safar, Chantal Reynaud, Haifa Zargayouna. Alignment-based Partitioning of Large-scale Ontologies. Chapter in Advances in Knowledge Discovery And Management, Vol 292, Springer. 2010.
5. Tanni Mittra1 and Muhammad Masroor Ali1, 8th International Conference on Electrical and Computer Engineering. 20-22 December, 2014, Dhaka, Bangladesh
6. Gayo Diallo, Mouhamadou Ba. Effective Method for Large Scale Ontology Matching . Proceedings of the 5th International Workshop on Semantic Web Applications and Tools for Life Sciences. Paris, France, November 28-30, 2012.

7. Wei Hu, Yuzhong Qu, Matching large ontologies: A divide-and-conquer approach. Data & Knowledge Engineering journal. 2008.
8. Tanni Mittra, and Muhammad Masroor Ali. Ontology Matching by Applying Parallelization and Distribution of Matching Task within Clustering Environment. 8th International Conference on Electrical and Computer Engineering. 20-22 December, 2014, Dhaka, Bangladesh.
9. Holland, J. Adaptation in Natural and Artificial Systems. MIT Press, Cambridge. 1992.
10. Thomas R.Gruber.: A translation approach to portable ontology specification. Knowledge Acquisition Knowledge Acquisition journal Volume 5, Issue 2, June 1993.
11. Gomez P. A. et Benjamins V.R : Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods, Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trend.
12. Michelle Cheatham . MapSSS Results for OAEI 2011.
13. Jorge Martinez-Gil, Enrique Alba, and Jos_e F. Aldana-Montes. Optimizing Ontology Alignments by Using Genetic Algorithms.
14. Junli Wang, Zhijun Ding, Changjun Jiang1. GAOM: Genetic Algorithm based Ontology Matching, Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)
15. MapPSO and MapEVO Results for OAEI 2011 Jurgen Bock, Carsten Danschel and Matthias Stumpp
16. Marc Ehrig and Steffen Staab. QOM – Quick Ontology Mapping Marc Ehrig and Steffen Staab. ISWC 2004, LNCS 3298. 2004. Springer-Verlag Berlin Heidelberg.
17. Jerome Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Francois Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab-Zamazal, and Cassia Trojahn Final results of the Ontology Alignment Evaluation Initiative 2011.
18. MapSSS Results for OAEI 2011 Michelle Cheatham

Electromagnetism-Like Algorithm for Traveling Salesman Problem with Random Keys and Lin-Kernighan Algorithm

S. Büyüksaatçı[1] and Ş.A. Baray[2]

1. *Istanbul University, Faculty of Engineering, Department of Industrial Engineering, Avcılar – İstanbul / Turkey*
sinemb@istanbul.edu.tr

2. *Istanbul University, Faculty of Engineering, Department of Industrial Engineering, Avcılar – İstanbul / Turkey*
baray@istanbul.edu.tr

Keywords : electromagnetism-like algorithm (EM), traveling salesman problem (TSP), combinatorial optimization, physics based metaheuristic algorithm

1 Introduction

Optimization is a process of searching for the *best solution* for a specific problem of interest by satisfying given constraints if it is possible to measure *good* or *bad*. Depending on the aim of the problem of interest, the meaning of the best solution can be most/maximum or least/minimum. Therefore the word "optimum" is used as a general term instead of best in optimization problems. Throughout the years, optimization process has been used by many disciplines like engineering, physics, mathematics, economics, administration, commerce, social sciences, and even politics. With the advancement of time and technology, the solutions of real life problems became numerous and sometimes an infinite number of optimum solutions could be possible [1]. On the other hand, the size of optimum solutions that prevents an exhaustive search, the complexity and difficult constraints of the discussed problems caused the complete optimization techniques to be inadequate. In literature, this type of problems took part in combinatorial optimization problems and is called NP-hard. The time needed to solve these problems with classical optimization techniques grew exponentially with the size of problem. In order to overcome the shortage of time and computational complexity, approximate methods have become common in the 1980's [2].

Approximate methods can be branched off into two subclasses in general: approximation algorithms and heuristic algorithms. An approximation algorithm for an optimization problem is a polynomial-time algorithm that for all instances of the problem produces a solution whose value is within a factor of α of the value of an optimum solution [3,4]. Thus, approximation algorithms return an approximate solution that has a bound from the global optimum. However, heuristic algorithms, which yield a reasonable solution to a specific problem but not guarantee to produce the optimum solution, do not formulate a bound [5].

Metaheuristic algorithms, which are also a class of approximate methods, have emerged in the 1980's. The word "*heuristic"* has its origin in the old Greek word "heuriskein", meaning the art of discovering new strategies (rules) to solve problems. The suffix "*meta*", also a Greek word, means upper level methodology. Fred Glover [6] was the first researcher to introduce the term metaheuristic in the paper "Future Paths for Integer Programming and Links to Artificial Intelligence" [3]. A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space. Learning strategies are used to structure information in order to find efficiently near-optimal solutions [7]. In general, metaheuristic algorithms have been specially developed to find a solution that is "good enough", where a processing time is "small enough".

The bases of metaheuristic algorithms vary greatly. Some algorithms express the optimization process by using approaches apparently unrelated to optimization, like behavior of swarms (e.g. particle

swarm optimization, ant colony optimization, bat algorithm), natural evolution (e.g. genetic algorithm), physical changes (e.g. simulated annealing, harmony search, electromagnetism like algorithm) while some are based on biological fundamentals (neural networks) or chemical processes (chemical reaction optimization). However, in general, all metaheuristics contain a randomness structure [8].

In literature, it can be easily seen that metaheuristic algorithms have become more popular in different research areas and industries. There are lots of studies about continuous optimization problems, combinatorial optimization problems, design problems, layout problems, manufacturing problems, routing and transportation problems, scheduling and sequencing problems, multi-objective optimization problems, classification or clustering problems etc. that are handled with metaheuristics. In this study, a hybridization of the electromagnetism-like algorithm (EMA) with random keys (RK) and Lin-Kernighan algorithm is used to solve the traveling salesman problem (TSP). In literature, there is no publication to show that the Lin-Kernighan algorithm has been combined with EMA in any other study. Therefore, this study will provide an insight for this hybridization.

When searching for studies that focus on EMA for TSP, the literature has rather little to offer. Wu and Chiang [9] showed the ability of the EMA for solving TSP. Wu et al. [10] combined the EMA with 2-optimum method to avoid trapping in local minimum for big size problems. They compared their method with traditional EMA on test examples with 5,10,14 and 16 cities and presented their algorithm's efficiency. Javadian et al. [11] proposed a discrete binary version of the EM algorithm for solving TSP and tested it on six test problems from the TSP library. They reported that their algorithm gave better results by using the less number of iterations compared to Wu and Chiang [9] and Wu et al. [10]. Wu et al. [12] introduced a hybrid EMA with optimum and 2-optimum methods and RK technique to solve an integer-valued TSP. They also used 3^3 factorial designs to analyze how the parameters in the EMA affect the performance of the hybrid methods. Then they tested their methods on 30,50,75,100-cities problems respectively and compared the results with ant colony systems, genetic algorithms and simulated annealing. Yurtkuran and Emel [13] examined the TSP in which each customer should be visited within a given time window. They combined EMA with variable bounding strategy and penalty approach to handle time constraints. Computational analysis on well-known benchmark instances showed that the combined algorithm converged to feasible regions and found the best or near-optimum results. The algorithm also outperformed the Beam search-ant colony optimization and compressed annealing methods in terms of computational time.

The rest of this paper is organized as follows. Section 2 provides a brief introduction to TSP and lays out its mathematical formulation. The traditional EMA and hybrid EMA with the RK technique and Lin-Kernighan algorithm are proposed in Section 3. The information for the handled problem is given in section 4. Section 5 summarizes the computational results and finally section 6 concludes the paper.

2 Traveling Salesman Problem (TSP)

The TSP is a well-known classical combinatorial optimization problem, which aims to find the shortest tour for a salesman while visiting all given n cities and finally returning to the point of departure. In this problem, each city is visited only once and the distances between the cities are known.

The TSP was initially studied in the 1930's by the mathematician and economist Karl Menger in Vienna and Harvard. But mathematical infrastructures connected to the TSP were handled in the 1800's by the Irish mathematician Sir William Rowan Hamilton and by the British mathematician Thomas Penyngton Kirkman [14]. Because of the problem's application areas, many studies on the TSP have been made to this date and it forms the basis of many different problems.

In literature, two different types of TSP exist: the symmetric TSP and the asymmetric TSP. In the symmetric form of the problem, the distance between cities A and B is equal to the distance between cities B and A. On the other hand, in the asymmetric form, these distances are different. Thus, for given n cities, the number of tours in the asymmetric and the symmetric TSP is $(n-1)!$ and $(n-1)!/2$ respectively [15].

The constituents of TSP are: n is the number of cities indexed by i and j, $i,j \in \{1,2,\dots,n\}$; c_{ij} shows the distance between two cities, x_{ij} are the decision variables. x_{ij} is equal to 1 when (i,j) is included in the tour, otherwise it is equal to 0. The TSP can be presented as follows:

$$\text{Minimize} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1}$$

$$\text{Subject to} \sum_{i=1}^{n} x_{ij} = 1, \qquad i = 1, \dots, n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, \dots, n \tag{3}$$

$$x_{ij} \in X, \tag{4}$$

$$x_{ij} = 0 \; or \; 1, \qquad \forall i, j = 1, \dots, n \tag{5}$$

3 Methodology

In this section, the algorithms used in this study to solve the traveling salesman problem will be described in general terms.

3.1 Electromagnetism-like Algorithm (EMA)

The principles of the electromagnetism-like algorithm were introduced by Birbil and Fang in 2003 [16]. The algorithm is based on the attraction and repulsion forces of the electromagnetism theory in physics and developed for solving unconstrained global problems with bounded variables to minimize non-linear functions. According to the EMA, each candidate solution (each point in the solution space) is a charged particle in the multidimensional space and the location of these particles in space are defined with their "position vectors". The charge of each particle is proportional to its objective function value and this charge also determines whether the particles in population will apply a repulsive or attractive force to each other. In connection with Coulomb's law, the particle repulses other particles if its objective function value is greater; otherwise it attracts others. After the total force acting on each particle is calculated similarly to the electromagnetic force vector, the particles move in direction of this resultant vector. The intention is to reach the optimum point with this movement.

The EMA takes part in population-based metaheuristic algorithms. Considering its feature, the algorithm likes PSO, ACO etc. where information of all of the individuals in the population is used. The EMA is effective to solve optimization problems with bounded variables as given in Eq. (6).

$$\begin{gathered}\text{minimize } f(x) \\ \text{subject to } x \in [\mathbf{L}, \mathbf{U}]\end{gathered} \tag{6}$$

where $[\mathbf{L}, \mathbf{U}] = \{x \in \mathbb{R}^n \,|\, L_k \leq x_k \leq U_k, \quad k = 1, \dots, n\}$, n is the dimension of the problem, U_k is the upper bound in the kth dimension, L_k is the lower bound in the kth dimension and $f(x)$ is the objective function.

The EMA consists of four phases: (I) the "initialize" phase that includes determination of the parameters and obtainment of the particles, (II) "calculation of total force" for each particle, (III) "movement of particles" in line with this force and (IV) "local search (neighborhood search)", which is made in order to reach local optimum.

The general structure of the EMA is shown in figure 1.

ALGORITHM EM (*m, MAXITER, LSITER, δ*)

m: number of sample points

MAXITER: maximum number of iterations

LSITER: maximum number of local search iterations

δ: local search parameter, $\delta \in [0,1]$

1. Initialize ()
2. iteration ← 1
3. **while** (iteration<*MAXITER*) **do**
4. LOCAL (*LSITER*, *δ)*
5. **F** ← Calc()
6. Move (**F**)
7. iteration ← iteration +1
8. **end while**

Figure 1 : General structure of the EMA

3.1.1 Initialize

In this phase, m sample points are randomly generated from the n dimensional hyper-cube and the population is constructed. The description of the space is presented in Eq. (7).

$$X = \left\{x \in \mathbb{R}^n \middle| \, x_k^i = L_k + \lambda(U_k - L_k), \quad k = 1, \dots, n; \quad i = 1, \dots, m\right\} \tag{7}$$

where *X* is the search space, $\lambda \sim Unif(0,1)$ for each dimension of x_k^i . After the entire sample points are generated, the objective function value ($f(x^i)$) is calculated for each one and the best function value, which is identified in Eq. (8), is stored.

$$x^{best} = \text{argmin}\{f(x^i), \forall i\} \tag{8}$$

3.1.2 Local search

The local search procedure aims to look for a better solution in the neighborhood of the obtained solution. Birbil and Fang preferred to use an ordinary procedure rather than local search algorithms that are proven to achieve powerful and good results. The iterations start with the calculation of the maximum feasible step length for each dimension as given in Eq. (9).

$$s_{max} = \delta\left(\max_k (U_k - L_k)\right) \tag{9}$$

Secondly a chosen x^i is assigned to a temporary point y to store the initial information. Then two random values, λ_1 and λ_2, are selected and point *y* moves along according to the described rule below:

$$\begin{cases} \text{if } \lambda_1 \geq 0.5, y = y - \lambda_2(s_{max}) \\ \text{if } \lambda_1 < 0.5, y = y + \lambda_2(s_{max}) \end{cases} \tag{10}$$

Finally the objective function value is computed for y and if it possesses a better solution, x^{best} is updated.

3.1.3 Total force calculation

As stated by Edward Wesley Cowan [17], the force exerted on a point via other points is inversely proportional to the distance between the points and directly proportional to the product of their charges. In this phase, the charge of each point is evaluated as,

$$q^i = \exp\left(-n\frac{f(x^i) - f(x^{best})}{\sum_{i=1}^m [f(x^i) - f(x^{best})]}\right) \tag{11}$$

where n is the dimension of the solution space. In Eq. 11, the signs are ignored unlike electrical charges. The direction of a particular force between two points is decided by comparing their objective functions.

After calculating the charge of each point, the total force F^i exerted on point x^i is calculated with the following Eq. (12).

$$F^i = \sum_{j \neq i}^{m} \begin{cases} (x^j - x^i) \dfrac{q^i q^j}{\|x^j - x^i\|^2} & if \quad f(x^j) < f(x^i) \\ (x^i - x^j) \dfrac{q^i q^j}{\|x^j - x^i\|^2} & if \quad f(x^j) \geq f(x^i) \end{cases} \tag{12}$$

The total force is can be expressed on a figure; Figure 2 shows the resultant force (F_3) on point 3 for minimization and maximization problems [18].

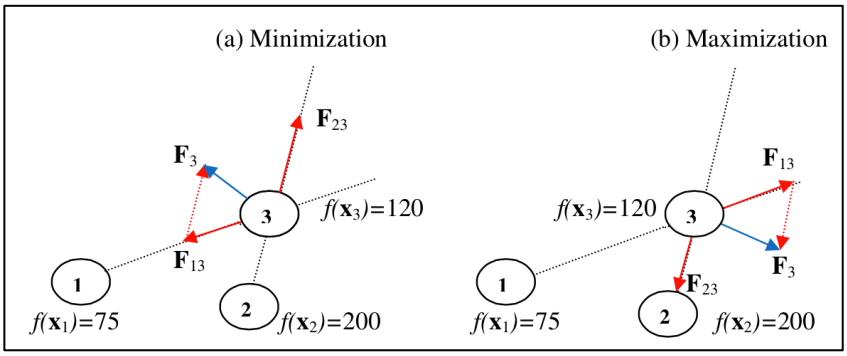


Figure 2 : Attraction and repulsion mechanisms on particle 3

3.1.4 Move along the total force

After calculating the total force F^i, the point x^i can move in the direction of the force by a random step length as defined by Eq. (13).

$$x^i = x^i + \lambda \frac{F^i}{\|F^i\|} (RNG) \tag{13}$$

where $\lambda \sim Unif(0,1)$. RNG is a vector denoting the allowed feasible movement towards the upper bound.

$$RNG = \begin{cases} u_k - x_k^i & if \quad F_k^i > 0 \\ x_k^i - l_k & if \quad F_k^i < 0 \end{cases} \quad k = 1,2,\dots,n \tag{14}$$

3.1.5 Stopping criteria

Birbil and Fang [16] used a maximum number of iterations to stop the EM procedure in their study. They stated that in general 25 iterations are satisfactory for converging to the optimum point for the medium difficulty functions. On the other hand, they suggested that the algorithm can be stopped if the found best point is not changed during the successive iterations.

3.2 The Hybrid EMA for TSP

As it is described in the previous section, TSP is a classical combinatorial optimization problem and it is based on discrete variables. On the other hand, The EMA is designed to cope with continuous optimization problems and uses real-valued variables. Thus, it is required to make some adaptations on EMA in order to implement it to the TSP. In this paper, two different methods are used during the implementation of the EMA to solve the TSP. The details of these methods are described below.

3.2.1 Random-Key (RK) Method

The Random Key method, which is developed in 1994 by James C. Bean [19], is a technique to rank continuous position values of particles. This method is used in wide-range problems in literature such as the

quadratic assignment problem, resource allocation problem, machine scheduling, vehicle routing, traveling salesman problem etc. The studies that aim to find solutions to combinatorial optimization problems frequently use the random key method. Debels et al. [20], Chen et al. [21], Tavakkoli-Moghaddam et al. [22], Chang et al. [23], Yurtkuran and Emel [18], Naderi et al. [24], Roshanaei et al. [25], Jamili et al. [26], Khalili et al [27] have integrated RK with EMA for solving project scheduling, machine scheduling, capacity constrained vehicle routing and workflow scheduling problems.

Lets suppose that for the 8-dimensional traveling salesman problem, the position vector of a particle is generated as [0.978, 0.348, 0.734, 0.519, 0.693, 0.187, 0.404, 0.837]. When the random key method is applied, the real-coded coordinate values are sorted in an increasing order and the new position vector becomes [0.187, 0.348, 0.404, 0.519, 0.693, 0.734, 0.837, 0.978]. The new-sorted indexes of this new position vector also represent the route. For example, the smallest coordinate value of the initial position vector is 0.187 with index=6. This means that customer/city 6 will be visited first. Then the route will be followed in a similar manner and will be 6-2-7-4-5-3-8-1.

3.2.2 Lin-Kernighan Algorithm

Lin-Kernighan algorithm has been developed by Lin and Kernighan [28] on the basis of Croes's [29] work. Croes presented a 2-optimum method for TSP to find the smallest closed loop that connects a number of points in a tour. In this method, 2 links of a given T tour is replaced with other 2 links that make the new tour length shorter. This iteration is continued until no more improvement is possible. One example of this step is given in Figure 3.

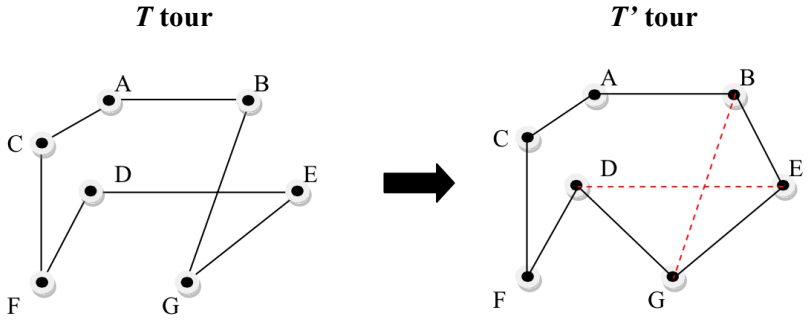


Figure 3 : 2-link replacement example

Lin and Kernighan generalized this method and they replaced k links until the shortest tour is achieved. They prevented the drawback of determining the number of k by enabling different k values during the iterations. In general, the larger value of k is preferred to find better solutions. However, as the value of k increases, the time complexity of the problem will also increase.

The Lin-Kernighan algorithm is used in the local search phase of the EM algorithm in this study.

4 Test Problem

In this study, a solution to the symmetric TSP involving Turkey's 81 cities is developed. The problem starts from any city and the seller returns to the starting point after all the cities are visited.

In the analysis of the problem, the distances matrix obtained from Turkey's highway data [30] was used.

Before starting the implementation of the proposed hybrid EMA to the problem, algorithm parameters, which are the number of sample points, upper and lower bounds and the stopping criterion, were defined.

In other studies, 20 particles have been found sufficient for the medium size problems. The computational time will increase as the number of particles increases. For this reason, 20 particles were adopted for analyzing the hybrid EMA in this study.

Upper and lower bounds, which were generated by considering some studies for EMA are listed in Table 1. Taheri et al. [31] chose the magnitude of the bounds large enough like 10^4. Therefore (0, 10^6) and (-,10^6 10^6) values were used for pair 6 and pair 7 respectively. Jolali et al. [32] used 0 and the number of machines for the lower bound and upper bound in their studies. Pair 8 was formed using the number of cities as upper bound.

	Lower Bound	**Upper Bound**		**Lower Bound**	**Upper Bound**
Pair 1	0	1	Pair 5	-10	10
Pair 2	-1	0	Pair 6	0	1000000
Pair 3	-1	1	Pair 7	-1000000	1000000
Pair 4	0	10	Pair 8	0	81

Table 1 : Upper and lower bound pairs from other studies

For each pair, 30 trials were run with 20 particles on the problem and the mean and standard deviation of the tour distance for 30 trials were calculated. The results are given in Table 2.

	Mean (km)	**Std. Dev. (km)**
Pair 1	10088.2	96.725
Pair 2	10072.9	135.392
Pair 3	10091.4	131.580
Pair 4	10111.4	106.140
Pair 5	10118.5	123.140
Pair 6	10190.7	156.184
Pair 7	10219.8	172.958
Pair 8	10101.1	131.158

Table 2 : Mean and standard deviation of tour distances for 30 trials

As seen from Table 2, the means of the tour distances calculated with pair 6 and pair 7 are prominently higher. Pair 1 and pair 2 have better results. However the standard deviation of the tour distance according to pair 1 is smaller with respect to pair 2. Therefore, it is decided to use pair 1 (0,1) for analysis of the proposed hybrid EMA.

Because of preventing that the algorithm converges quickly, 50 iterations, which can be considered as big enough, were chosen as the stopping criterion.

5 Computational Results

The proposed algorithms described in section 3 were coded in MATLAB 7.9.0 and run on a PC equipped with 2.3 GHZ Inter Core i5 processor and 4 GB RAM. The number of iterations, lower and upper limits and the number of particles were kept as given in section 3 and the proposed hybrid algorithm tested on problem.

Söyler and Keskintürk [33] suggested that 9954 km is the optimum result for the TSP that was calculated with ant colony optimization and includes Turkey's 81 cities. Other methods' results, which were also presented in [33] and the hybrid EMA result are given in Table 3.

METHODS	**TOTAL TOUR DISTANCES**
Ant colony optimization	9954
EMA combined with random key method and Lin-Kernighan algorithm	10005
2-optimum algorithm	11136
Nearest neighbor search algorithm	13120
Lin-Kernighan algorithm	14392

Table 3 : Comparison of different methods' results

The proposed hybrid EMA produced a close result to the best distance found by Söyler and Keskintürk. The algorithm also outperformed the 2-optimum algorithm, nearest neighbour search algorithm and Lin-Kernighan algorithm.

6 Conclusion

In this study, a hybridization of the electromagnetism-like algorithm (EMA) with random keys (RK) and Lin-Kernighan algorithm is used to solve the traveling salesman problem (TSP), which involves Turkey's 81 cities. Before starting the analysis, appropriate number of sample points, upper and lower bounds and stopping criterion for EMA were investigated. The real distances that were obtained from Turkey's highway data were used for computational analysis. The results showed that the hybrid EMA finds a near-optimum solution for the problem. The hybrid algorithm also achieves better results with respect to the 2-optimum algorithm, nearest neighbour search algorithm and Lin-Kernighan algorithm. For future work, it can be suggested that the hybrid algorithm can be tested on other optimization problems like the vehicle routing problem, facility layout problem etc. which are related with TSP.

Acknowledgement

The authors thank for support by Scientific Research Projects Coordination Unit of Istanbul University under Project number BYP-2016-21005.

References

[1] Antoniou, A., & Lu, W-S. (2007). *Practical Optimization-Algorithms and Engineering Applications,* Springer Science and Business Media, LLC, ISBN: 0-387-71106-6.

[2] Gonzalez, T. F. (Ed.). (2007). *Handbook of approximation algorithms and metaheuristics*. CRC Press.

[3] Talbi, E-G. (2009). *Metaheuristics from Design to Implementation*, John Wiley & Sons, Inc., Hoboken, New Jersey, ISBN: 978-0-470-27858-1.

[4] Williamson, D. P., & Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge university press.

[5] Rothlauf, F. (2011). *Design of modern heuristics: principles and application*. Springer Science & Business Media.

[6] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, *13*(5), 533-549.

[7] Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations research*, *63*(5), 511-623.

[8] Yang, X-S. (2010). *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, ISBN: 1-905986-28-9.

[9] Wu, P., & Chiang, H. C. (2005). The Application of electromagnetism-like mechanism for solving the traveling salesman problems. In *Proceeding of 2005 Chinese Institute of Industrial Engineers Annual Meeting, Taiwan, ROC*.

[10] Wu, P., Yang, K. J., & Fang, H. C. (2006, August). A revised EM-like algorithm+ K-OPT method for solving the traveling salesman problem. In *Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on* (Vol. 1, pp. 546-549). IEEE.

[11] Javadian, N., Alikhani, M. G., & Tavakkoli-Moghaddam, R. (2008). A discrete binary version of the electromagnetism-like heuristic for solving traveling salesman problem. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence* (pp. 123-130). Springer Berlin Heidelberg.

[12] Wu, P., Hung, Y. Y., & Chiang, H. C. (2012). A Hybrid Electromagnetism-like Mechanism: A Metaheuristic Algorithm for Solving the Travelling Salesman Problem. *International Journal of Operations Research*, *9*(1), 53-65.

[13] Yurtkuran, A., & Emel, E. (2014). Efficient Constraint Handling in Electromagnetism-Like Algorithm for Traveling Salesman Problem with Time Windows. *The Scientific World Journal*, vol. 2014, 1-12.

[14] Maredia, A. (2010). *History, Analysis, and Implementation of Traveling Salesman Problem (TSP) and Related Problems* (Doctoral dissertation, University of Houston-Downtown).

[15] Bonyadi, M. R., Shah-Hosseini, H., & Azghadi, M. R. (2008). *Population-based optimization algorithms for solving the travelling salesman problem*. INTECH Open Access Publisher.

[16] Birbil, Ş. İ., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of global optimization*, *25*(3), 263-282.

[17] Cowan, E. W. (1968). *Basic electromagnetism*. Academic Press, New York.

[18] Yurtkuran, A., & Emel, E. (2010). A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. *Expert Systems with Applications*, *37*(4), 3427-3433.

[19] Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, *6*(2), 154-160.

[20] Debels, D., De Reyck, B., Leus, R., & Vanhoucke, M. (2006). A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research*, *169*(2), 638-653.

[21] Chen, S. H., Chang, P. C., Chan, C. L., & Mani, V. (2007). A hybrid electromagnetism-like algorithm for single machine scheduling problem. In *Advanced intelligent computing theories and applications. With Aspects of Artificial Intelligence* (pp. 543-552). Springer Berlin Heidelberg.

[22] Tavakkoli-Moghaddam, R., Khalili, M., & Naderi, B. (2009). A hybridization of simulated annealing and electromagnetic-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness. *Soft Computing*, *13*(10), 995-1006.

[23] Chang, P. C., Chen, S. H., & Fan, C. Y. (2009). A hybrid electromagnetism-like algorithm for single machine scheduling problem. *Expert Systems with Applications*, *36*(2), 1259-1267.

[24] Naderi, B., Tavakkoli-Moghaddam, R., & Khalili, M. (2010). Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan. *Knowledge-Based Systems*, *23*(2), 77-85.

[25] Roshanaei, V., Balagh, A. K. G., Esfahani, M. M. S., & Vahdani, B. (2010). A mixed-integer linear programming model along with an electromagnetism-like algorithm for scheduling job shop production system with sequence-dependent set-up times. *The International Journal of Advanced Manufacturing Technology*, *47*(5-8), 783-793.

[26] Jamili, A., Shafia, M. A., & Tavakkoli-Moghaddam, R. (2011). A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem. *Expert Systems with Applications*, *38*(5), 5895-5901.

[27] Khalili, M., & Tavakkoli-Moghaddam, R. (2012). A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. *Journal of Manufacturing Systems*, *31*(2), 232-239.

[28] Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, *21*(2), 498-516.

[29] Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, *6*(6), 791-812.

[30] http://www.kgm.gov.tr/SiteCollectionDocuments/KGMdocuments/Root/Uzakliklar/ilmesafe.xls

[31] Taheri, S. H., Ghazvini, H., Saberi-Nadjafi, J., & Biazar, J. (2007). A hybrid of the restarted Arnoldi and electromagnetism meta-heuristic methods for calculating eigenvalues and eigenvectors of a non-symmetric matrix. *Applied Mathematics and Computation*, *191*(1), 79-88.

[32] Jolai, F., Tavakkoli-Moghaddam, R., Golmohammadi, A., & Javadi, B. (2012). An Electromagnetism-like algorithm for cell formation and layout problem. *Expert Systems with Applications*, *39*(2), 2172-2182.

[33] Söyler, H., & Keskintürk, T. Karınca Kolonisi Algoritması İle Gezen Satıcı Probleminin Çözümü. *8. Türkiye Ekonometri ve İstatistik Kongresi*, 1-11.

A Plant Propagation Heuristic Scheme for the Consecutive Ones Submatrix Problem

Abdellah Salhi, Eric Fraga, Salim Haddadi

A (0,1)-matrix *M* has the Consecutive Ones Property (C1P) if there exists a permutation of columns which puts all the 1's consecutively in all rows [4]. The Consecutive Ones Submatrix (C1S) problem is, given *M*, find the largest number of columns of M forming a submatrix with the C1P property. This problem occurs in a number of applications including genome sequencing in bioinformatics [2], chronologically ordering archeological deposits [3], railway optimization [1] and so on. But, the interest here is in potentially exploiting this submatrix to solve pure integer linear programming problems as mixed integer ones taking advantage of the submatrix with C1P property the variables of which can be considered as continuous in the solution process.

Recall that the Plant Propagation Algorithm (PPA) [] implements the idea that plants in good spots send many short runners (exploitation) and those in poor spots send few long runners (exploration). To implement this idea we first need to represent a plant/solution which is a permutation of the columns of (0,1)-matrix *M*. We also need to rank these solutions/plants according to how good/fit their locations are. An objective or fitness function is therefore required. Such a function is not obvious at all. However, if we were to favour column permutations leading to matrices with large blocks of 1's in all rows of the given matrix, then the following function may do. Consider for a given row of (0,1)-matrix *M* the expression

$$\sum_{i=1}^{k} \sum_{j=1}^{l_i} j, \qquad (1)$$

where k is the number of blocks of 1's in the row, and l_i is the length of block i. For example row (1 0 1 1 0 0 1 1 1 1 0) of *M* has k=3 blocks of 1's, with lengths l_1=1, l_2=2 and l_3 = 4. Its fitness according to the above formula is 1 + 3 + 10 = 14.

This should be repeated for every row and then the fitness of the matrix is computed by adding up the fitness values of its rows.

To implement PPA, we also need to represent the notions of short and long runners. Given the discrete nature of the solutions (permutations) the usual metrics available in continuous spaces are not readily applicable here. We settle for a simple idea akin to the 2-Opt rule of Lin and Kernighan [5]. A short runner corresponds to a single column swap, while a long runner is implemented with a sequence of single column swaps or a single swap of blocks of columns of the same individual (see example below). With this in mind, a PPA scheme for max-C1S problem can be devised.

The paper will explain such an implementation and consider alternative objective functions. It will then be tested on non-trivial matrices arising from practical problems such set-covering and compared against the Genetic

Algorithm and possibly other heuristics. The paper will also describe a procedure for recognizing the maximum submatrix with C1P property after applying PPA and GA and how it can be exploited to speed up the solution of pure integer programming problems. Computational results will be included.

References

[1] S. Mecke and D. Wagner. Solving geometric covering problems by data reduction. In Proceedings of the 12th Annual European Symposium on Algorithms (ESA '04), volume 3221 of LNCS, pages 760–771. Springer, 2004.

[2] J. Tan and L. Zhang. The consecutive ones submatrix problem for sparse matrices. Algorithmica, 48(3):287–299, 2007.

[3] D. Kendall. Incidence matrices, interval graphs and seriation in archeology. Pacific Journal of mathematics, 28(3):565_570, 1969.

[4] A.Tucker. A structure theorem for the consecutive 1's property. Journal of Combinatorial Theory, Series B, 12(2):153_162, 1972.

[5] S.Lin and B.W. Kernighan. An Effective Heuristic Algorithm for the Traveling-Salesman Problem". Operations Research **21** (2): 498–516, 1973.

[6] A.Salhi and E.S.Fraga. Nature-Inspired Optimisation Approaches and the New Plant Propagation Algorithm, Proceedings of the ICeMATH2011, pp. K2-1 to K2-8, 2011.

[7] B.I.Selamoglu and A.Salhi. The Plant Propagation Algorithm for Discrete Optimisation: The Case of the Travelling Salesman Problem, In Nature-Inspired Computation in Engineering, X-S. Yang (Ed.), pp. 43-61, 2016.

A Case Study for health care Supply Chain

Sylvain Bertel*. Pierre Fenies**

* CRCGM (EA 384), Université D'Auvergne,Clermont-Ferrand, France, (e-mail :Sylvain.bertel@udamail.fr.
** CEROS (EA 4428), Université Paris Ouest Nanterre La Défense**,** Nanterre , France, (e-mail :pfenies@u-paris10.fr)

Abstract: This article deals with the reorganization of a French dental care hospital using a modeling approach by multiple incremental process. This article is original by its dual approach in management science and engineering. This work is a part of a double complexity (algorithmic and systemic) in the implementation of tools for the hospital supply chain. For this purpose, several formal multicriteria models have been made. Later, they will subsequently help for the dialogue with the various stakeholders of the dental hospital chain for its reorganization.

Keywords: Health care system, knowledge management, decision making.

I. Introduction (*HEADING 1*)

We present in this paper a methodology for Business Process Management and apply it to the specific context of health care systems. The presentation, characteristics and objectives of French Dental Care Center is summarized and analyzed in section 2. Section 3 of this paper present a deployment of this methodology on a supply chain, (a dental University Hospital located to Clermont-Ferrand), the knowledge collected and the use made of this gathered and formalized knowledge.

II. French Public health care

A. Presentation

French Public health care agencies consider that the problems of French hospitals come from a lack of organization rather than a lack of resources (Belorgey, 2010). Hospital systems have different complexities:
- A systemic complexity (structural and functional) which results in the difficulty of evaluating simple performance criteria.
- An algorithmic complexity refers to the complexity of combinatorial optimization problems to solve. It results in the search for a solution that optimizes one or more performance criteria.

This modeling project concerns a university Dental Care Center and wants to develop new tools to monitor operational performance. To study the use of the tool during the design phase, we use a methodological device based on participant observation, interviews and questionnaires. The first paragraph of this section presents the characteristics of the French Dental Care Center (DCC) studied and the objectives of the project flow modeling of this public institution. The second paragraph describes the results associated with the phases of the business process specification DCC.

B. Characteristics and objectives of DCC modeling

The characteristics of DCC are described below:

- DCC is organized to give courses of education and care,
- These courses give rise to clinical vacations for students of the Faculty of Dental Surgery. These educational streams, however, are attached to hospital units;
- DCC system works in "Omni Practice", where medical resources are in turn care for the patient (not the patient that flows from one service);
- DCC has an objective to train future dentists with "in situ" cases;
- DCC hosts a patient base consists of difficult social cases, people in serious financial distress or retirees who also come to discuss;
- DCC was less concerned by the imperatives of rationalization and improvement productivity of health care systems, because their main objective is to give practices for future dentists;
- A student follows the patient from beginning to end of treatment;
- The Hospital Practitioners Doctors control and validate the acts of treatments performed by students;
- A student has to obtain a number of points to pass to the next year, reflecting a number of validated acts in each course;
- The DCC are among the Public Hospital, and financial constraints become stronger (occupancy armchairs, team performance) streamlining care DCC while maintaining the quality of learning students in an objective satisfaction patient base is a necessity;
- While making parties of public hospitals, DCC are built in faculties of dentistry and are often physically independent of other public health care system.

Clermont-Ferrand DCC will move from the actual University of Dentistry and be physically integrated in the New Hospital d'Estaing. The patient demand will increase of 20% to 40%, due to the novelty of the situation, but human resources are constant. There is therefore a need to rationalize the teaching and caring for activity to cope with this challenge. It shows

the proposed modeling and assessment of current and future performance of DCC with the following objectives:

- Build a generic tool that can be extended to other DCC, plan current and prospective activities,
- Develop managerial methods change way to manager students and hospital doctors, becoming a facilitator to improve the structure.

C. State of the art

In this section, we propose to evaluate our problem by several performance criteria for hospital care chain quoted in the literature of operations research. The various criteria considered in our problem are:

- The physical criteria (number of treated patients, rates of acts made / received number of patients)
- The financial criteria (Total realized costs, available Cashflow available)
- Social criteria (Respect for the staff specialty and time slots)
- Environmental criteria (CO2 Rejection, energy calculation used to perform all acts).

The table below shows the referenced articles according to the optimization criteria of the care problem. We can see in table below (See TAB 1) that few articles take into account simultaneously all the performance criteria of hospital care chain. Indeed, they do not present systematic approaches to solving our problem [[Toma et al, 2010], [Mc Intyre, 2009], [Dos Santos et al, 2009]]. Some articles propose methods but on incomplete hospital problems [[Sghaier et al, 2009], [Rodier, 2010], [Barichard et al, 2009], [Bertel et al, 2008], [Féniès 2006]]. This non-simultaneity causes tension between the different actors in the process of hospital chain: those who want to meet the legitimate needs of patients in fairness [[Sghaier et al, 2009], [Belorgey, 2010]], and those who want respect budget performance imposed by the Financial Officer [[Toma et al, 2010], [Mc Intyre, 2009], [Dos Santos et al, 2009]].

	Physical Criteria	**Financial Criteria**	**Social Criteria**	**Ecologic Criteria**
Barichard &al, 2009] , [Bertel & al, 2008], [Féniès, 2006]	X	X		
Rodier, 2010], [Hernandez-Matias &al, 2008]	X			
Belorgey, 2010]			X	
Sghaier et al, 2009]			X	X
Toma et al, 2010], [Mc Intyre, 2009], [Dos santos et al, 2009]	X	X	X	X

Table 1: State of the art

D. Analysis and specification of studied DCC

Analysis and system specification steps have a goal to raise the knowledge associated with this system to build scenarios to improve the functioning of the organizational process system and test the relevance of these organizational scenarios in models of decision support (exact models, computer simulation ...) Analysis and specification steps have been carried out by two people during 5 months (April to August 2012). All staff of DCC was interviewed during the vacations and 97 care processes made by the students were fully analyzed. The 2 observers were accepted by the members of the structure as colleagues and not "external", which is particularly rare in the field of health public. In table 1, we present one of the use of this knowledge formalization, which are the diagnosis of the cause and consequences of the DCC dysfunction. Figure 1 shows flow formalization for the DCC.

Dysfunction	**Causes**	**Conséquences**
Fluctuating occupancy chair	-Appointment of non-compliance by patients -Changes in students courses based on academic period	- Bad distribution of the load on the year - Student Learning is not enough good and standardized
Within two months to obtain an appointment in Home consultation	-Vacation Home = bottleneck (single official entry to the clinic visits, 6 patients vacations). -Unavailable schedules from the end of the academic year to September -Difficulties to find a sufficient number of students wishing to take care of a new patient at	-Patient dissatisfaction about the delay. - Inefficient Vacation in terms of business for the center. -Patients waiting students and students without patients

	the end of the year	
Vacations are sometimes inefficient	- Multiple students displacements: Search-and / or practitioners' expectations. - Overloading-supervisors at certain times. - Lack of mastery in the implementation of actions by students in training (very long time for the 4th year…)	-□Support patient and long extension of the vacations duration. - Intervention-chair-reinforced practitioners by year of study and discipline of students (especially pedodontics cursus).

Table 2: Causes and consequences of malfunctions

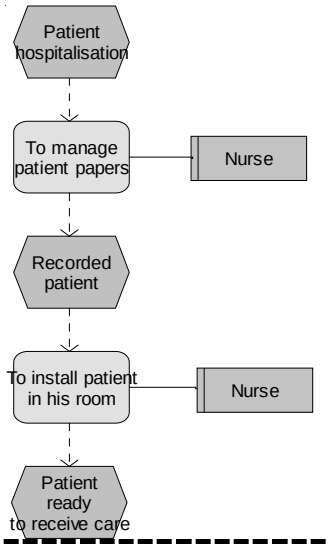


Figure 1: Example of physical flow in DCC

The official standards may not be used in this study for the use of students to perform acts (under the control of hospital practitioners); therefore, processing time is different from those made by a professional.

E. Design and implantation of computer models

We do not present all the models which were designed for the DCC project, but simply an example of a model allowing to affecting a caregiver to a patient. It is a tool to plan operatively the load. Our first approach is to provide care to the pole a tool to affect patients at various care stations in the hospital. The pole of care has several workstations.
Each workstation has 3 categories of hospital staff (intern medical, nurse, and help career). It should allow to affecting trainee doctors, nurses and caregivers to posts; our approach is to model the complete operation flow of the dental service with a linear modeling. The originality of this model is in the integration of pedagogical criteria in addition to conventional criteria. The process is as follows: the doctor has expertise, student also. The patient has symptoms that require a doctor's specialty and or student in dentistry. The goal is to assign the patient to the best resource available to perform the act on the patient. The patient is satisfied if he restarts quickly and that the act was performed. To solve this problem, we propose an approach by phase more exactly 3 phases using two different models. This approach is described in Figure 2.

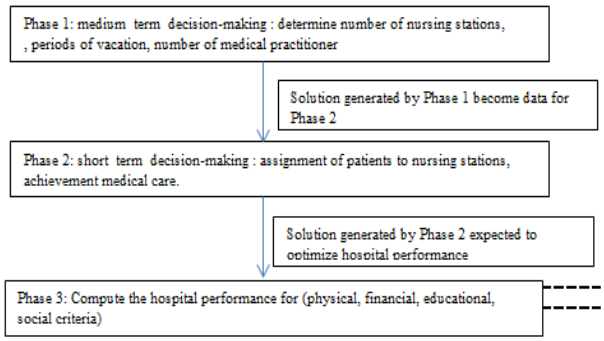


Figure 2: General approach to solving problems of resource allocation

D.1 Description of the Phase 1
The first phase uses a linear model that affects student's vacation.
Data :
- The number of promotions of students and the number of students per class,
- Vacations mandatory for students by type (conservative dentistry, prosthodontics, emergencies,
pedodontics ...),
- Number of shifts available throughout the week for students,
- Number of positions available care attendances,
- Information on care positions (financial cost ...),
- Quantity type vacations to ensure period.

Constraints:
- A student has to do more than one session at the same time,
- Each vacation on a post care is affected by a student,
- The number of vacations per student has to be respected,
- Respect the number of shifts required per period,
- A shift is assigned for one type of vacation.

Decision variables:
- Variables allow Boolean whether a student is assigned to a position of care on a vacation and a type of vacation.

Objective function: - In this paper, We Minimize the number of shifts unaffected.

D.2 Description of the Phase 2

The second phase uses linear models that affect patients with student's vacation.

Data:

-Acts to be carried with their characteristics (redemption price of the act, specialty act, act duration)

-The Rooms (budget cost, possible specialties, Energy Expenditure.)

-The Staff (Cost of the employee, legal working time, availability of an employee, physician specialty)

Act i	Processing time (per hour)	inancial cost for act i for 1 hour	Specialty act to perform i	Rejection act i CO2 / hour	Refund Act i
	Duree_acte[i]	Coût_acte[i]	Spec_acte[i]	Rejet_acte[i]	Remb_acte[i]
Act 1	0.5	100€	1	5g	300€
Act 2	1	50€	2	8g	150€

The doctors characteristics :

Interne	specialty	Financial Cost for 1 hour
	Spec_Med[i]	Cout_Med[i]
M1	1	20€
M2	2	30€
M3	3	40€

Doctors Availability

Dispo_Med[i]	Période 1	Période 2
M1	1	0
M2	1	1
M3	1	1

Nurses availability:

Dispo_Inf[i]	Période 1	Période 2
I1	1	1
I2	1	0
I3	1	1

Help carers Availability :

Dispo_A[i]	Période1	Période 2
A1	1	0
A2	1	1
A3	1	1

The room specialty for act:

Spec_Salle[i]	specialty 1	Période 2
1	1	0
2	1	1
3	1	1

The rooms characteristics :

Spef_Salle[i]	Energy Cost / hour	Rejection of CO2/hour
	Cout_salle[i]	Rejet_salle[i]
1	10€	5g
2	20€	20g
3	10€	5g

Variables :

X_ijk= 1 if act i is processsed at period j for the care station k. 0 otherwise.

M_djk=1 if doctor d is assigned to the care period j for the care station k. 0 otherwise.

I_fjk=1 if nurse f is assigned to the care period j for the care station k. 0 otherwise.

A_ajk=1 if carer a f is assigned to the care period j for the care station k. 0 otherwise..

problem constraint:

Constraint 1 : Each act i must be assigned to more than 1 care station and 1 period.

$$\sum_{j}\sum_{k} X_{ijk} \leq 1 \ \forall i$$

Constraint 2 : For each period, a doctor should be assigned to more than 1 care station.

$$\sum_{k}\sum_{d} M_{djk} \leq 1 \quad \forall j$$

Constraint 3 : For each period, a nurse must be assigned to at most 1 care station.

$$\sum_{k}\sum_{f} I_{fjk} \leq 1 \quad \forall j$$

Constraint 4 : For each period, a help carer must be assigned to at most 1 care station.

$$\sum_{k}\sum_{a} A_{ajk} \leq 1 \quad \forall j$$

Constraint 5 : Respect of doctor specialty allocated for each act i.

$$\mathrm{Spec}_{\_acte[i]} + \mathrm{HV} \times \left(1 - \mathrm{X}_{ijk}\right) \leq \sum_{d} M_{djk}.Spec_{\_Med[d]} \quad \forall i \ \forall j \forall k$$

Constraint 6 : Respect of working time (doctors, nurses and caregiver) for each period

$$\sum_{i} duree\_acte[i] \times X\_ijk \leq 2 \ \forall j \ \forall k$$

Constraint 7 : Each workstation must have the 3 categories of personnel (doctor, nurse, carer).

$$\sum_{i} X_{ijk} \leq \sum_{d} M_{djk} \quad \forall j \ \forall k$$

(you need at least one doctor to the position k)

$$\sum_i X_{ijk} \leq \sum_f I_{fjk} \quad \forall j \; \forall k$$

(you need at least one nurse to the position k)

$$\sum_i X_{ijk} \leq \sum_a A_{ajk} \quad \forall j \; \forall k$$

- (you need at least one carer to the position k)
- The criteria for the model are as follows:

Criteria 1 : This physical criteria is a criteria that tries to maximize the number of patients treated

$$maximize \sum_i \sum_j \sum_k X_{ijk}$$

This physical criterion to know the customer satisfaction rate. It is the primary criterion for doctors.

Criteria 2 : This financial criterion seeks to maximize profit made, so as to minimize the total cost of procedures performed

:

$$min \; \sum_i \sum_j \sum_k X_{ijk} * \text{Cout\_salle}[k] * duree_{acte[i]} + \sum_d \sum_i \sum_j \sum_k M_{djk} * cout_{med[d]} * (duree_{acte[i]} * X_{ijk}$$

This financial criterion is the primary criterion for the hospital manager.

Criteria 3 : The environmental criterion seeks to minimize the CO2 emissions generated from operating activities

:

$$minimize \sum_i \sum_j \sum_k X_{ijk} * \text{rejet\_salle}[k] * duree_{acte[i]} + \sum_i \sum_j \sum_k X_{ijk} * \text{rejet\_acte}[i] * duree\_acte[i$$

This environmental criterion is not a conscious criterion for hospital staff but it can measure the impacts of activities on environment

.

Criteria 4 : This social criterion seeks to maximize the work of doctors in their specialties:

$$\sum_d \sum_j \sum_k M_{djk} * duree_{acte[i]} * spec_{med[d]} - \sum_i \sum_j \sum_k X_{ijk} * duree_{acte[i]} * spec_{acte[i]})$$

This criterion is important to ensure good internal training.

These models can be applied so lexicographically choosing the hospital setting. The purpose of this model is to find an optimal solution for each of the four criteria expressed in section C but to formalize the operation of the service to reflect on its future reorganization.

Note: This step has to be restarted several times to challenge the initial planning for the arrival of a patient emergency or lack of student resources.

D.3. Resolution linear models

Different models have been implemented in CPLEX and solved for small instances in order to analyze and validate the objectives of DCC. However the size of actual instances to process the DCC are:

-10000 Patients to the year 400 / day,
-50 And 75 nursing stations programmed simultaneously,
- More than 300 students affected,
- 25 hospital professors and doctors,

Solvers market remains ineffective because these models are too large in memory or computation time facing the size of the real problems. It is impossible to use them to present solutions to decision makers.

It is therefore necessary to validate our choices by looking for other resolution tools, and propose heuristics problem solving assignment.

It should also provide models of discrete event simulation to test different organizational scenarios and improving management practices of the DDC.

III. A modeling methodology for multiple and incremental processes

Health care system should be modeled as a Supply Chain (Pascal, 2000 ; Fenies, 2006, 2011). A Supply Chain is a coalition of autonomous organizations coordinated by an integrated logistics process. A process is composed by a finite number of activities, and each activity may even become a process in another granularity level; the evaluation of a process as a whole becomes the central element to evaluate the performance of a complex system.

The state of the art developed by (Weske, 2009) shows the important points about the process modeling, and limitations of current methods and tools. Given the complexity of the objects containing business processes and their human dimension and non- mechanistic design as the construction of objects of research for decision support requires for the computer and to engineer integrate the process modeling concepts of other scientists including Humanities and Social Sciences fields. We seek to establish a modeling methodology for the evaluation of organizational processes. The implementations of this modeling methodology on a health care supply chain enable us to provide solutions to model and evaluate its logistics processes. This methodology, based on a systemic approach and integrating tools analysis, specification, and modeling performance evaluation should enable the development of a model description of the structure and operation of the studied process and its translation into one or more models computer to assist in its evaluation.

In this paper, we propose to integrate in ASDI methodology the use of process modeling ; ASDI, which means Analysis Specification, Design and Implementation is a modeling methodology (Sarramia 2002, Chabrol et al 2007, Fenies et al, 2007, Rodier, 2010, Fenies 2011). This methodology should, while allowing the collaboration of experts in modeling, allowing design research objects incorporating all flows and decision support on business processes, including the context of public organizations and health care system. Thus, the left part shows the modeling phase of the field while the right side shows the process modeling a system (Figure 3).

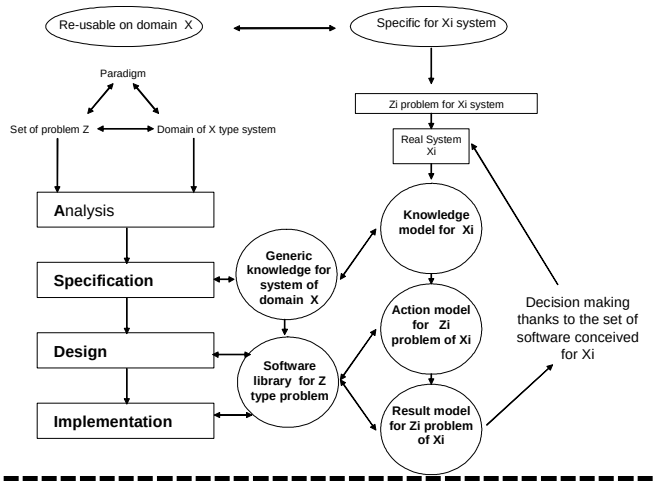


Figure 3: ASDI Generic component in the Methodology.

This modeling project of Dental Care Center allows presenting the characteristics of the French Dental Care Center (DCC) studied and the objectives of the project flow modeling of this public institution in section 1 but also to discover dysfunctions of DCC (see table 2). However, if Medical staff knew some of these problems but none were able to provide a full comprehensive list. Table 2 gives this full comprehensive list and propose scenarios to improve DCC running. The solutions proposed and used in the simulation model have been discussed with the Head of Department and the medical staff.

Dysfunctions	Proposed scenarios to improve DCC running
Fluctuating occupancy chairs	- Recall patients before their appointment (students doing "home" vacation during downtime ...) - - Forcing students to follow always the same patient in order to correctly assign patients directly or through a supervisor (first visit) and know in real time the available seats, - Studying seasonality of clinical activity to better plan the activity of DCC.
Within two months of obtaining an appointment in Home consultation	- Delete the vacation "home" (and thus integration of medical and dental assessments either one of the other actions in prosthesis and Pedodontics vacation) - Empowering the admissions office to assign patients directly to a student or practitioner who will assign one of its students by level of difficulty of the actions and their availability (requires that all students be present).
Some vacations are inefficient	- Skip to 1:30 for vacations instead of 2h - Place a work for "four hands": 16 students per box of 8 armchairs; for example, one plays the role of the dentist and the other that of the assistant.

Table 2: Organizational scenarios

3.4 The solutions envisaged by the CSD:

Simulation model use several solutions considered in table 2 to reduce the dysfunctions of the care center. This simulation model described in figure 5 used original data and changed data:
Original data are:

- care personal (the Head of Department can't increase its workforce)
- patients (We can't predict the future)

Changed data are:

- New scheduling rules (rules implemented in simulation model)
- Architecture of new hospital (2 Essential changes : new dental care center have no vacation home and more seats)

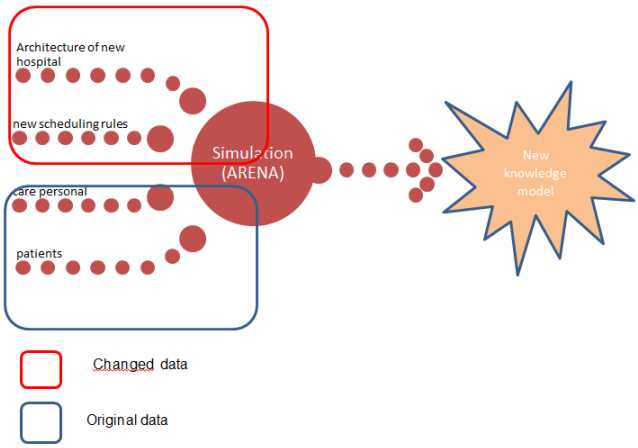


Figure 4: Simulation model.

We list below the scheduling rules retained implemented in simulation model :

- Remove the vacation home that serves neck,
- Changing schedules and duration of student vacations,
- Allow entries to assign patients to student vacations,
- To limit movement of Clinical Supervisors(A supervisor have a box of 8 seats affected in the dental care center).

The solutions implemented in simulation have been discussed with the Head of Department and the medical staff. The other rules are not fixed and need more discussions between the medical staff.

3.5. The Simulation results

In this section, we present the final results : Data and observation results obtained by the study of the health care center by students over a period of two months. A sample of 250 people was assessed using the quota method.

Figure 5 present observation results got by students in the health care according to average, maximum , minimum time spent by patient at each care station.
Previously, we described the characteristics of the health care center in section 2. Remember, that the vacation time is 2 hours for a student. By observing Figure 5, we can see that a patient may spend more than 2 hours in a care position. The consequence of this waiting period is that the student can't validate its healing session and the patient is not satisfied by this unusual wait times. This is due to the organization and operating rules of the care center

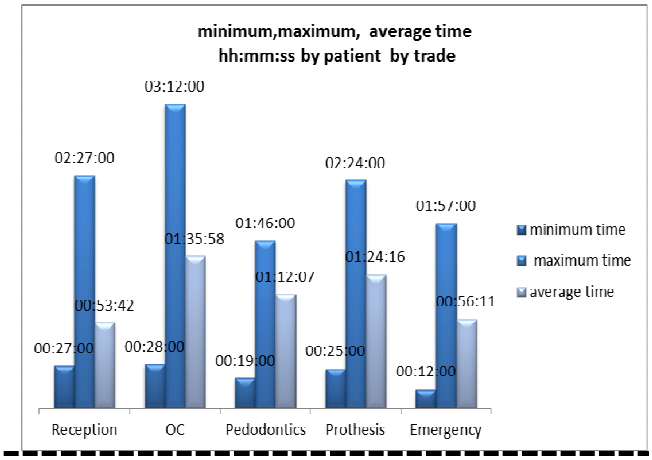


Figure 5: Time spent by patient at care station. (Observation results)

Applying the simulation model on the same parameters as those harvested by the students on the sample of 250 people, Figure 6 present results got by average, maximum, minimum time spent by patient at each care station. to the organization and operating rules of the care center have been modified in the health care center. We proposed in section 2 to reduce the vacation time of 30 minutes and incorporate the operating rules outlined in Table 2. It can be seen that these changes have helped to respect vacation time, since no patient spends more than allowed time at each workstation. Finally, which allows students to validate the care session and reduce the patient's length of presence at each care station.

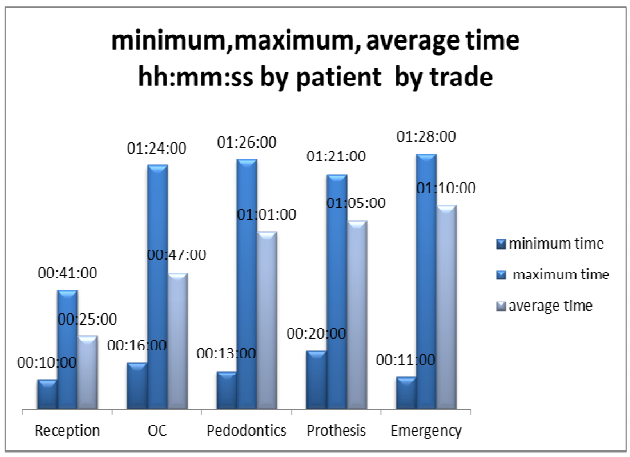


Figure 6: Time spent by patient at care station. (Simulation results).

To conclude this section,we present the other indicators collected by the simulation; :
- Waiting time for patients has decreased by 60% before management.
- We spend a vacation during 2 hours to 1h30 without reducing the number of patients treated.
- Improved efficiency: total processing time of patients has decreased by 35%.
- Reduced travel clinics Supervisors
- The development of nursing stations is validated by the simulation model.
The results given by the simulation model have helped decision makers to validate the new knowledge model with new rules of the DCC in the new hospital. Of course, all the rules are not present in this first operational model and it can easily be improved when all stakeholders will agree on procedures to follow. However, Nowadays, Managers have a tool to validate the changes and discuss with medical personal of Dental care center.

Acknowledgment

The authors thank Housni Boubaker and students of Management school for the help and the work done during the step of knowledge collection in DCC.

Conclusion

In this paper, we have proposed to implement a process view in ASDI. The proposed environment, which is a bridge between computer sciences, management sciences and operational research, gives a solution to conceive decision making tools which should also be also used in meeting in order to manage changes. The proposed model has been shown at a part of medical staff in order to show how we use collected data. Our model was used to reduce medical change resistance. Model presentations induce a lot of questions from medical staff about its future work.
Therefore, the extension of discrete event simulation and OR models as a tool for changes management will open a new

research field between computer sciences and management sciences.

This paper has also presented a methodological evaluation process of organizational running of a French care center dental service. Phase of information gathering (2 full-time staff on 2012) is complete. Phase of knowledge exploitation is well underway. If the modeling approach should be generalized to other public health services, the factual results are contextual to the studied system. However, the ability to generalize on all French DCC tools, diagrams and models decision support is a major issue for this research project which will enable it to overcome the purely local context of relocation.. Linear models are proposed to formalize the data, constraints and objectives but does not solve instances of actual size, hence the need to develop heuristics or models based on simulation. We develop simulation models based on scheduling rules (ARENA…) to:

- Propose a set of applied with real data to evaluate the performance of the DCC models.
- To test and propose new organizations for DCC.
- Assess the robustness of each of the proposed solutions.
- Propose approaches for addressing the field quickly re-allocation of patients to internal as well as students in the event of change in the daily schedule.

Nowadays, this tool is operational to help decision maker to take the good scenarios and then to discuss of these with medical staff of the dental care center.

References

[1] Barichard V. T'kindt, M. Ehrgott, X. Gandibleux (2009). Multiobjective programming and goal programming: theoretical results and practical applications. Springer, Hagen

[2] Belorgey, N. "L'hôpital sous pression''. La Découverte, Paris, 2010.

[3] Cooper R., Kaplan R., ''The Design of Cost Management System'', 2nd Ed., Prentice Hall International, London,Englewood Cliffs,1991 .

[4] Chabrol M., Fenies P, ''Le projet de modélisation et de simulation des flux du Nouvel Hôpital Estaing : aide a la décision versus management du changement''. Revue Française de Gestion Industrielle Volume 26, 1 2007.

[5] Fenies P, "Une méthodologie de modélisation par processus multiples et incrémentiels : application pourl'évaluation des performances de la Supply Chain", Thèse de Doctorat, Université Blaise Pascal, Clermont-Ferrand. 2006.

[6] Fenies P , Gourgand S., Rodier S., Interoperable and Multi-flow Software Environment: Application to Health Care Supply Chain, Lecture Notes in Computer Sciences (Springer), Vol 4103; p. 311-322, 2006.

[7] Fenies P, ''Une approche pour la prise en compte de la rationalité limitée des acteurs dans l'aide à la décision : mise en oeuvre en contexte de logistique hospitalière'', revue Management et Avenir, Vol 48, Octobre 2011.

[8] Gartner, Application Development and Maintenance Research Note M-16-8153, The BPA Market Cathes another Major Updraft. Available from: http://www.gartner.com, . 2002.

[9] Green, P., Roseman, M.: ''Integrated Process Modelling: an ontological evaluation''. In Information systems, Vol. 25, 2000.

[10] Giard V., Midler C., ''Pilotages de projet et entreprises'', Paris, Economica, 1993.

[11] Gupta A., ''A four-faceted Knowledge Based Approach to surmounting borders'', Jounal of knowledge management, Vol 5, Issue 4, p. 291-299, 2001.

[12] Hammer M., Champy J., ''Reengineering the Corporation : A Manifest for Business Revolution'', 1st ed., Harper Business, New York, 1993.

[13] Holland C.P., Shaw D.R., Kawalek P., '' BP's multi-enterprise asset management system'', Information and Software Technology, Vol. 47, p. 999-1007, 2005.

[14] Hult, G.T.M., Ketchen Jr., D.J., Slater, S.F, '' Information processing, knowledge development, and strategic supply chain performance''. Academy of Management Journal vol 47 (2), p.241-253, 2004

[15] Lorino P. ''Comptes et récits de la performance, Essai sur le pilotage de l'entreprise'', Les éditions d'organisations, 1995.

[16] Pascal C , ''Gérer les processus à l'hôpital''. Thèse de Doctorat, Université Jean Moulin, Lyon 3, 2000.

[17] Raghu T.Z, Vinze R., A business process context for Knowledge Management., Decision Support Systems, 2005.

[18] Rodier S. . Une Tentative d'Unification et de Résolution des Problèmes de Modélisation et d'Optimisation dans les Systèmes Hospitaliers : Application au NHE. Thèse de Doctorat, Université Blaise Pascal, Clermont-Ferrand,2010.

[19] Sarramia D., ''ASCI-mi : une méthodologie de modélisation multiple et incrémentielle : application aux systèmes de traffic urbain''., Thèse de doctorat, LIMOS, Université Blaise Pascal,2002.

[20] Sghaier W., E. Hergon, J-c Bocquet and A.Desroches . Méthode de (re)conception des processus pour une performance multicritère : application à l'établissement français du sang. In TCB, 16, 505–511, 2009.

[21] Teil A. . Défi de la performance et vision partagée des acteurs : Application à la gestion hospitalière. Thèse de Doctorat, Université Jean Moulin, Lyon 3, 2002.

[22] Toma O., L.Vassalo . Le développement durable et solidaire en santé. Les études hospitalières, Bordeaux, 2010.

[23] Van der Aalst W.M.P., Hofstede A.H.M Weske M., , Business process management: A survey in: W.M.P. Proceedings of BPM 2003, Lecture Notes in Computer Science, Volume 2678, p. 1-12, 2003.

[24] Wang M., Wang H., From process logic to business logic—A cognitive approach to business process management, Information & Management, Volume 43, p. 179-193, 2006.

[25] Weske M.,. Van der Aalst W.M.P ., Verbeek H.M.W., Advances in business process management, Data & Knowledge Engineering Volume 50, p.1-8, 2004.

General framework for transforming continuous metaheuristic in discrete or binary

Broderick Crawford, Ricardo Soto,Gino Astorga and José García

Pontificia Universidad Católica de Valparaíso, 2362807 Valparaíso, Chile.
broderick.crawford@pucv.cl , ricardo.soto@pucv.cl, gino.astorga@uv.cl,
joseantonio.garcia@telefonica.com

1 Introduction

There are many optimization problems with discrete binary search spaces. They need binary algorithms for their solution. Most of the well-known metaheuristics operate in continuous search spaces. This current continuous algorithms must be adapted to solve a binary problem. Examples of adaptation we can found in evolutionary algorithms such as Particle Swarm , Magnetic Optimization Algorithm , Gravitational Search Algorithm, Firefly Algorithm, Shuffled Frog Leaping Algorithms, Fruit Fly Optimization Algorithm, Cuckoo Search Algorithm, Cat Swarm Optimization Algorithm and Bat Algorithm. There are different methods to develop the binary version of a continuous metaheuristic while preserving the concepts of the search process.This work in progress, has as proposal compile, group and explain the main existing techniques for discretization and binarization of continuous metaheuristics with the propose of develop a general framework with the intention of automate the selection of the different binarization methods.

2 Discretization of continuous metaheuristics:

There are many problems that require discrete search spaces. In the investigation of these methods, we found a lot of names. However can be classified in two main groups. Rounding off techniques and Priority position techniques.

Rounding off: This approach is one of the most commonly used for the management of discrete variables, due to its simplicity and low computational cost. It is based on the strategy of rounding to the near integer. The metaheuristic operators are used without modifications and exist two strategies to implement the discretization. The first one, applies a rounding-off near integer operation to feasible solution in every iteration. In the second approach, it is applied at the end of the optimization process. There are multiple problems that use this method, for example optimization of transport aircraft wing [11], task assignment problem [8]. The main metaheuristics that use round-off method are Ant colony , PSO optimization, Firefly . Its disadvantages include the possibility that the solution is in a non-viable region. For other hand, the value of the fitness in rounded point can be very different from the original point.

Priority position: The first methods that used priority position is random-key encoding scheme. Let start with a solution X of n dimension, in each position is assigned a random number in [0,1]. To decode the position from the real random-key space in a discrete space, the position are visited in ascending order generating a $Z \in \mathbb{Z}^n$ discrete solution . This method has been used with Gravitational search algorithm by Chen [2], resolving travelling salesman problem and scheduling problem. The result of gravitational algorithm map as a random key, where small values map to the top position. The same method, but call Small Value Position (SVP) was used first time by Tasgetiren to resolve Single machine total weighted tardiness problem using a PSO algorithm. Later Yousif [14] used SVP, in Scheduling jobs on grid computing. Great value position (GVP) method, use in his first step a priority position discretization.

3 Binarization of continuous metaheuristics

In our study and conceptualization of binarization techniques, we found two main class of binarization methods. The first group of techniques we call Two steps binarization. These techniques,

let to work with the continuous metaheuristic without operators modifications and include a two steps after the continuous iteration. Examples of this techniques are Great Value Position, Transfer functions and angle modulations. The second technique is called continuous-binary operators transformation, redefines the algebra of the search space, therefore reformulate the operators. Examples of this metaheuristics are Set based approach, Binary approach, Binary based on estimation of distribution and Quantum based approach.

Two steps binarization methods: This method working with the continuous operators without modifications. To do the binarization, appeal a two additional steps. The First step, corresponds to introduce operators that put solutions in a inter-space. For example $\mathbb{R}^n \to \mathbb{Z}^n$, $\mathbb{R}^n \to [0,1]^n$ or $\mathbb{R}^n \to \{\text{fuction space}\}$. The Second step transforms from the inter-space into a binary space.

Transfer Function First Step: The transfer function is the most used binarization method. It was introduced by [6]. The transfer function is a very cheap operator, his range provides probabilities values and try to model the transition of the particles position. This function is responsible of the first step of the binarization method and mapping the $\mathbb{R}^n$ solutions in $[0,1]^n$ solutions.In the research have been used two types of functions the S-shaped [13] and V-shaped [4].

Second Step Binarization: The Second Step is the Binarization technique, the vector $\vec{I}$ is transform in a binary solution $B = (b_1, .., b_n)$, applying a binarization rule like standard, complement, roulette, static probability, elitist.

Angle Modulation First Step: This method, used a trigonometric function who has four parameters, this parameters control the frequency and shift of the trigonometric function.

$$g_i(x_j) = \sin(2\pi(x_j - a_i) * b_i * \cos(2\pi(x_j - a_i) * c_i)) + d_i \tag{1}$$

In binary heuristic optimization, this method was first applied in PSO, using a set of benchmark functions. Let a binary problem of n-dimension, and $X = (x_1, x_2, ...x_n)$ a solution. We start with a four dimensional search space, each dimension represent a coefficient of the equation (1).

Second Step Rule: Then every solution (a_i, b_i, c_i, d_i) in this space , get a g_i trigonometric function. For each component single element x_j apply the rule 2:

$$b_{ij} = \begin{cases} 1 \text{ if } g_i(x_j) \geq 0 \\ 0 \quad \text{otherwise} \end{cases} \tag{2}$$

Then for each initial 4-dimension solution (a_i, b_i, c_i, d_i), we get a binary n-dimension solution $(b_{i1}, b_{i2}, ..., b_{in})$ who is a feasible solution of our n-binary problem. The Angle modulate technique, has been apply to network reconfiguration problems [7] using a binary PSO method, multi-user detection technique [10] using a binary adaptive evolution algorithm and the antenna position problem using a angle modulate binary bat algorithm [15].

Continuous-binary operators transformation: These methods are characterized by re-defined the operators of the metaheuristic.There are two main groups. The first one, the algebraic operations are re-written, examples are set approach and boolean approach. The second group, the operators are restructured in terms of select promising regions in the search spaces. This selection is using a probability vector. Examples of this group are Quantum-based binary approach and Binary method based on estimation of distribution. In this section, we describe Quantum and boolean approach.

Quantum binary approach: This approach has been developed in PSO optimization [9]. It has been inspired in uncertainly principle. The individual particles, the PSO algorithm work in a different fashion and we need to rewrite the operators. In the quantum approach, each feasible solution have a position $X = (x_1, x_2, .., x_n)$ and a quantum vector $Q = [Q_1, Q_2, ..., Q_n]$. Q_j. Q represents the probability of x_j take the value 1. For each dimension j, a random number between [0,1] is generated and compared with Q_j, if $rand < Q_j$, then $x_j = 1$, else $x_j = 0$. After that, the new P_{best} and P_{global}, are calculated using the objective function. Finally we update the transition probability using of the equations $Q_{self}(t) = \alpha P_{best}(t) + \beta(1 - P_{best})$ and $Q_{social}(t) = \alpha P_{global}(t) + \beta(1 - P_{global})$ to calculate the best local and global. Finally we used 3. to calculate the new probability vector

$$Q(t+1) = C_1 Q(t) + C_2 Q_{self}(t) + C_3 Q_{global}(t) \tag{3}$$

The Quantum method has been apply to a Swarm optimization algorithm, in combinatorial optimization [12]. Recently this approach has been used to solve the set covering problem [3]

Boolean Approach: This method let transform the real operators in binary operators. This transformation is performed using boolean operations. The operators act over the binary solutions. This approach was born as binarization technique of particle Swarm Optimization [1]. The velocity and position boolean equation with inertia weight are presented in 4 and 5. $P_{best,i}$ and P_{global} belong to best position of select particle and position to global best solution respectively, c_1 and c_2 are random vectors. $V_i(t)$ corresponds to velocity at time t.

$$V_i(t+1) = w \oplus V_i(t) \odot c_1 \otimes (P_{best,i} \oplus X_i) \odot c_2 \otimes (P_{global} \oplus X_i) \tag{4}$$

$$X_i(t+1) = X_i(t) \oplus V_i(t+1) \tag{5}$$

This method has been applied to different binary optimization problems, using particle swarm method [5]. Boolean approach introduce a efficient velocity bounding strategy based in thymic negative selection of the T-cells.

References

1. F. Afshinmanesh, A. Marandi, and A. Rahimi-Kian. A novel binary particle swarm optimization method using artificial immune system. *Computer as a Tool*, 1:217–220, 2005.
2. H. Chen, S. Li, and T. Zheng. Hybrid gravitational search algorithm with random-key encoding scheme combined with simulated annealing. *International journal of computer science and network security*, 11(6):208–217, 2011.
3. B. Crawford, R. Soto, N. Berríos, F. Johnson, F. Paredes, C. Castro, and E. Norero. A binary cat swarm optimization algorithm for the non-unicost set covering problem. *Mathematical Problems in Engineering*, 2015, 2015.
4. B. Crawford, R. Soto, M. Olivares-Suarez, W. Palma, F. Paredes, E. Olguin, and E. Norero. A binary coded firefly algorithm that solves the set covering problem. *Romanian Journal of Information Science and Technology*, 17(3):252–264, 2014.
5. K. Deligkaris, Z. Zaharis, and D. Kampitaki. Thinned planar array design using boolean pso with velocity mutation. *IEEE Transactions on Magnetics*, 45(3):1490–1493, 2009.
6. J. K. J and R. Eberhart. A discrete binary version of the particle swarm algorithm. *IEEE*, 4105:4104–4108, 1997.
7. W. Liu, L. Liu, and D. Cartes. Angle modulated particle swarm optimization based defensive islanding of large scale power systems. *IEEE Power Engineering Society Conference and Exposition in Africa*, pages 1–8, 2007.
8. A. Salman, I. Ahmad, and S. Al-Madani. Particle swarm optimization for task assignment problem. *Microprocess Microsyst*, 26(8):363–371, 2002.
9. J. Sun, B. Feng, and W. Xu. Particle swarm optimization with particles having quantum behavior. *IEEE Congress on Evolutionary Computation*, pages 325–331, 2004.
10. D. Swagatam, M. Rohan, and K. Rupam. Multi-user detection in multi-carrier cdma wireless broadband system using a binary adaptive differential evolution algorithm. *Proceedings of the 15th annual conference on Genetic and evolutionary computation, GECCO*, pages 1245–1252, 2013.
11. G. Venter and J. Sobieszczanski-Sobieski. Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization. *Struct Multidisc Optim*, 26(1):121–131, 2004.
12. I. Wang, Y. Zhang, and Y. Zhou. Discrete quantum-behaved particle swarm optimization based on estimation of distribution for combinatorial optimization. *IEEE Evolutionary Computation*, pages 897–904, 2008.
13. Y. Yang, Y. Mao, P. Yang, and Y. Jiang. The unit commitment problem based on an improved firefly and particle swarm optimization hybrid algorithm. In *Chinese Automation Congress (CAC), 2013*, pages 718–722. IEEE, 2013.
14. N. S. Yousif A, Abdullah A. Scheduling jobs on grid computing using firefly algorithm. *Journal of Theoretical and Applied Information Technology*, 33(2):155–164, 2011.
15. D. Zakaria and D. Chaker. Binary bat algorithm: On the efficiency of mapping functions when handling binary problems using continuous-variable-based metaheuristics. *Computer Science and Its Applications*, 456:3–14, 2015.

Metaheuristics and machine learning

A Multi-objective Evolutionary Algorithm with Adaptative Selection of Operators

A. Santiago[1,4], B. Dorronsoro[1], A. J. Nebro[2], Juan J. Durillo[3], and H. Fraire[4]

[1] University of Cádiz, Spain
alejandro.santiagopi@mail.uca.es
bernabe.dorronsoro@uca.es
[2] University of Málaga, Spain
antonio@lcc.uma.es
[3] University of Innsbruck, Austria
juan@dps.uibk.ac.at
[4] Technological Institute of Ciudad Madero, Mexico
automatas2002@yahoo.com.mx

1 Introduction

Multi-objective optimization is the process of finding the optimum of several conflicting functions or objectives [2]. Its important popularity in the optimization domain is justified by the many real-world problems that have a multi-objective nature. Therefore, there is a plethora of algorithms, particularly metaheuristics, that have been proposed for such problems. It is known that the accuracy of metaheuristics greatly depends on the operators they use, which offer different performance depending on the characteristics of the problem being solved [3].

In this work, we propose a new algorithm called Multi-objective Evolutionary Algorithm with Adaptive Selection of Operators (MEASO). This technique is featured by implementing a number of variation operators that are automatically chosen, based on a fuzzy logic engine, to provide the best benefit. This way, the accuracy of the proposed algorithm is not limited to the performance of a given operator for the considered problem, making the algorithm more reliable for a wider variety of problems, with respect to other techniques with fixed variation operators. MOEASO is briefly presented next.

2 The MEASO Algorithm

MEASO is a steady-state genetic algorithm that implements four different variation operators: SBX, differential evolution, polynomial mutation and uniform mutation. In every iteration, the algorithm chooses the operator to apply according to their success in the last generations, based on a fuzzy logic engine with triangular shaped membership functions. The success means that the solution generated is an environmental survivor. The fuzzy engine makes use of 2 inputs of information: the frequency of use of the operator in the last time window, and the number of non successful offspring generated (as a measure of stagnation); and outputs the probability of use for every operator.

Another distinguishing feature of MEASO is that it works on an archive of solutions, and not on a population. This means that the size of the set of solutions it evolves is variable (limited to a maximum of 100), and when a new solution is inserted in the archive, the dominated solutions are removed from it. Working directly on the archive of solutions allows a fast convergence of the Pareto front approximation, while the induced diversity loss is overcome thanks to the adaptive selection of variation operators.

3 Results

Table 1 presents the results obtained by our proposed algorithm, compared versus two state-of-the-art metaheuristics, namely MOEA/D-DE [4] and SMS-EMOA [1]. The compared algorithms are configured with standard settings. The results shown in the table are the median and interquartile range of the Pareto front approximations obtained by the algorithms after 30 independent

runs, measured by the unary additive epsilon ($I^1_{\epsilon+}$) and hypervolume (HV) performance metrics, for which lower and higher values are better, respectively. The best results are emphasized with dark grey background, while the light grey background is used for the second best results. We also performed the rank-sum test Wilcoxon test to look for significant difference (at 95% confidence level) between MEASO and the compared algorithms. Therefore, a '▼' symbol means that the algorithm is significantly worse than MEASO for that problem, while '▲' is used when the compared algorithm outperforms MEASO. Otherwise, there is no statistical difference between the algorithms. For this comparison study, we selected two complicated benchmarks: the one proposed for CEC2009 competition on performance assessment of multi-objective optimization algorithms (UF1 to UF10), and the recently proposed GLT problems. Problems UF8 to UF10, as well as GLT5 are composed of three objective functions, while the others have two.

Table 1. Results of the algorithms, presented as median and interquartile range of the two considered performance metrics after 30 independent runs. Dark and light grey background cells indicate, respectively, the best and second best results.

	Unary additive epsilon ($I^1_{\epsilon+}$)			Hypervolume (HV)		
	MEASO	MOEA/D-DE	SMS-EMOA	MEASO	MOEA/D-DE	SMS-EMOA
UF1	$1.63e-02_{5.1e-03}$	$1.40e-02_{2.6e-03}$▲	$1.76e-01_{1.8e-02}$▼	$6.55e-01_{1.7e-03}$	$6.56e-01_{6.4e-04}$▲	$5.44e-01_{8.5e-03}$▼
UF2	$5.09e-02_{2.9e-02}$	$6.93e-02_{1.9e-02}$▼	$1.16e-01_{3.6e-02}$▼	$6.51e-01_{4.8e-03}$	$6.46e-01_{4.3e-03}$▼	$6.33e-01_{9.1e-03}$▼
UF3	$1.08e-01_{1.3e-01}$	$4.79e-02_{6.0e-02}$▲	$4.63e-01_{9.5e-02}$▼	$5.88e-01_{6.5e-02}$	$6.42e-01_{3.1e-02}$▲	$3.79e-01_{5.8e-02}$▼
UF4	$4.50e-02_{5.1e-03}$	$8.45e-02_{7.8e-03}$▼	$3.95e-02_{5.5e-03}$▲	$2.70e-01_{3.4e-03}$	$2.32e-01_{1.5e-02}$▼	$2.76e-01_{9.7e-04}$▲
UF5	$5.26e-01_{2.0e-01}$	$6.50e-01_{4.6e-01}$▼	$4.26e-01_{2.2e-01}$	$1.82e-01_{1.4e-01}$	$1.16e-01_{2.3e-01}$	$1.81e-01_{8.0e-02}$
UF6	$5.43e-01_{3.1e-01}$	$6.42e-01_{4.1e-01}$	$4.32e-01_{1.6e-01}$▲	$2.39e-01_{1.5e-01}$	$2.15e-01_{1.1e-01}$	$2.43e-01_{8.1e-02}$
UF7	$4.61e-02_{7.7e-02}$	$4.13e-02_{2.1e-02}$	$5.66e-01_{5.3e-01}$▼	$4.87e-01_{4.6e-03}$	$4.86e-01_{4.6e-03}$	$3.08e-01_{2.0e-01}$▼
UF8	$2.93e-01_{2.5e-01}$	$7.08e-01_{8.3e-04}$▼	$7.08e-01_{4.5e-01}$	$1.63e-01_{7.5e-02}$	$1.55e-01_{1.7e-02}$	$2.17e-01_{1.0e-01}$▲
UF9	$5.81e-01_{5.2e-02}$	$4.47e-01_{8.8e-02}$▲	$4.29e-01_{3.0e-01}$▲	$1.70e-01_{1.0e-01}$	$4.32e-01_{2.0e-01}$▲	$5.56e-01_{1.0e-01}$▲
UF10	$8.72e-01_{1.2e-01}$	$1.04e+00_{1.6e-01}$▼	$8.43e-01_{2.2e-01}$	$2.11e-02_{3.8e-02}$	$0.00e+00_{1.6e-02}$▼	$7.04e-02_{1.6e-01}$▲
GLT1	$7.73e-03_{1.3e-02}$	$9.45e-03_{1.3e-03}$	$2.42e-01_{8.4e-03}$▼	$3.68e-01_{8.2e-03}$	$3.72e-01_{1.6e-04}$▲	$3.16e-02_{6.4e-02}$▼
GLT2	$9.86e-03_{2.0e-03}$	$7.32e-02_{7.9e-03}$▼	$2.13e-01_{1.7e-01}$▼	$7.79e-01_{5.0e-04}$	$7.53e-01_{5.8e-03}$▼	$6.34e-01_{1.4e-01}$▼
GLT3	$2.79e-02_{1.1e-02}$	$1.66e-02_{8.3e-03}$▲	$4.73e-02_{1.5e-02}$▼	$9.41e-01_{5.2e-03}$	$9.44e-01_{1.4e-03}$▲	$9.26e-01_{9.3e-03}$▼
GLT4	$1.30e-01_{4.0e-01}$	$1.27e-02_{1.6e-03}$	$1.86e-01_{2.8e-01}$▼	$4.56e-01_{2.4e-01}$	$4.94e-01_{4.8e-04}$	$3.33e-01_{3.0e-01}$▼
GLT5	$4.89e-02_{5.2e-03}$	$8.38e-02_{2.2e-03}$▼	$7.13e-02_{5.4e-02}$▼	$9.63e-01_{7.0e-04}$	$9.22e-01_{2.5e-03}$▼	$9.63e-01_{1.2e-02}$
GLT6	$4.34e-02_{2.8e-02}$	$1.13e-01_{3.4e-02}$▼	$5.21e-02_{1.3e-02}$▼	$9.59e-01_{1.3e-03}$	$9.17e-01_{7.5e-03}$▼	$9.58e-01_{4.0e-03}$

We can see in Table 1 how MEASO finds the best, or the second best, solution for all problems, according to the two studied metrics. The only exception is problem UF9, one of the 3D problems. In addition, we found statistical significance in most of the differences between MEASO and the other two algorithms. We performed the Friedmann rank of the algorithms for the two metrics and MEASO resulted the overall best algorithm in the two cases, followed by MOEA/D-DE (see Table 2).

Table 2. Average ranking of the algorithms

$I^1_{\epsilon+}$		HV	
Algorithm	Ranking	Algorithm	Ranking
MEASO	1.75	MEASO	1.9375
MOEA/D-DE	2.0625	MOEA/D-DE	2.0
SMS-EMOA	2.1875	SMS-EMOA	2.0625

In order to further analyze the results of the two best compared algorithms, we show in Fig. 1 their Pareto front approximations with the best HV value (out of the 30 independent runs made) for two sample problems, namely GLT2 and GLT4. The true Pareto front is also represented for comparison purposes (called the reference front in the pictures). It can be seen how our MEASO algorithm finds accurate approximations with a better distribution of the solutions, without any important gap, confirming the results of the performance metrics used. On the contrary, the Pareto front approximations of MOEA/D-DE are more populated with solutions in the right part of the front, with large gaps not covering the true Pareto front on the leftmost side.

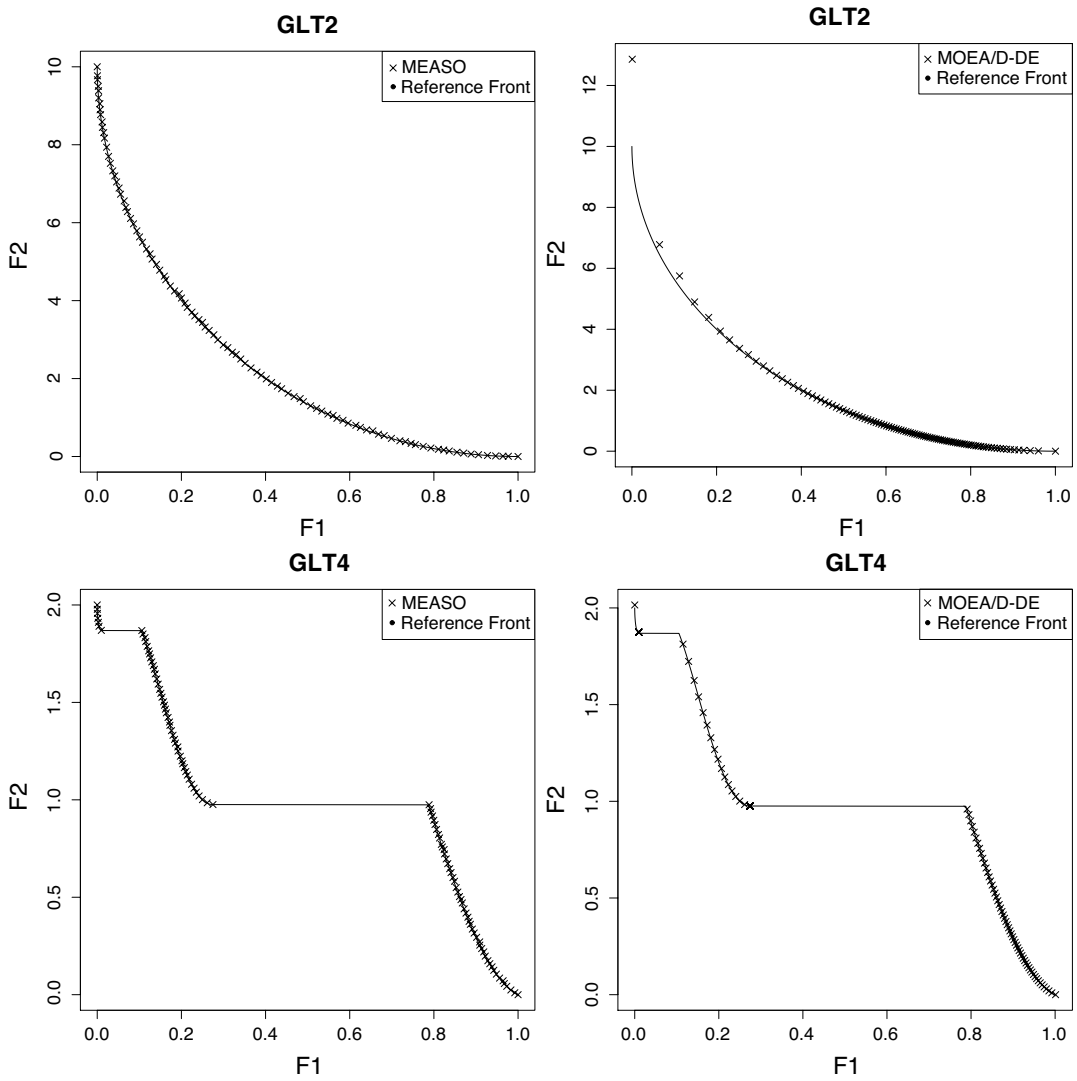


Fig. 1. Fronts with the best Hypervolume values on problems GLT2 and GLT4 obtained by MEASO (left) and MOEA/D-DE (right).

Acknowledgments

A. Santiago would like to thank CONACyT Mexico, for the support no. 360199. B. Dorronsoro would like to acknowledge the Spanish MINECO for the support provided under contracts TIN2014-60844-R (the SAVANT project) and RYC-2013-13355 (Ramón y Cajal program). A.J. Nebro acknowledges support of grants TIN2011-25840 (Ministerio de Ciencia e Innovación) and P11-TIC-7529 and P12-TIC-1519 (Plan Andaluz de Investigación, Desarrollo e Innovación).

References

1. Beume, N., Naujoks, B., and Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3):1653–1669 (2007)
2. Coello Coello, C., Lamont, G. B., and van Veldhuizen, D. A.: Evolutionary Algorithms for Solving Multi-Objective Problems. 2nd ed. Springer-Verlag (2007)
3. Deb, K., Sinha, A., and Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. In GECCO06, pp. 1141–1148 (2006)
4. Zhang, Q., and Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 11(6):712–731 (2007)

Intrusion Detection System based on a behavioral approach

Mehdi Moukhafi[1], Seddik Bri[2], Khalid El Yassini [1]

1.Computer and Applications Laboratory, Department of Mathematics and Computer Sciences, Moulay Ismaïl University, Meknes, Morocco
mehdi.moukhafi@gmail.com, .khalid.elyassini @gmail.com

2.Materials and Instrumentations (MIN), Department of Electric Engineering, ESTM, Moulay Ismail University, Meknes, Morocco
briseddik@gmail.com

Abstract

The emergence of computer networks as very powerful work tools, offering various and accessible services by a very big number of users, led to the needs of more and more numerous. The needs of users are very varied, relating to types of manipulated data and to the mechanisms for the management of the services offered. Furthermore, the convergence of new communication technologies to compatible platforms and reusable has generated unprecedented threats on information systems. Also rising from the dynamics of the architectures, these threats can cause significant damage and sometimes irreparable. In this context, network security has become a priority for the development of communications, for this reason, the different mechanisms are put in place to ensure the security of information systems, such as the authentication mechanisms, cryptography, antivirus, firewall and the Intrusion Detection Systems (IDS). In this article, we present a detailed study on attack detection approaches focus on putting on the detection based on a behavioral approach by providing the most extensive existing solutions in this area. Finally, the methodology and testing tools of IDS are illustrated.

Keywords: Network security, Intrusion Detection System (IDS), Program behavior, Soft computing, Machine Learning

1 Introduction

The Information and Communications Technology (ICT) play a central role in the daily lives of people and in society in general. According to the report of the International Telecommunications Union [1] carried out in 2015, 43% of the world population uses the Internet. Indeed, areas falling under the private life such as the sending of mail or the paying online, but also other strategic areas such as the banking sector or the military communications, are mainly based on our days massively on the ICT. In this respect, the attacks carried out by malicious users to exploit the vulnerabilities of these systems are more and more frequent; especially with the easy access to security test tools which are more accessible to professionals as well as hackers.

Such attacks can for example harm the image of the owner of the information system or cause significant financial damages. The problem of security, therefore, becomes a key issue for both users and administrators of these information systems. It is, therefore, necessary that the actors in the field of information technologies offer adequate protection mechanisms to better secure these technologies. These mechanisms can be: tools for authentication, data encryption, firewalls to filter the connections, anti-virus to detect malware or well of intrusion detection systems (IDS) to detect unauthorized activities.

In this context, Intrusion Detection Systems (IDS) are used to monitor and analyze the events in an information system. Presented for the first time by Anderson in 1980 [2] and formalized later by Denning

[3], the IDS can be used in a global security policy, which includes other tools of protection, such as firewalls and anti-virus; where it is essential to take advantage of the collaboration of these tools and their complementarity. The Intrusion Detection System may exist alone, and in this case it is very important to optimize its processes. Nevertheless, in all cases, it is essential to improve the performance of IDS.

2 Intrusion Detection System

We can define an Intrusion Detection System (IDS) as any tool, methods and resources that help us to predict or identify any unauthorized activity in a network [2]. In the architecture proposed by the group IDWG5 of the IETF6 [4, 5], we can find three modules as the following: Sensors Module, Analyzing Module and Managing Module, as shown in figure 1. In this architecture, the objective was the definition of a standard for communication between the components of the intrusion detection system. This architecture defines a format for the exchange of message for the IDS: Intrusion Detection Message Exchange Format (IDMEF), which implicitly contains a data model.

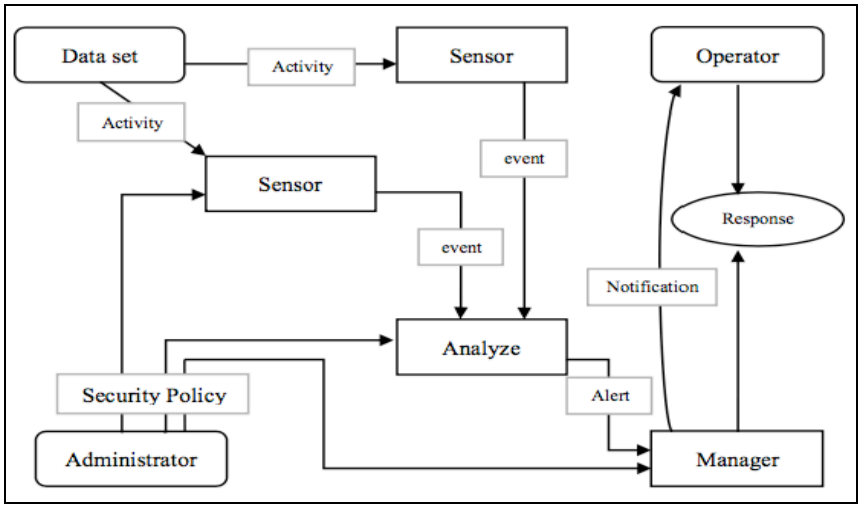


Figure 1: Architecture of an IDS

The intrusion detection features include:

- Monitoring and Analysis the user's activities and system.
- Configurations of system analysis.
- Ability to identify typical patterns of attacks.
- Analysis of the models of abnormal activity.

Intrusion detection systems tend to ensure the five following properties [6]:

- The accuracy of detection: it means a perfect detection of attacks with a minimum risk of false positives.
- Performance: A fast detection of intrusion thorough the analysis of events which is essential to conduct effective detection in real time.
- Completeness: A detection of known and unknown attacks.
- Fault Tolerance: intrusion detection systems must resist attacks as well as their consequences.
- Speed: A fast analysis of the data which allows you to undertake the necessary measures to stop the attack and protect the resources of the network and the intrusion detection system.

3 Type of approaches of IDS

The intrusion detection systems are generally based on two approaches: Behavioral and by signature. The techniques of intrusion detection based on the behavioral approach suppose that the intrusion can be detected by the observation of the behavioral abnormality in relation to the normal behavior or prediction of the system or of the users. By contrast, the signature approach is based on the accumulated knowledge of specific attacks and vulnerabilities in the system (Table1).

Table 1: Comparison of the Two Approaches

Approaches	Advantages	Disadvantages
Behavioral	• Not in need of a base attack. • Detection of Possible Unknown intrusions: few false negatives.	• Top rate of false positive • A malicious user can change slowly his behaviour to familiarize the system of a intrusive behaviour: risk of false negatives.
Signature	Few false positives	• Based of signature which is difficult to be built; especially maintaining: risk of false negatives. • No detection of unknown attacks: risk of false negatives.

3.1 Signature Approach

This approach is based on the accumulated knowledge on attacks denoted and vulnerabilities of systems. The system of detection of Intrusion contains the information about the vulnerabilities then look for any attempt to exploit them. If the IDS detected such attempt, an alarm would react. In other words, any action that is not explicitly known as an attack is considered legitimate. By consequent, the precision of IDS based on the signature approach is good. However, this precision depends always on the up dated knowledge on the attacks which must be regular [6].

3.2 Behavioural Approach

The techniques of intrusion detection based on the behavioural approach [7] assume that the intrusion can be detected by the observation of the deviation from the normal behaviour or expected of the system or of the users. At the beginning, the model of the normal behaviour is extracted from the reference information collected by various means. Then, the intrusion detection system compares this model with the current activity. If a deviation is detected, an alert will be triggered. In a general way, we can say that this approach considers any behaviour that is not previously recorded as intrusion. Therefore, this approach may be complete, but the accuracy remains his greatest concern.

Intrusion Detection behavioural or by anomaly is based on the assumption that an attack causes an abnormal use of resources or manifest a strange behaviour on the part of the user. As a result, the different approaches that have been proposed to learn the normal behaviour to be able to detect any significant deviation. We can classify this work in three categories, namely the statistical approach, the immunological approach and the search of data.

3.2.1 Immunological approach

In this approach, the normal behaviour of a process is characterized by a set of execution traces recorded during the learning phase. These traces contain in an orderly manner the whole system calls made by the process during the entire duration of its execution. During the Discovery phase, the last n system calls made by the process are compared to the whole of sub-sequences of n elements contained in the Traces characterizing the basis of reference of this process. In practice, the detection is done from the list of sequences of system calls to fixed size n statically constructed from traces of reference. Forrest [7] was the first to use the immunological approach to model the process on a machine. Its approach is to describe the normal behaviour or the "self" via a finite sequence of calls systems. The sequences called N-gram are used as a basis for comparing the calls of systems process at a monitoring phase. Wespi, steel and Debar [8] consider a more general case by analysing the examination events. They generate sequences of events of varying sizes to model the normal state of the system. Then a pattern is selected if there are reasons that the follow directly otherwise the score of anomaly is incremented by one and an alarm is triggered when the score exceeds the threshold tolerated.

3.2.2 Data Mining Approach

The techniques of data mining (Data Mining) apply equally to intrusion detection by anomaly for intrusion detection based on knowledge. [9] The normal behavior of the statistical analysis of the system is one of the first approaches to intrusion detection.

- **Statistical approach**

The statistics are used to mathematically describe a mechanism observed. Generally, the observations allow to get a rough description. For this, the value of certain observations is considered random variables. For each of its comments, a statistical model is used to describe the set of distributions of the corresponding random variable.

Denning [3] presents a model in which a profile connects via a random variable subject (user process) to an object (resources) and Thottan and Ji [11] propose a statistical processing technique of the signal based on the exponential change detection to detect anomalies in traffic at the network level.

Sui Ling Song and Li [10] developed a technique based on aggregate flows that significantly reduces the amount of monitoring data and manages high amounts of statistics and packet data. This technique uses the measuring mechanism debit Netflow [12], thus they collect the data stream and select the five key needed to identify a malicious traffic (Source IP address, destination IP address, port number, source, number of destination port, protocol layer three).

- **Approach based on soft computing**

Intrusion detection technique used by Saied and al. [13] to detect DDoS attacks known and unknown environments in real time, based on the use of an artificial neural network algorithm (ANN) [14] based on specific characteristics (models) that differentiates between legitimate traffic and attack traffic using learning by back propagation coupled with a sigmoid activation function, the authors selected three ANN topological structures, one for each of the most used in DDOS attacks protocols (TCP, UDP, ICMP), each with three layers (input, hidden layers and output). The number of nodes in each topological structure is different, the ICMP topological structure consists of three inputs and four hidden nodes, the topology structure TCP consists of five inputs and four hidden nodes and the topological structure of UDP consists of four inputs and three hidden nodes treat the calculation process with respect to input and output nodes. The output layer consists output node for the attacks, and an output node for legitimate traffic.

Monowar et al [15] was put in place a procedure based mutual information and general, a technique of entropy function for selecting a relevant non-redundant subset of features, based on a clustering based trees to generate a set of reference points and a function of aberrant score to classify incoming network traffic to identify anomalies.

An outlier detection technique based on the distance is presented by Knorr et al. [16]. Defines a point to a remote outlier if at least fraction user-defined points in the dataset are further than a certain minimum distance.

Approach based on the distance presented by Ertoz et al. [17] developed an intrusion detection system known as the outlier detection unattended based on distance and prediction. The outlier detection is to find n outliers in a set D, that is, n D objects with the greatest weight. After the step of predicting outliers corresponds to decide whether an input object is an outlier relative to D.

Nadiammai and Hemalatha [18] have established a DOS attack detection hybrid mechanism which is based on step two, the first is the submission of the traffic flow has an intrusion detection system based signature (SNORT) to pretreat the incoming flow and detect known attacks, the second step is the recovery of flows classified as legitimate and applies a classification based learning supervised (SVM) with the use of the kernel function Radial basis function.

Singh et al. developed an OS-ELM system (Online Sequential Extreme Learning Machine) [19] that learns normal and abnormal behavior by analyzing network traffic in various reference data sets. The technique uses alpha-profiling to reduce the time complexity while extraneous functions are ignored by using a set of filter, correlation and selection techniques. Instead of sampling the alpha-profiling is used to reduce the size of the set of training data. The authors OS-ELM is designed to overcome the limitation of slow learning of the neural network. It offers a good generalization performance with a fast learning speed.

Amudhaet al. propose a hybrid classifier model (ABC-SVM) [20] based on Artificial Bee Colony (ABC) and Support Vector Machine (SVM) algorithms to construct an IDS using KDDCup'99 dataset. 10-fold cross validation method was used. The performance results indicated that the hybrid approach achieved better accuracy in classifying the dataset and to distinguish the network traffic as normal and abnormal.

4 Methodology and Tools

4.1 Presentation of the DataSet

Cyber Systems and Technology Group of MIT Lincoln Laboratory [12] simulated LAN US Air Force LAN with multiple attacks and captured nine weeks TCPdump data. This database was first used for competitions kdd99, but since it has become the test database to the IDS based on a behavioral approach, Table 2 shows the fields in the kdd99.

Each connection record consists of approximately 100 bytes. This was converted into approximately 4.9 million connecting vectors each contains 41 fields [21].

Table 2: List of features of KDD99 Dataset

Feature no.	Feature name	Feature no.	Feature name	Feature no.	Feature name
1	Duration	15	Su attempted	29	Same srv rate
2	Protocol type	16	Num root	30	Diff srv rate
3	Service	17	Num file creations	31	Srv diff host rate
4	Flag	18	Num shells	32	Dst host count
5	Source bytes	19	Num access files	33	Dst host srv count

6	Destination bytes	20	Num outbound cmds	34	Dst host same srv rate
7	Land	21	Is host login	35	Dst host diff srv rate
8	Wrong fragment	22	Is guest login	36	Dst host same src port rate
9	Urgent	23	count	37	Dst host srv diff host rate
10	Hot	24	Srv count	38	Dst host serror rate
11	Number failed logins	25	Serror rate	39	Dst host srvserror rate
12	Logged in	26	Srvserror rate	40	Dst host rerror rate
13	Num compromised	27	Rerror rate	41	Dst host srvrerror rate
14	Root shell	28	Srvrerror rate	42	Class label

This database is collected by simulating different on different platforms such as Windows, Unix. Four gigabytes of raw data compressed TCP dump is transformed into five million connections files.

The attack classes presented in the KDD99 data set are grouped into four categories [23] (Table3):

1. DOS: Denial of service is an attack category, which depletes the victims resources thereby making it unable to handle legitimate requests – e.g. syn flooding. Relevant features: “source bytes” and “percentage of packets with errors”.
2. Probing: Surveillance and other probing attacks objective is to gain information about the remote victim e.g. port scanning. Relevant features: “duration of connection” and “source bytes”.
3. U2R: unauthorized access to local super user (root) privileges is an attack type, by which an attacker uses a normal account to login into a victim system and tries to gain root/administrator privileges by exploiting some vulnerability in the victim e.g. buffer overflow attacks. Relevant features: “number of file creations” and “number of shell prompts invoked”.
4. R2L: unauthorized access from a remote machine, the attacker intrudes into a remote machine and gains local access of the victim machine. E.g. password guessing Relevant features: Network level features – “duration of connection” and “service requested” and host level features - “number of failed login attempts”.

Table 3: List of attacks present in KDD99 Dataset

Attack group	Attacks
Probe	ipsweep, mscan, nmap, portsweep, saint, satan
DoS	apache2, back, land, mailbomb, Neptune, processtable, pod, udpstorm, smurf, teardrop
U2R	buffer_overflow, httptuneel, loadmodule, perl, rootkit, xterm, ps, sqlattack
R2L	ftp_write, imap, guess_passwd, named, multihop, phf, sendmail, snmpgetattack, snmpguess, spy, warezclient, worm, warezmaster, zsnoop, xlock

Example of connection:

0,TCP,SMTP,Sf,946,494,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,64,243,0.94,0.05,0.02,0.02,0.00,0.00,0.00,0.00,normal.

25,TCP,telnet,Sf,269,2333,0,0,0,0,0,1,0,1,0,2,2,1,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,69,2,0.03,0.06,0.01,0.00,0.00,0.00,0.00,0.00,smurf.

4.2 Test Metric IDS based on a behavioral approach

For a binary classifier confusion matrix (Table 4) shows the results of the real class compared to the predicted result. In these experiments, normal connections are positive events while abnormal represent negative events. In Table 4, True Positive (TP) is the number of real normal logon events that were correctly classified as normal. false positive (FP) is the number of abnormal connections of events that have been incorrectly classified as normal connections. False negative (FN) is the number of normal connections of events that have been incorrectly classified as abnormal connections. True negative (TN) is the number of abnormal connections of events that were correctly classified as abnormal.

Table 4: Confusion matrix

Predicted class	Actual class	
	Normal	Anomalous
Normal	TP	FP
Anomalous	FN	TN

The parameters evaluated are: precision, cadence, TP, TN rate, the rate FP, rate Fn, the accuracy and the time to test. The duration of the test represents the time actual CPU taken by IDS to classify all the connections of test. The values of the other parameters are calculated as defined by the equations (1) - (6) :

- $Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$ (1)
- $TP\ rate = \frac{(TP)}{(TP+TN)}$ (2)
- $TN\ rate = \frac{(TN)}{(FP+TN)}$ (3)
- $FP\ rate = \frac{(FP)}{(FP+TN)}$ (4)
- $FN\ rate = \frac{(FN)}{(FN+TP)}$ (5)
- $Precision = \frac{(TP)}{(TP+Fp)}$ (6)

5 Conclusion

The entire network traffic data is huge and unbalanced. Distributing connections to certain protocols is higher than others. An intrusion detection system (IDS) today has difficulty treating this huge dataset and traditional intrusion detection techniques consume a lot of resources and do not detect unknown attacks. In this context, researchers are opting more and more for the use of data mining techniques. However, despite their many advantages, these techniques also suffer from low accuracy and false positive detection rate high. In this article, we studied a variety of techniques used by IDS. Our work is focused on intrusion detection based on behavioral approach. We are interested now an intrusion detection algorithm based on artificial neural networks and also on the pre-treatment phase and selection of input data training. We noticed to boost the detection results and the performance of the system, the selection of learning values and pre-treatment significantly increases the detection rate.

References

[1] Rapport de l'Union internationale des télécommunications (UIT) (2016). Le Monde en 2015 : faits et chiffres sur l'information, la communication et les technologies. www.itu.int/fr/pages/default.aspx

[2] JP. Anderson. (1980). Computer Security Threat Monitoring and Surveillance. Technical report

[3] DE. Denning (1987). An Intrusion-Detection Model. In IEEE Transactions on Software Engineering. 13, 118.

[4] D. Curry and H. Debar (2006). Intrusion detection message exchange format data model and extensible markup language (xml) document type definition. https://tools.ietf.org/html/draft-ietf-idwg-idmef-xml-16.

[5] Intrusion Detection Message Exchange Format (IDMEF), www.ietf.org/rfc/rfc4765.txt

[6] H. Debar, M. Dacier, A. Wespi (2000), A Revised Taxonomy for Intrusion-Detection Systems, Annales des Télécommunications. 55(7-8), 361-378.

[7] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff (1996). A sense of self for Unix processes. In Proceedings of the 1996 IEEE Symposium on Security and Privacy. IEEE Computer Society, 120.

[8] A. Wespi, M. Dacier, and H. Debar (2000). Intrusion detection using variable-length audit trail patterns. In Proceedings of the 3rd International Workshop on the Recent Advances in Intrusion Detection,1907,110-129.

[9] W. Lee, S.J. Stolfo and K.W. Mok (1999). A data mining framework for building intrusion detection models, in: Proceedings of IEEE Symposium on Security and Privacy.120–132.

[10] S. Song and L. Ling (2006). Flow-based Statistical Aggregation Schemes for Network Anomaly Detection, IEEE International Conference on Networking, Sensing and Control. 786 – 791.

[11] M. Thottan and C. Ji (2003). Anomaly detection in IP networks, IEEE Trans. Signal Process. 51 (8), 2191–2204.

[12] CSICO Company, Cisco IOS NetFlow http://www.cisco.com/warp/public/732/Tech/nmp/netflow/

[13] A. Saied, R. E. Overill and T. Radzik (2016).Detection of known and unknown DDoS attacks using Artificial Neural Networks. 172, 385–393.

[14] T.M. Mitchell (1997). *Machine Learning*. 1st ed. McGraw- Hill Science/Engineering/Math, New York, United State, 81–117.

[15] M. H. Bhuyana , D.K. Bhattacharyya and J.K. Kalitac(2016). A multi-step outlier-based anomaly detection approach to network-wide traffic, Information Sciences. 348, 243–271.

[16] E.M. Knorr, R.T. Ng and V. Tucakov (2000). Distance-based outliers: algorithms and applications, VLDB Journal, 8 (3–4), 237–253.

[17] L. Ertoz, E. Eilertson, A. Lazarevic, P. Tan, J. Srivastava, V. Kumar and P. Dokas(2004). *Minds-minnesota intrusion detection system*. NextGeneration Data Mining, 199-218.

[18] G.V. Nadiammai , M. Hemalatha(2014), Effective approach toward Intrusion Detection System using data mining techniques, Egyptian Informatics Journal, 15 (1),37–50.

[19] Raman Singha , Harish Kumarb, R.K. Singlac (2015). An intrusion detection system using network traffic profiling and online sequential extreme learning machine, Expert Systems with Applications, 42 (22), 8609–8624.

[20] P. Amudha, S. Karthik and S. Sivakumari (2016). An Experimental Analysis of Hybrid Classication Approach for Intrusion Detection, Indian Journal of Science and Technology, 9 (13), 1-8.

[21] Data set kdd99. http://kdd.ics.uci.edu/databases/kddcup99

[22] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani (2009). A Detailed Analysis of the KDD CUP 99 Data Set, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications, 1 – 6.

A Probabilistic Finite State Machine Design of Particle Swarm Optimisation

Malek Sarhani[1] , Oussama Aoun[1] , and Abdellatif El Afia[1]

1.ENSIAS - Institute of computer science, Mohamed V University, Rabat, Morocco.
malek.sarhani@um5s.net.ma
oussama.aoun@um5s.net.ma
a.elafia@um5s.net.ma

Abstract Nowadays, control is the main concern with emergent behaviours of multi-agent systems and state machine reasoning. This paper focuses on the restriction of this general issue to swarm intelligence approaches designed for solving complex optimization problems. Indeed, we propose a probabilistic finite state machine for controlling particles behaviour of the particle swarm optimization algorithm. That is, our multi-agent approach consists of assigning different roles to each particle based on its probabilistic finite state machine control which is used to address this issue. We performed evaluations on ten benchmark functions to test our control scheme for particles. Experimental results show that our proposed scheme gives a distinguishable outperformance on a number of state of the art of PSO variants.

Keywords: particle swarm optimization, probabilistic finite state machine, multi-agent learning, population control, hidden markov model.

1 Introduction

Decentralized approaches have become promising to solve complex problems, especially in artificial intelligence. Among such approaches fall into the area multi-agent systems in which a number of agents have to solve together cooperatively problems. In particular, various efforts have been made to integrate multi-agent approaches for modeling the collective search behavior of optimization algorithms mainly in swarm intelligence approaches such as particle swarm optimization [1] and ant colony optimization [2]. The main purpose of these essays is to enable swarms to collaborate and share information with each other and learn from each other. Furthermore, it may help agents to react rapidly to unexpected variations and control the variables of their subsystems when they are communicating with other agents.

Concerning particle swarm optimization (PSO), attempts have been made to formalize the design of such cooperative multi-agent behaviour of particles as a mean to enhance the diversity of the algorithm or to achieve a trade-off between exploration and exploitation. A commonly used cooperation form of PSO is based on the idea of considering multi-swarms (multi-populations), it consists in dividing the whole search space into local subspaces, each of which might cover one or a small number of local optima, and then separately searches within these subspaces. Another way to define multi-agent in PSO is to assign different roles to particles. Thus, different particles can play different roles, and each one of these particles can play different roles during the search processes. A challenging task within this PSO variant is how each particle has to decide which role it will assume. In this paper, Finite state machines (FSM) is applied to model the decision making of an agent with the aim of guiding particles to move toward different promising sub-regions. To do that, the swarm behavior can be represented as finite state machines based on very simple agents and simple interaction rules. That is, a behavior specification defines a set of finite state machines, called options and a set of predefined behavior routines, called basic behaviors. Several approaches have been used to implement

probabilistic FSM. In this paper, we integrate Hidden Markov Model (HMM) - commonly used probabilistic FSM [3]- to learn and predict the most probable states of the probabilistic FSM in order to control particles behaviour of PSO. This process is performed through the Viterbi algorithm that gives the most likely path of states for each particle in each PSO iteration.

The rest of the paper is organized as follows: in the next section, we outline the related works. In section 3, we present our approach. Section 4 presents the obtained results for the experiments. Finally, we conclude and present perspectives to our work.

2 Literature review.

In recent years, there has been increased interest in the use of automation approaches inside PSO to control its behaviour and then to improve its performance. That is, various methods have been proposed to control PSO behaviour and to improve learning capability of particles. We can differentiate between two kinds of control approaches which have been used inside PSO in the literature. In the first one, the control depends on the iteration and then the whole swarm follow the same strategy. In the second one, the control depends on the particle itself. That is, at each iteration, particles are grouped into sub-swarms, and the particles of each swarm have a specific role in the swarm (as in multi-agent systems). This type corresponds to the adaptivity control.

On the one hand, an example of the first case can be found in our previous work [4] (an extension of this paper has been accepted to be published as a chapter in a Springer book) in which hidden markov model (HMM) has been used inside PSO to have a stochastic control on the state classification at each iteration. The mentioned paper has extended the adaptive PSO proposed by [5] which has proposed to update the values of the acceleration factors c1 and c2 according to four defined states which are: exploration, exploitation, convergence and jumping-out.

On the other hand, concerning the second type, [6] proposed four operators which play similar roles as the four of states the adaptive PSO defined in [5]. Their approach is based on the idea of assigning to each particle one among different operators based on their rewards. Furthermore, this type of control is related to the concept of cooperative swarms which has been introduced by [7]. This principle has been achieved in their paper by using multiple swarms to optimize different components of the solution vector cooperatively. This issue can also be treated by clustering approaches as proposed by [8]. Their approach consists of assigning particles to different promising subregions basing on a hierarchical clustering method. We can see that this variant of PSO (multi-swarm PSO) has been presented in the literature as a specific and separate algorithm known by multi-swarm optimization [9]. in the proposed variant, the authors has been inspired by the quantum model of atoms to define the quantum swarm. Also, another grouping approach has been suggested by [10].

More generally, four main categories have been proposed to improve PSO performance, which are: configuration of the parameters (adaptivity control), the study of neighbourhood topology of particles in the swarm of PSO, hybridization with other optimization algorithms and integration of learning strategies (diversity control). Concerning the two types mentioned at the beginning of this section. The former correspond to the first type, while the latter is related to the second one. Furthermore, the control the PSO parameters has been proposed in a number of papers with the purpose of achieving a trade-off between the diversity and the convergence speed. It has generally been done using learning strategies such as the comprehensive learning [11] approach in where each particle learns from another particle which is chosen according to a learning probability. Concerning hybridization, it is a long standing of PSO and example of improvement can be found in [12].

The issue of the interaction between swarm intelligence and multi-agent systems has been given much attention in the last few years in particular by the popularization of the swarm robotic field. In particular, [13] affirmed the concept of swarm appears nowadays closely associated with intelligent systems in order to carry out useful tasks. The author also analysed qualitatively the impact of automation concepts to define the intelligent swarms. Moreover, [14] have outlined the main characteristics of swarm robotics and analysed the collective behaviour of individuals in some fields. They affirmed that finite state machines are one of the most

used adequate approaches to model this behaviour. Another commonly used approach for this purpose is reinforcement learning.

A practical example of using FSM in swarm robotics is described in [15]. More generally, [16] presented a detailed description about using Hierarchical FSM for behaviour Control, The use of FSM for this purpose can be justified by the fact that there are a finite number of priori pre-defined organizational forms of multi-agent systems. In particular, probabilistic FSM has been used [17] for aggregation purpose in swarm robotics. Concerning HMM (which is a type of probabilistic FSM used especially for learning [18]), it has been successfully applied for controlling purpose in a number of engineering problems [19].

An example of using the multi-agent concept in PSO can be found in [20]. That is, incremental social learning which is often used to improve the scalability of systems composed of multiple learning agents has been used to improve the performance of PSO. Furthermore, [21] proposed a multi-agent approach which combines simulated annealing (SA) and PSO, we can remark that their idea is related to the generic notion of hyper-heuristics which consists of finding the most suitable configuration of heuristic algorithms. [22] has cited the may features obtained by using agents in configuring metaheuristics which are distributed execution, remote execution, cooperation and autonomy.

The using of multi-agent concepts can be useful to self-organise particles in PSO using simple rules as defined by [23].Their main idea was to define six states which are cohesion, alignment, separation, seeking, clearance, and avoidance. Furthermore, the finite state machine has been used for movement control. That is, the states of the FSM has been defined by a collection of velocity components and their behavior specific parameters. Furthermore, the population has been divided into two swarms in order to introduce the divide and conquer concept using genetic operators. Another automation approach which can be used inside PSO is cellular automata (CA). CA can be considered as an arrangement of FSM. It can be used for instance split the population of particles into different groups across cells of cellular automata. [24] has integrated it in the velocity update to modify the trajectories of particles.

Regarding Hybridization between PSO and FSM, it has also be done in most cases with the aim of improving FSM performance by PSO as in [25].

3 Probabilistic FSM design of PSO

3.1 Approach background

Firstly, the native PSO has been introduced by [1]. The PSO concept consists of changing the velocity of each particle at each iteration toward its pBest and gBest locations. Then the velocity and the position of each particle is updated according to Eqs. (1) and (2)

$$v_i = w\, v_i + c_1\, r_1(pBest - x_i) + c_2\, r_2(gBest - x_i) \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

The detail of these parameters can be found for instance in [4].

In our approach, we interest especially on introducing probabilistic FSM to control particles state in PSO based on the defined states in [6]. Each particle is viewed as automata having four finite states which are exploration, exploitation, convergence and jumping-out. More detail on these

states is described in the following sections. Among the existing probabilistic FSM [3], we have chosen HMM to address this issue as the most common type of probabilistic FSM. Indeed, HMM has the ability to learn states of our automata from hidden observation based on the maximum likelihood estimation [18], this learning feature of HMM is used to control the particles individually cross PSO iterations.

3.2 Probabilistic FSMs definition

We define each particle as a machine having different finite states. A particle of PSO can have many alternative traces for a given input because of the random behavior of the PSO algorithm. FSM associated with a particle is a little machine that feeds with inputs and provides many outcomes across iterations. We can say that the particle is associated with a random process.

During iterations, a particle is a probabilistic FSM related to a state $\{x_i\}_{i\in N}$ that generates outcome or also called observation $\{y_i\}_{i\in N}$.

This definition yield to have at each iteration several groups of particles, each one plays a defined role according to its identified state machine. So, we can have for instance 40 particles, in which 20 particles explore throughout the search space, 10 others are exploiting, 7 are converging, and 3 are jumping out. Thus, particles are divided to sub-swarms with different states. The change of state or role of particles during iterations is governed by their associated probabilistic FSMs which is defined by the following formalism for each individual particle:

- Outcomes $\{y_i\}_{i\in N}$
- State $\{x_i\}_{i\in N}$
- A=(a_{ij}) The state transition matrix: $P(x_t = i|x_{t-1} = j)\ i,j \in N$, t : iteration number.
- $B = (b_{jk})$ The emission probabilities of outcomes: $P(Y_t = k|X_t = j)\ \ k,j \in N$, t : iteration number.

Our approach consists of finding the most suitable current state by finding the most probable trace for the given input of states across iterations. This problem constitutes a classical decoding problem in HMM theory and resolved by Viterbi algorithm.

3.3 Probabilistic FSM parameters

As mentioned earlier, four states of the Probabilistic FSM have been defined which are: exploration, exploitation, convergence and jumping-out. These states generate an outcome to define below.

Particles are moving around the search space and change position and velocity at each iteration step according to eq 1 and 2. Consider the mean distance of each particle i to all the other particles as d_i. And calculate the measure :

$$l = \left|\frac{d_{Pbest} - d_i}{d_{max} - d_{min}}\right|, Pbest \text{ is the best particle of current iteration.} \quad (3)$$

l is considered as outcome for as particle viewed as an FSM. l belong to subintervals of [0,1] ([0,0.2], [0.2,0.3], [0.3,0.4], [0.4,0.6], [0.6,0.7], [0.7,0.8], [0.8,1]). We divide [0,1] to seven subintervals, so the outcomes $\{y_i\}_{i\in[1,7]}$ will be number of subintervals which belong l. Emission probabilities are deduced from defuzzification process of an evolutionary factor in [5] as follow:

$$P = \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0.25 & 0.25 & 0 \\ 0 & 0.25 & 0.25 & 0.5 & 0 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 2/3 \end{bmatrix} \quad (4)$$

We take state transitions probabilities as equiprobable for all possible transitions, the same as in [4].

3.4 Our proposed algorithm

In our approach, we have use Viterbi algorithm [26] at each iteration and for each particle of the multi-agent PSO to control particles states. Transitions between the four states is updated according to rules governed by the Probabilistic FSM. Moreover, according to each state for the particle given by PFSM, PSO parameters are adjusted, especially acceleration parameters c_1, c_2 and inertia weight w with elastic learning in convergence state [5]. It is done based on APSO parameters update in [5] and [4], see algorithm 1 and eq 5 :

Algorithm 1: Adaptive acceleration update in APSO [5]

Data: Position and accelerations factors
Initialization: positions and accelerations factors c1 and c2 ;
if state = exploration **then** Increasing c1 and Decreasing c2 ;
else if state = exploitation **then**
Increasing c1 and Slightly Decreasing c2
else if state = jumping out **then**
Increasing Slightly c1 and Increasing c2
end
else if state = convergence **then**
Decreasing c1 and Increasing c2
end
Return c1 and c2
Result: Updated acceleration factors

$$\omega(l) = \frac{1}{1 + 1.5e^{-2.6l}} \in [0.4, 0.9] \; \forall \, l \in [0, 1] \qquad (5)$$

The PFSM-PSO evaluate the population distribution and particle fitness, a real-time state estimation procedure is performed to identify one of four states: exploration, exploitation, convergence, and jumping out. It enables the automatic control of every particle state.

As follow, the algorithm designing the PFSM-PSO iterations:

Algorithm 2: PFSM-PSO

Data: The objective function (f)
Initialization: positions, velocities of particles, accelerations factors and PFSM parameters; Set t value to 0;
while (number of iterations $t \leq t_{max}$ not met) **do**
for i = 1 to the number of particles **do**
Decoding specific particle state (viterbi) ; Update w according to Equation (5) ;
Update c1 and c2 values according to the corresponding state (algorithm 1) ;
Update velocities and positions according to Equation (1) and (2) ;
compute $f(x_i)$;
if ($f(x_i) \leq$ fbest) **then**
fbest $\rightarrow f(x_i)$;
pbest $\rightarrow x$;
end
if (f (pbest) $\leq$ fGbest) **then**
fGbest $-\!\!\rightarrow$ fbest ;
gbest $-\!\!\rightarrow$ Xbest ;
end
if state = convergence **then**
Elistic learning [5];
end
end
t $\rightarrow$ t + 1 ;
end
Return p_{best} and f_{best} ;
Result: The solution based on the best particle in the population and corresponding fitness Value

4 Tests and comparisons

In this part, tests of the proposed design approach PFSM-PSO are performed. Experimentations are done with several benchmark functions and compared with other state of the art PSO variants.

4.1 Parameters setting

Ten benchmark functions constitute fitness function used for experimentation as shown in Table 1. Cross executions, and for each function, the best and the average value used for comparison.

Test functions	Name	Type
$f_1 = \sum_{i=1}^{D} [100\,(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	Rosenbrock	Unimodal
$f_2 = \sum_{i=1}^{D} (\lvert x_i + 0.5 \rvert)^2$	Step	Unimodal
$f_3 = \sum_{i=1}^{D} x_i^2$	Sphere	Unimodal
$f_4 = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$	Tablet	Unimodal
$f_5 = \sum_{i=1}^{D} (\sum_{i=1}^{D} x_i)^2$	Quadric	Unimodal
$f_6 = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	Rastrigrin	Multimodal
$f_7 = -20\exp(-0.2\sqrt{\frac{1}{D} x_i^2})$	Ackley	Multimodal
$f_8 = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \Pi\cos(x_i / \sqrt{i}) + 1$	Griewang	Multimodal
$f_9 = \sum_{i=1}^{D} x_i \sin(\sqrt{x_i})$	Schwefel	Multimodal
$f_{10} = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{1/2\,(x_1^2 + x_2^2) + 2}$	Drop wave	Multimodal

Table 1 Description of Benchmark functions

Table 2 shows ten chosen PSO variant from literature. The same initial values of acceleration and inertia weight coefficients ($c_1 = c_2 = 2, \omega = 0.9$) are used. Swarm size is 30 with dimension of 30. Each run contains 1000 generation of optimization process.

Algorithm	Name	Reference
YSPSO	PSO with compressibility factor	[27]
SELPSO	Natural selection based PSO	[28]
SecVibratPSO	Order oscillating PSO	[29]
SecPSO	Swarm-core evolutionary PSO	[30]
SAPSO	Self-adaptive PSO	[28]
RandWPSO	Random inertia weight PSO	[31]
LinWPSO	Linear decreasing weights PSO	[31]
CLSPSO	Cooperative line search PSO	[30]
AsyLnCPSO	Asynchrous PSO	[28]
SimuAPSO	PSO with Simulated Annealing	[30]

Table 2 Compared variants of PSO

4.2 Performance evaluation

For each of benchmark functions shown in Table 1, and each PSO variant of Table 2, ten executions are done with 1000 generations. The best and the average value resulted from experimentations are given in the table below:

Functions		APSO	PSO	SimuA-PSO	Sec-PSO	RandW PSO	YSPSO	SelPSO	SecVibratPSO	SAPSO	LinWPSO	AsyLnCPSO	PFSM-PSO
f_1	Best	49	13566	115719	4689	9843	1420	16382	275	5095	7067	3765	**0.48**
	Mean	150	24618	288102	10930	33384	2726	27998	32132	14850	20280	11045	**27**
f_2	Best	0	5.1e-09	5.8e-06	1.9e-31	2.1e-12	0	8.6e-10	7.3e-10	0	0	0	**0**
	Mean	0	6.5e-05	0.04	9.8e-12	3.6e-05	0	1.8e-05	0.03	0	0	3.1e-30	**0**
f_3	Best	0.01	26.55	93.19	20.58	37.7	6.77	36.01	0.8	18.20	17.66	22.89	**5.74-6**
	Mean	0.05	50.15	188.56	38.79	64.09	13.77	63.59	71.05	31.96	40.53	34.58	**5.79-5**
f_4	Best	0.02	94.97	251.36	71.4	110.48	23.72	30.47	17.42	37.21	51.24	21.73	**5.74-6**
	Mean	0.05	135.82	396.21	110.48	246.78	42.31	77.51	199.04	84.79	95.74	44.32	**5.01-5**
f_5	Best	16435	629e+5	587e+6	421e+5	102e+6	127e+4	839e+5	107e+6	459e+5	165e+5	131e+5	**941**
	Mean	67851	196e+6	210e+7	978e+5	434e+6	843e+4	210e+6	562e+6	166e+6	155e+6	384e+5	**223**
f_6	Best	8.24	266.20	358.22	208.54	293.88	142.44	285.16	221.93	165.66	170.64	193.83	**7.53**
	Mean	16.14	307.24	462.02	262.89	322.35	175.84	315.60	344.76	273.31	261.77	291.54	**8.55**
f_7	Best	0.03	4.79	6.9436	4.8963	5.403	3.1785	5.665	1.2753	3.8279	4.7261	5.8372	**0.001**
	Mean	0.31	5.61	8.6336	5.2716	6.5613	4.2219	6.1951	4.4528	5.2531	5.6829	6.8189	**0.004**
f_8	Best	7.50e-5	0.17	0.52	0.15	0.25	0.05	0.27	0.05	0.13	0.14	0.08	**1.6-7**
	Mean	0.01	0.39	0.87	0.25	0.45	0.12	0.41	0.42	0.25	0.31	0.23	**0.003**
f_9	Best	-118.35	-3e+287	-7.2+47	-4+158	-	-3+34	-1e+308	-	-1+231	-1e+220	-3e+47	**-118.35**
	Mean	-118.34	-3e+286	-1e+47	-9e+157	-	-3e+33	-	-	-1e+230	-1e+219	-3e+46	**-118.34**
f_{10}	Best	-1	-1	-0.92	-1	-0.94	-1	-0.99	-0.93	-1	-1	-1	**-1**
	Mean	-1	-0.95	-0.74	-0.96	-0.93	-0.98	-0.95	-0.82	-0.97	-0.96	-0.98	**-1**

Table 3 Results comparisons with other variants of PSO

The proposed approach has very distinguishable in Table 3, better results than the majority of the state of the art. It exceeds in some cases the order of 10^4 improvements compared to other PSO variants. Very impressive PSO performances in term of solution accuracy are clearly observable.

In term of convergence speed, plots from figure 1 to 5 gives good convergence in favour of PFSM-PSO. In most cases, the solution is found after less than 80 iterations except for Rastrigrin function.

Figure 1 Comparison Rosenbrock and Step functions

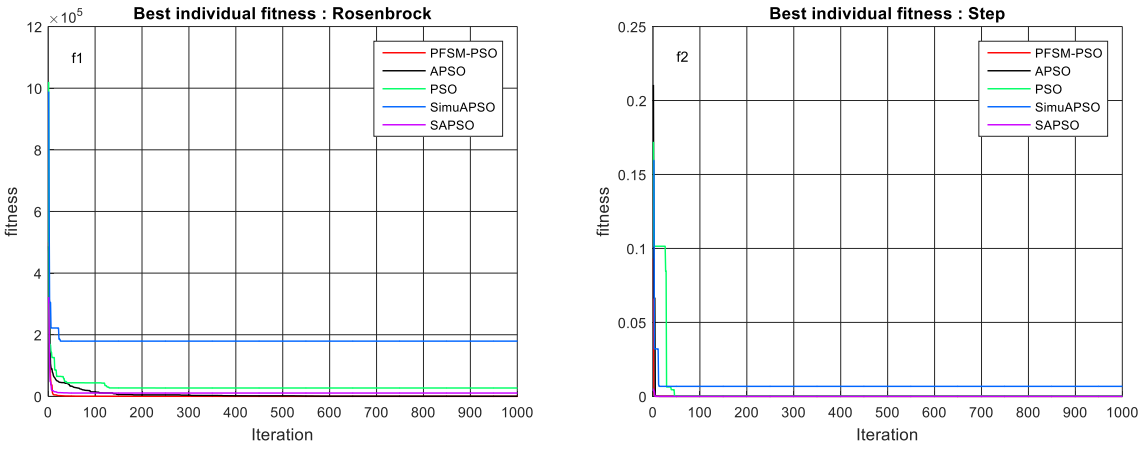


Figure 2 Comparison on Sphere and Tablet functions

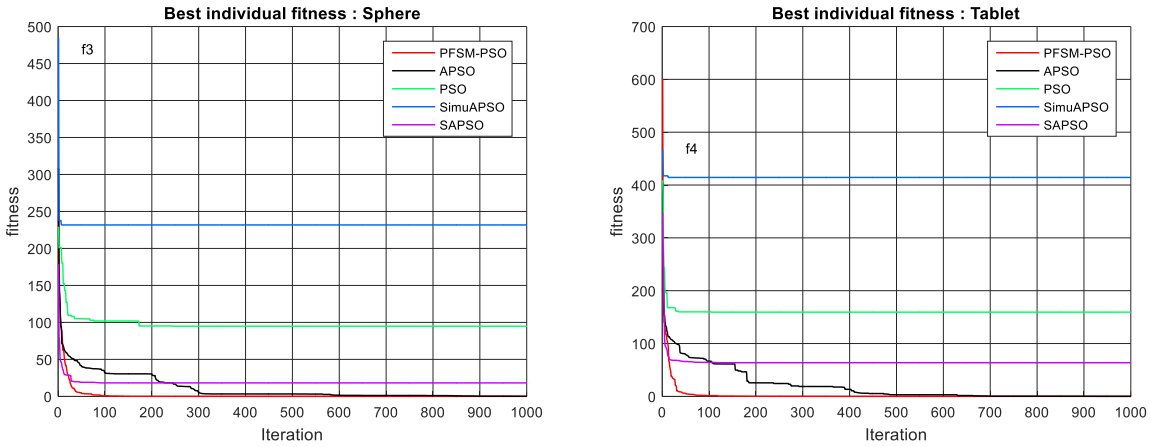


Figure 3 Comparison on Quadric and Rastrigrin functions

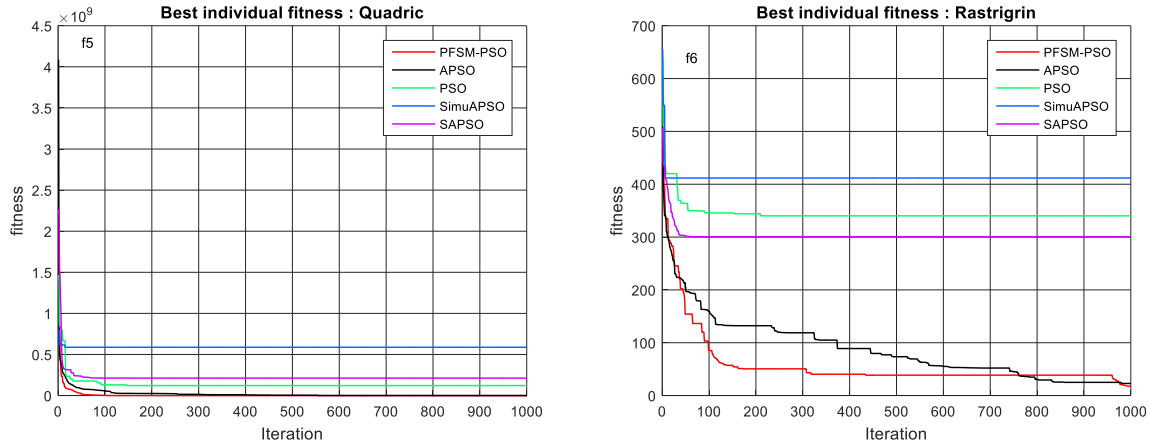


Figure 4 Comparison on Ackley and Griewang functions

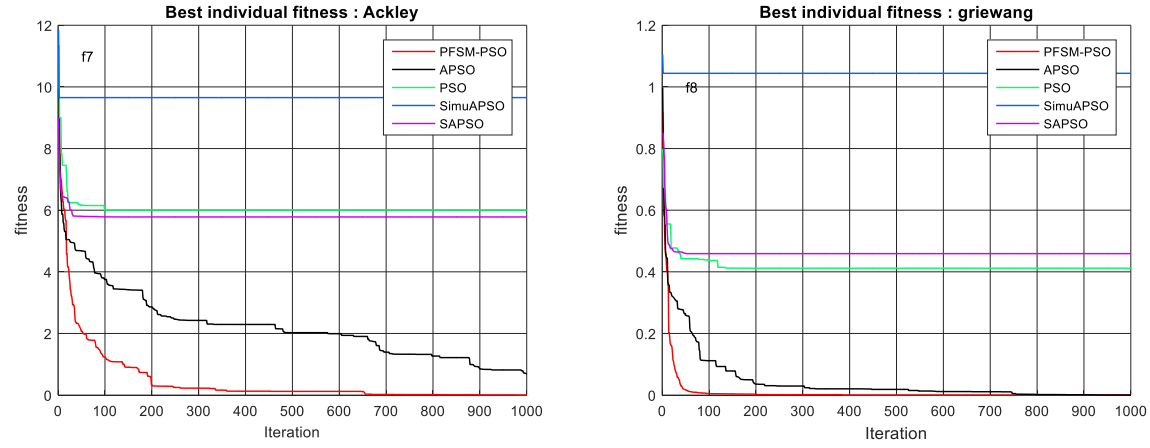
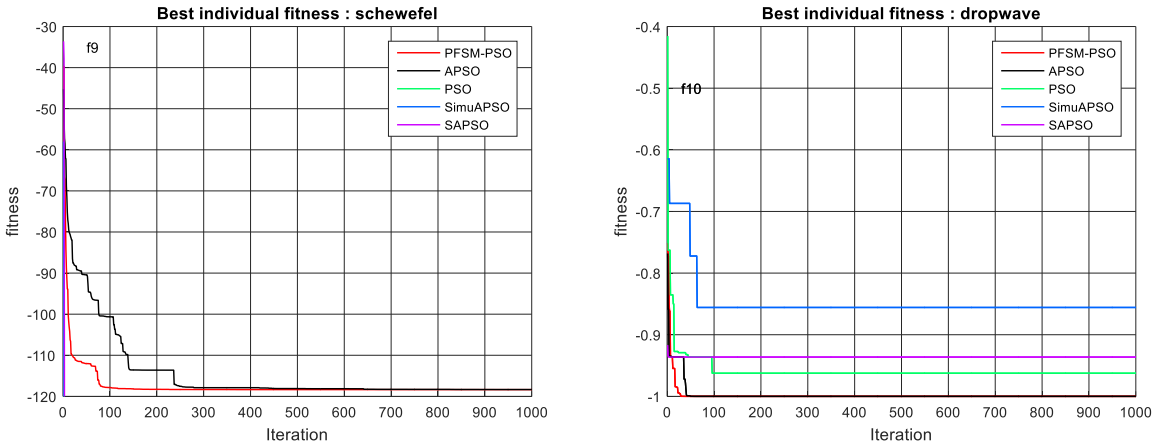


Figure 5 Comparison on Schwefel and Drop wave functions



Even if our approach has given more accurate results, it is much more computationally expensive because the control mechanism must be done for each particle at each iteration. Therefore, our approach may be useful especially in swarm robotics which may complex optimization problems.

5 Conclusion

In this paper, we have shown how a probabilistic FSM, which has been often designed for the control of multi-agent systems, can be used for the control of particles behaviour in PSO. To do that, individual state control by PFSM is associated to each particle during the search process. Experimental results have shown that has given competitive computational performance than some PSO variants. Future research should attempt to define collaborative rules between sub-swarms of particles and illustrate its applicability to real problems.

References

[1] Kennedy, J., Eberhart, R..: Particle swarm optimization. Proceedings of IEEE international conference neural networks IEEE (1995) 1942–8

[2] Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. Evolutionary Computation, IEEE Transactions (1997) 53–66

[3] Vidal, E., Thollard, F., De La Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite-state machines-part ii. Pattern Analysis and Machine Intelligence, IEEE Transactions (2005) 1026–1039

[4] Aoun, O., Sarhani, M., Afia, A.E.: Hidden markov model classifier for the adaptive particle swarm optimization. In: The XI Metaheuristics International Conference (MIC), Agadir, Morocco (2015)

[5] Zhan, Z.H., Zhang, J., Li, Y., Chung, H.H.: Adaptive Particle Swarm Optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics (2009) 1362–1381

[6] Li, C., Yang, S., Nguyen, T.T.: A self-learning particle swarm optimizer for global optimization problems. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions (2012) 627–646

[7] van den Bergh, F., Engelbrecht, A.: A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation (2004) 225–239

[8] Yang, S., Li, C.: A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. IEEE Transactions on Evolutionary Computation (2010)

[9] Blackwell, T., Branke, J., et al.: Multi-swarm optimization in dynamic environments. In: EvoWorkshops. Volume 3005., Springer (2004) 489–500

[10] Mirjalili, S., Lewis, A., Sadiq, A.S.: Autonomous particles groups for particle swarm optimization. Arabian Journal for Science and Engineering (2014) 4683–4697

[11] Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. Evolutionary Computation, IEEE Transactions (2006) 281–295

[12] Sarhani, M., Afia, A.E.: Facing the feature selection problem with a binary pso-gsa approach. In: The XI Metaheuristics International Conference (MIC), Agadir, Morocco (2015)

[13] Beni, G.: From swarm intelligence to swarm robotics. Swarm robotics. Springer (2004) 1–9

[14] Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence **7** (2013) 1–41

[15] Konig, L., Mostaghim, S., Schmeck, H.: Decentralized evolution of robotic behavior using finite state machines. International Journal of Intelligent Computing and Cybernetics (2009) 695–723

[16] Risler, M.: Behavior control for single and multiple autonomous agents based on hierarchical finite state machines. PhD thesis, tuprints (2010)

[17] Soysal, O., Sahin, E.: A macroscopic model for self-organized aggregation in swarm robotic systems. In: Swarm robotics. Springer (2006) 27–42

[18] Dupont, P., Denis, F., Esposito, Y.: Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms. Pattern recognition (2005) 1349–1371

[19] Elliott, R.J., Aggoun, L., Moore, J.B.: Hidden Markov models: estimation and control. Volume 29. Springer Science & Business Media (2008)

[20] De Oca, M.A.M., Stützle, T., Van den Enden, K., Dorigo, M.: Incremental social learning in particle swarms. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on (2011)

[21] Aydin, M.E.: Coordinating metaheuristic agents with swarm intelligence. Journal of Intelligent Manufacturing (2012) 991–999

[22] Diaz, D.M.: Agent-Based Configuration of (Metaheuristic) Algorithms. PhD thesis, Humboldt University of Berlin (2005)

[23] Bengfort, B., Kim, P.Y., Harrison, K., Reggia, J.A.: Evolutionary design of self-organizing particle systems for collective problem solving. In: Swarm Intelligence (SIS), 2014 IEEE Symposium on, IEEE (2014) 1–8

[24] Shi, Y., Liu, H., Gao, L., Zhang, G.: Cellular particle swarm optimization. Inf. Sci. (2011)

[25] El-Maleh, A.H., Sheikh, A.T., Sait, S.M.: Binary particle swarm optimization (bpso) based state assignment for area minimization of sequential circuits. Applied soft computing (2013)

[26] Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE (1989) 257–286

[27] LIU, L.l., GAO, X.b.: An adaptive simulation of bacterial foraging algorithm. Basic Sciences Journal of Textile Universities (2012) 022

[28] Jiang, W., Zhang, Y., Wang, R.: Comparative study on several pso algorithms. In: Control and Decision Conference (2014 CCDC), The 26th Chinese. (2014) 1117–1119

[29] Jianxiu, H., Jianchao, Z.: A two-order particle swarm optimization model [j]. Journal of Computer Research and Development (2007) 004

[30] Wang, S., Chen, M., Huang, D., Guo, X., Wang, C.: (Dream effected particle swarm optimization algorithmâ)

[31] Wu, Z.: Optimization of distribution route selection based on particle swarm algorithm. International Journal of Simulation Modelling (IJSIMM) (2014)

Hidden Markov Model for a Self-Tuning of Simulated Annealing Geometric Cooling Law Parameter

M. Lalaoui[1], A. El Afia[1], R. Chiheb[1]

[1] *ENSIAS - Mohammed V University, Rabat – Morocco*
med.lalaoui@yahoo.fr, a.elafia@um5s.net.ma, craddouane@yahoo.fr

Abstract. The Simulated Annealing (SA) is a stochastic local search algorithm. The efficiency of the simulated annealing algorithm involves the adaptation of the cooling law. In this paper, we integrate Hidden Markov Model (HMM) in SA to adapt the geometric cooling law at each iteration, based on the history of the search. An experiments was performed on many benchmark functions and compared with others SA variants.

Keywords: Simulated Annealing, Hidden Markov Model, Classification.

1 Introduction

Since the publication of the first article related to the SA algorithm [19], researchers have been trying to increase the rate of convergence and to reduce the execution time of simulated annealing. Researchers focused since, the first version of simulated annealing algorithm described by [19], on two strategies in order to improve the performance of SA. The first strategy was the implementation of parallel simulated annealing [1, 6, 13]. The second one was about the optimization of cooling schedule and the adaptation of parameters.

The cooling schedule is an important set of parameters that governs the convergence of SA. The set of annealing schedule as defined by [5], includes the cooling factor, starting and stopping temperature, and the number of moves at each temperature. The cooling factor is the most influential feature among the set of annealing schedule. This factor can be defined as the method for which the algorithm reduces the temperature to its next value. If the temperature is reduced very quickly, a convergence to a local minimum may occur. However, if it is reduced too slowly, the algorithm takes a long time to converge.

Various strategies of cooling schedules have been investigated by researchers. It has been firstly proven by [18] that logarithmic schedule ensures the SA's convergence to a global minimum. In addition, its temperature's decrease at each step t is governed by the formula $\theta_t = \theta_0/log(1+t)$, but this logarithmic cooling is too slow in practice [15]. A faster schedule was proposed by [9] called the Cauchy schedule in which θ_t converges to the global minimum when moves follow a Cauchy distribution. The most frequently used decrement rule is geometric schedule [19] in which $\theta_t = \alpha.\theta_{t-1}$, where $0.85 \leq \alpha \leq 0.96$and α is a constant.

This article presents a new approach to enhance SA, which consists of tuning the geometric cooling law during runtime, using the Hidden Markov Chain. The main idea is to predict the best cooling law parameter based on history of the run. To do that, we train the HMM model by updating its parameters using Baum-Welch algorithm. Then we proceed to a classification process through the Viterbi algorithm which gives the most probable cooling law parameter.

The rest of this paper is organized as follows. First, section 2 is devoted to literature review, section 3 describes hybridization methodology of HMM, SA, then section 4 presents and discusses the experimental results, and finally, we conclude the paper in Section 5.

2 Literature review

The geometric cooling law is the most used in the literature. It originates from work on cooling schedule by Kirkpatrick [19]. The major drawbacks of this cooling policy is that it does not take into account the state of the system during the search. Furthermore, it can be trapped in local minima and it takes too long to find a reasonable solution.

The cooling rate is critical for the effectiveness of the algorithm. Thus, the quality of solution obtained by SA highly depends on the rate of cooling followed by the algorithm described as the method for which simulated annealing reduces the temperature to its next value, a quick cooling will result in loss of good solution candidates, whereas a too slow cooling will increase the computation required and the time for it [7].

To overcome this, researchers apply a different cooling rates, which depend on the search phase. As a result, more time will be spent at low-temperature phases. When the temperature reaches a certain level, the system will not be able to accept a broad diversification. Abramson et al. [4] proceeds to a temperature rise during annealing process when a local minimum is detected. The disadvantage of this approach according to [4], is that it may take a number of cycles until the temperature is high enough to allow changes. In contrast, an excessive heating increases the total time needed to solve a problem due to the introduction of useless movements, while not sufficient heating could cause a blockage in a local minimum.

According to Battiti and al. [16], a self-analysis during the search can enhance the performance of the algorithm. Because the simulated annealing is a memory less heuristic. During the execution of this algorithm, the next move depends only on the current state and not on previous states history.

In addition, [7] confirms that there is no practical cooling schedule that can guarantee the optimality or near-optimality of the annealing process. [22] proposed a simulated annealing algorithm with automatic learning. The search history is incorporated into the conventional simulated annealing algorithm. Unlike classical SA, the SA with self-learning capability has the ability to learn as the search progresses by means of comparing the new states with those memorized. For a large problems his approach can increase the memory use. Another learning method was proposed by [20]. This method uses a knowledge base updated at each iteration. This base holds the search history and guides the further search.

[21] developed an adaptive SA using Fuzzy Logic Controller. His approach was applied to the geometric cooling parameter with the aim to control temperature. The Fuzzy Logic Controller improved the search for solution by reducing unnecessary search time in solution space where no improvement in solutions occurs, and concentrating search where improvement in solution may happen by continuously observing the status of solution improvement and the current temperature. At each stage, the next temperature is determined by to the rules set at Fuzzy Logic Controller. [8] adopts an online parameter tuning where the parameters are controlled and updated dynamically throughout the execution of the SA algorithm. By sampling the progress of the global optimization process the Mamdani fuzzy controller changes internal parameters, trying to escape from local minima attraction basins. It is shown that, by increasing the algorithm perception of slow convergence, it is possible to speed it up significantly, thereby reducing considerably the user's task of parameter tuning.

One major drawback of fuzzy logic controllers is the difficulty encountered in the construction of a rule-base that is suitable for the controlled process. In the one hand, the maintenance of a rule base becomes complex and time-consuming as the size of a system increases. In the other hand, fuzzy logic is not adequate for reasoning about uncertain evidence, and the other disadvantage of the method is the lack of appropriate tools for analysis of the controllers performance, such as stability and optimality [3].

[17] proposed another learning approach, where the simulated annealing parameter's was selected as a classification problem. His approach builds a rule for choosing the set of parameters most likely to perform well for a given instance, on the basis of specific features. In addition, in order to better manage the time in possession, [17] employs a cutoff-based cooling scheme, so the algorithm is allowed to decrease the temperature prematurely. This speeds up the initial part of the search and saves iterations that can be used later in the search.

The Hidden Markov Model (HMM) [11] is a probabilistic model. It has been applied in many fields where information is hidden and depends on other observable data. Unlike Fuzzy Logic Controller, HMMs success is due to ability to deal with the variability by means of stochastic modeling. HMMs was very successful in speech recognition where signals are time varying and in process monitoring because of their strong capability in representing non-stationary signals by statistical parameters. Furthermore, in pattern recognition, there always exists uncertainty, randomness and incompleteness originating from various sources.

Thus, the main idea behind our approach is to control the behavior simulated annealing at each temperature stage by predicting the optimal geometric cooling law parameter according to history of the run using HMM.

3 Proposed approach

To enhance the performance of SA, an hybridization with the HMM was adopted. During the run, the hidden Markov model perform classification based on observable sequence generated from a set of rules. This sequence allow the model to guess the hidden state which can be a slow cooling to help the algorithm converge to the global minima, a rapid cooling to speed up the search when no improvement in solution occurs , or re-annealing to escape from a local minimum (fig. 1).

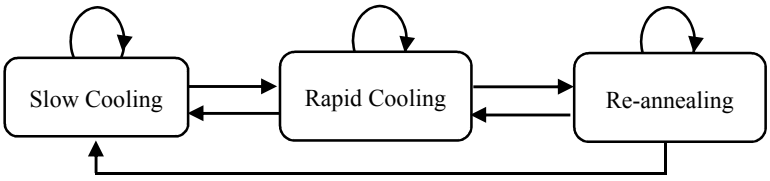


Fig. 1. Markov chain for simulated annealing algorithm

The Hidden Markov Model can be defined as 6-tuple ($SVABO\pi$ ^{0}) where:

- S= $\{S_1, S_2, S_3\}$ is set of hidden states, which is respectively: exploration, exploitation and escape from a local minimum.
  - S_1: is the slow cooling state correspond to the classical variant of simulated annealing with a geometric temperature decrease law, where the cooling factor α=0.9.
  - S_2: is the rapid cooling state correspond the same classical variant of simulated annealing, where the cooling scheduler is faster than the previous variant (α=0.5).
  - S_3: is a fast simulated annealing variant with a reannealing abilities to escape from locals minima.
- $V = (V_1V_2,..V_5)$ is the set of the observation state.
- $O = (O_1O_2,..O_T)$ is a sequence of observations.
- A $= (a_{ij})$ is a transition probability matrix, where a_{ij} is the probability that the state at time $t + 1$ is S_j , is given when the state at time t is S_i.
- $\pi^0 = (\pi_1^0\pi_2^0\pi_3^0)$ is the initial probability, where π_i^0 is the probability of being in the state S_i.
- $B = (b_{it})$ is the observation probabilities, where b_{it} is the probability of observing O_t in state S_i. This observations matrix B of hidden markov model is estimated at early stage by Maximum Likelihood Estimation (MLE).

The main purpose of this model is to estimate state sequence S that best explains the observation sequence O. To generate the observable sequence of HMM model. We use a progression rate ρ, and a measure of the acceptance rate w_t of the proposed solution described in Eq.1.

$$\rho = \frac{Number\ of\ proposal}{Inner\ loop\ \times\ Outer\ loop} \qquad w_t = \frac{Number\ of\ accepted solutions}{Number\ of\ proposal} \qquad (1)$$

Where, the number of proposal is the number of solution generated by the neighborhood function in each iteration. The inner and the outer loop are the maximum number of iterations established for SA to find the best solution. The number of accepted solutions at iteration t is the accumulated number of accepted solution until the current iteration.

The acceptance rate w_t, and the progression rate ρ are then used to generate a sequence of class from a set of rules as follow:

- V_1: Observation of little decrease of acceptance rate.
- V_2: Observation of no improvement in cost function even if the progression rate is less than 50%.
- V_3 : Observation of a great decrease of acceptance rate.
- V_4 : Observation of a little increase of acceptance rate.
- V_5: Observation of a huge increase of acceptance rate.

Observations States	Rules
V_1	Rule 1: $0 \leq \frac{w_{t-1}-w_t}{w_{t-1}} \leq 0.5$
V_2	Rule 2: $0 \leq \frac{w_{t-1}-w_t}{w_{t-1}} \leq 0.1$ and $\rho < 50\%$
V_3	Rule 3: $0.5 < \frac{w_{t-1}-w_t}{w_{t-1}} < 1$
V_4	Rule 4: $0 \leq \frac{w_t-w_{t-1}}{w_{t-1}} \leq 0.5$
V_5	Rule 5: $0.5 < \frac{w_{t-1}-w_t}{w_{t-1}} < 1$

Table 1. : List of rules

Algorithm 1: Generate_Observation

Input: w_t, ρ, Rule_1$(w_t\rho)$, Rule_2$(w_t\rho)$, Rule_3$(w_t\rho)$, Rule_4$(w_t\rho)$, Rule_5$(w_t\rho)$
Output: O Current Observation
  If Rule_1$(w_t\rho)$ ==TRUE **then** O←1 **End**
  If Rule_2$(w_t\rho)$ ==TRUE **then** O←2 **End**
  If Rule_3$(w_t\rho)$ ==TRUE **then** O←3 **End**
  If Rule_4$(w_t\rho)$ ==TRUE **then** O←4 **End**
  If Rule_5$(w_t\rho)$ ==TRUE **then** O←5 **End**
Return O

The main purpose of this model is to estimate states S that best explains the observation sequence O. Given the observation sequence $O = O_1 O_2 \dots O_T$ and a model $\lambda = (AB\pi)$. Firstly, we estimate the transition and emission probabilities from the first sequence of observation using a supervised training. In which we count frequencies of transmissions and emission of the model:

Algorithm 2: MLE

Input: $O = (O_1 O_2 \dots O_T)$
Output: **A**$=(a_{ij})$,**B** $=(b_{it})$
For i = 1 to T-1 **do** $a_{O_i O_{i+1}} = a_{O_i O_{i+1}} + 1$ **End**
For i = 1 to T **do** $b_{O_i O_i} = b_{O_i O_i} + 1$ **End**
For i = 1 to 3 **do** $A_i = \sum_{j=1}^{3} a_{ij}$ and $B_i = \sum_{j=1}^{T} b_{ij}$ **End**
For $i = 1$ **to** 3 **do**
  For j=1 **to** 3 **do** $a_{ij} = a_{ij}/A_i$ **End**
  For t=1 **to** T **do** $b_{it} = b_{it}/B_i$ **End**
End
Return A

Then we use the Viterbi to select the corresponding state sequence $Q = q_1 q_2 \dots q_T$ that best explains observations, secondly, the Baum Welch adjust the model parameters $\lambda = (AB\pi)$ to maximize $P(O|\lambda)$, i.e., the probability of the observation sequence given the model.

3.1 Viterbi algorithm

After model parameters definition, the Viterbi algorithm is used to build HMM classification process. This algorithm is used to compute the most probable path as well as its probability.

Algorithm 3: Viterbi

Input: S$=(s_1 s_2 s_3)$ $O = (O_1 O_2 \dots O_T)$, A $= (a_{ij})$, B $= (b_{it})$, $\pi^0 = (\pi_1^0 \pi\ {}_2^0 \pi\ {}_3^0)$
Output: $s^* = (s_1^* s_2^* s_3^*)$ the most probable sequence of states
For i = 1 *to* 3 **do** $\delta_1(i) = b_i(o_1)\pi_i$ and $\psi_1(i) = 0$ **End** {Initalization}
For t = 2 to T **do**
  For j=1 to 3 **do**
    $\delta_t(j) = max_{i=1}^{3}[\delta_{t-1}(i)a_{ij}b_j(o_t)]$
    $\psi_t(j) = argmax_{i=1}^{3}\ [\delta_{t-1}(i)a_{ij}]$
  End

End
$P_{max} = max_{i=1}^{3} [\delta_T(i)]$
$s_{10}^* = argmax_{i=1}^{3}[\delta_T(i)]$
For t= $T-1$ to 1 do $s_t^* = \psi_{t+1}(s_{t+1}^*)$ **End**
Return s^*

3.2 Baum Welch algorithm

The Baum–Welch algorithm is used to adjust the parameters of HMM. This training step is based on Forward-Backward algorithm.

- *Forward algorithm*

The first algorithm used by the Baum-Welch algorithm is the Forward algorithm.This algorithm returns the forward variable $\alpha_j(t)$ defined as the probability of the partial observation sequence until time t, with state S_j at time t, $\alpha_j(t)=P(O_1O_1 \dots O_t q_t = S_j|\lambda)$, and we define $P(O|\lambda)$ as the probability of the observation sequence given the model λ.

Algorithm 4: Forward

Input: S=($s_1 s_2 s_3$),O=($O_1 O_2 \dots O_T$), A = (a_{ij}), B =(b_{it}), $\pi^0 = (\pi_1^0 \pi_2^0 \pi_3^0)$
Output : $\alpha = (\alpha_{t+1}(j))$, $P(O|\lambda)$
For $i = 1$ *to* 3 do $\alpha_i(i) = \pi_i b_{i1}$ **End**
For $t = 1$ **to** T-1 **do**
For $j = 1$ **to** 3 **do** $\alpha_{t+1}(j) = \left(\sum_{i=1}^{3} \alpha_t(i) a_{ij}\right) b_{jt+1}$ **End**
End
$\boldsymbol{P(O|\lambda)} = \sum_{i=1}^{3} \alpha_T(i)$
Return $\alpha P\ (O|\lambda)$

- *Backward algorithm*

The second algorithm used by Baum-Welch Backward. This algorithm calculate the backward variable $\beta_i(t)$ defined as the probability of the partial observation sequence after time t, given state S_i : $\beta_i(t) = P(O_{t+1}O_{t+2} \dots O_T|q_t = S_i, \lambda)$

Algorithm 5: Backward

Input : S=($s_1 s_2 s_3$), O=($O_1 O_2 \dots O_T$), A=(a_{ij}), B=(b_{it}), $\pi^0 = (\pi_1^0 \pi_2^0 \pi_3^0)$
Output : $\beta = \beta_t(i)$ the probability of the partial observation sequence
For $i = 1$ **to** 3 **do** $\beta_T(i) = 1$ **End**
For $t = T-1$ **to** 1 **do**
For $i = 1$ to 3 **do** $\beta_t(i) = \sum_{j=1}^{3} a_{ij} \beta_{t+1}(j) b_{jt+1}$ **End**
End
Return β

The Baum-Welch is then used to re-estimate the parameters of the model $\boldsymbol{\lambda}$, which maximizes the probability of the observation sequence. This algorithm is described as follow:

Algorithm 6 : Baum-Welch

Input: S=($s_1 s_2 s_3$), O=($O_1 O_2 \dots O_T$), A= (a_{ij}), B=$(b_{it})\pi^0 = (\pi_1^0 \pi_2^0 \pi_3^0)$,
$\beta = \beta_t(i)\ \alpha = (\alpha_t(i))$, $P(O|\lambda)$
Output: $(\bar{A}) = (\bar{a}_{ij})\ \bar{B} = (\bar{b}_{it})$
Repeat
$[\alpha P\ (O|\lambda)] \leftarrow$**Forward**($O$,A,B,$\pi^0$)
$\beta \leftarrow$**Backward**(O, A, B, π^0)
For t=1 to T do
For i=1 to 3 do
For j=1 to 3 do $\xi_t(ij) = \frac{\alpha_t(i) a_{ij} b_j \beta_{t+1}(j)}{P(O|\lambda)}$ **End**
$\gamma_t(i) = \sum_{j=1}^{3} \xi_{ij}(t)$
End
End

```
  For  k=1 to T   do
     For i=1 to 3   do
       For j=1 to 3  do π̄_i ← γ_i(1) ; ā_ij ← Σ_{t=1}^{T-1} ξ_t(ij) / Σ_{t=1}^{T-1} γ_t(i)   ;  b̄_ik ← Σ_{t=1∩O_t=v_k}^{T} γ_t(i) / Σ_{t=1}^{T} γ_t(i)   End
     End
  End
While (P(O|λ) increase)
Return Ā, B̄
```

In the following we will implement a variant simulated annealing based on hidden Markov models. The interest behind hybridization and simulated annealing with the HMM is improved simulated annealing performance. This approach will allow the SA to overlappe between slow and rapid cooling, this by changing the cooling schedule. Thus, the simulated annealing algorithm hybridization with HMM using Baum Welch and Viterbi algorithms is presented as follow:

Algorithm 7: HMM-SA algorithm

```
Data: The objective function f
Initialization : O : Empty sequence of observation, θ_0: initial temperature , θ_f :final
temperature , starting point x ← x_0 , cmp ← 1
Repeat
       Repeat
           x_new ← x + u   {u is a Random vector from the uniform distribution over [0,1)}
           If   f(x_new) − f(x) ≤ 0 then x ← x_new else P(x ← x_new) = exp(− (f(x_new) − f(x)) / T)  End
       Until equmbrium is approached sufficiently closely at θ_n
       O_n ← Generate-Observation(w_t, Rule_1, Rule_2, Rule_3, Rule_4, Rule_5)
   If cmp ≤ 10 then
                    O ← [O O_n]
                    [A,B] ← MLE(O)
                    state←Viterbi(O,A,B)
                    cmp ←cmp+1
               else
                    O ← [O_{n−9},.. O_n]
                    [A,B] ← Baum-Welch(O,A,B)
                    state←Viterbi(O,A,B);
   End
   Cooling_law ← state
   θ_{n+1} ← Cooling_law(θ_n)
   n ← n + 1
Until θ_n ≤ θ_f indicating that the system is frozen
```

4 Experiment

Our experiments were designed to measure the effects of hybridization of HMM and SA and to show how the proposed approach can improve the solution quality.

4.1 Experiment setup

We have chosen ten benchmarks selected from the literature. They are divided into two groups, unimodal functions with no local minimum except the global one and multimodal functions with many local minima. f_{min} is the known optima of the functions.

Name	Function Formula	Global Minimum
Six-Hump Camel	$f_1(\boldsymbol{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	-1.0316
Rastrigin	$f_2(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	0
De Jong	$f_3(x) = \sum_{i=1}^{D} x_i^2$	0
Schuffer N°4	$f_4(x) = 0.5 + \frac{\cos^2(\sin(\lvert x_1^2 - x_2^2\rvert)) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	0.292579
Colville	$f_5(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$	0
Levy N° 13	$f_6(x) = sin^2(3\pi x_1) + (x_1 - 1)^2[1 + sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + sin^2(2\pi x_2)]$	0
Branin	$f_7(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	0.397887
Quadric	$f_8(x) = \sum_{i=1}^{D}\left(\sum_{i=1}^{D} x_i\right)^2$	0
Shubert	$f_9(x) = \left(\sum_{i=1}^{5} i\, cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\, cos((i+1)x_2 + i)\right)$	-186.7309
Bukin	$f_{10}(xy) = 100\sqrt{\lvert y - 0.01x^2\rvert} + 0.01\lvert x + 10\rvert$	0

Table 1. : Benchmark Functions

The proposed hybridization of SA algorithm and HMM was coded in Scilab programming language and experiments were conducted on a PC with an Intel Core i7-5500U 2.40 GHz (4 CPUs) and 8 GB of RAM. For convenience, SA algorithm with the classical neighborhood relationship and logarithmic cooling schedule is denoted as CSA [2], and SA with geometrical cooling schedule is denoted DEF-SA. The Very Fast Simulated Annealing from L. Ingbe is denoted as VFSA [10]. The Szu and Hartley Fast simulated annealing is referred as FSA [9] and SA with Huang temperature decrease law is labeled as Huang-SA [12].

These variants have been tested using the benchmark functions presented above with dimension D=5. Each function was tested over 50 trials. We eliminated the effects of other factors which play an important role in the performance of algorithm, by choosing the same starting points for all methods (in each run) and their location was chosen to be far from basins of attraction of global minima. Also, we have chosen the same initial acceptance probability and an identical length of the inner and outer loops.

The final temperature of the cooling process, θ_{Final} have been taken close to zero. We fixed θ_{Final} at 10^{-5}. The initial temperature,θ_0, have been calculated from mean energy rises Δf during the initialization. Before the start of the SA, the mean value of cost rises is estimated by a constant number of moves equal to 100. Then, initial temperature θ_0 is calculated using the following formula $\theta_0 = \frac{-\Delta f}{\ln P_0}$ [14], where P_0 is the initial average probability of acceptance and is taken equal to 0.95. The length T of observed sequence was chosen equal to 10.

4.2 Numerical results

The computational results and statistical analyses are summarized in table 2. It provides the details of the results for the test functions. The overall best solution of the total 50 replications is shown in bold. HMM-SA provide the best solution for the test functions f_1,..f_{10}.

In general, HMM-SA algorithm overcomes others variants in most of benchmark functions. In the most cases, our approach gives the better solution except for functions f_3 and f_8.

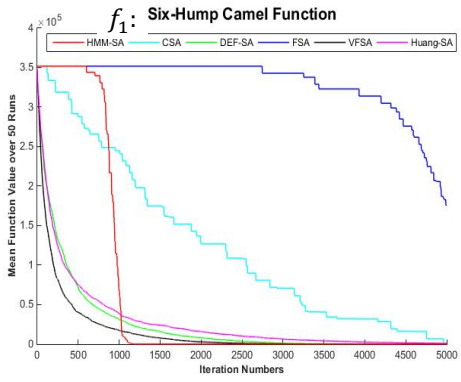
Functions		HMM-SA	CSA	DEF-SA	FSA	VFSA	Huang-SA
f_1	best	**-1.03E+00**	1.24E-03	-1.02E+00	4.65E+01	**-1.03E+00**	**-1.03E+00**
	mean	**-1.03E+00**	8.85E+02	8.98E+00	1.76E+05	3.00E+00	9.68E+02
	std	**1.58E-06**	2.52E+03	2.28E+01	1.57E+05	9.81E+00	2.28E+03
	p-value	0.713	0.514	0.928	0.278	0.464	0.878
f_2	best	**5.99E-07**	2.61E-06	1.46E+02	2.51E-01	6.37E+01	4.40E+01
	mean	**1.27E-03**	2.75E-03	2.76E+02	2.21E+00	9.71E+01	9.47E+01
	std	2.91E-03	3.79E-03	4.75E+01	1.24E+00	7.46E+00	3.41E+01
	p-value	0.896	0.288	0.680	0.231	0.435	0.439
f_3	best	8.77E-12	**1.26E-12**	4.84E-04	1.12E-04	1.51E-08	7.81E-03
	mean	2.48E-06	**1.06E-06**	1.97E-03	2.79E-03	7.58E-02	1.17E+00
	std	4.58E-06	1.15E-06	9.03E-04	2.00E-03	1.68E-01	1.32E+00
	p-value	0.260	0.160	0.489	0.741	0.279	0.350
f_4	best	**5.00E-01**	5.56E-01	5.20E-01	**5.00E-01**	5.24E-01	5.17E-01
	mean	**5.00E-01**	6.91E-01	5.26E-01	**5.00E-01**	5.27E-01	5.24E-01
	std	5.03E+00	2.89E+00	3.94E+02	3.25E-04	1.72E-05	1.27E+00
	p-value	0.368	0.549	0.303	0.529	0.564	0.695
f_5	best	**1.22E-03**	6.08E+00	3.60E+02	1.54E+06	8.93E+01	2.44E+02
	mean	**2.43E+02**	3.33E+05	2.44E+03	1.54E+06	1.71E+03	4.43E+03
	std	3.17E+03	6.24E+05	1.29E+03	9.43E+02	9.93E+02	2.35E+03
	p-value	0.888	0.615	0.268	0.500	0.466	0.324
f_6	best	**3.60E-09**	1.18E-06	6.47E+00	1.83E-04	5.00E+01	7.13E+00
	mean	**4.32E-05**	1.21E-02	7.60E+01	2.25E-02	7.85E+01	8.81E+01
	std	2.27E-04	2.54E-02	2.95E+01	2.78E-02	1.12E+01	4.28E+01
	p-value	0.377	0.550	0.526	0.701	0.512	0.280
f_7	best	**3.98E-01**	8.45E-01	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**
	mean	**3.98E-01**	2.88E+00	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**
	std	1.49E-05	2.40E+00	2.14E-05	2.34E-05	9.49E-06	7.37E-03
	p-value	0.691	0.382	0.302	0.170	0.279	0.833
f_8	best	1.08E-07	1.16E-06	2.39E-03	6.86E-02	**6.79E-12**	9.98E+00
	mean	6.92E-04	1.18E-03	4.87E-02	4.52E-01	**2.62E-09**	1.01E+02
	std	1.17E-03	1.69E+01	1.69E+01	1.69E+01	1.69E+01	1.69E+01
	p-value	0.500	0.500	0.500	0.500	0.500	0.500
f_9	best	**-1.87E+02**	1.37E-09	-7.94E+01	**-1.87E+02**	-4.65E+01	-1.24E+02
	mean	**-1.87E+02**	3.72E-06	-2.95E+01	**-1.87E+02**	-1.79E+01	-5.75E+01
	std	5.96E-09	5.34E-06	1.95E+01	5.72E-02	8.17E+00	1.98E+01
	p-value	0.767	0.325	0.509	0.466	0.136	0.159
f_{10}	best	1.00E-01	2.03E-01	3.10E-01	**8.36E-03**	2.47E-01	2.90E-01
	mean	**1.00E-01**	7.97E-01	6.22E-01	2.46E-01	4.01E-01	6.71E-01
	std	1.19E-04	2.03E-01	7.33E-02	6.82E-05	6.88E-04	7.87E-03
	p-value	0.628	0.452	0.817	0.244	0.408	0.773

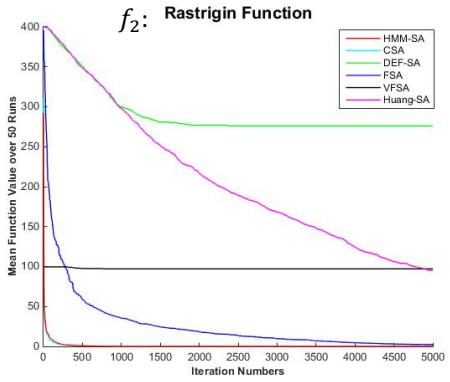
Table 2. Results comparisons between HMM-SA and SA variant's

4.3 Comparison of convergence performance

To obtain further insights into the convergence behavior our approach, HMM-SA method was compared to the three SA variants. The experiments were designed to measure the effects of the hybridization of SA and HMM presented in the previous section. It was noticed that the HMM-SA can converge rapidly to global minimum. The time gained in early stage can be used to converge to a better solution. This behavior is depicted in Fig.2.

It was noticed that the HMM-SA can converge rapidly to global minimum. The time gained in early stage can be used to converge to a better solution. This behavior is depicted in Fig. 2.





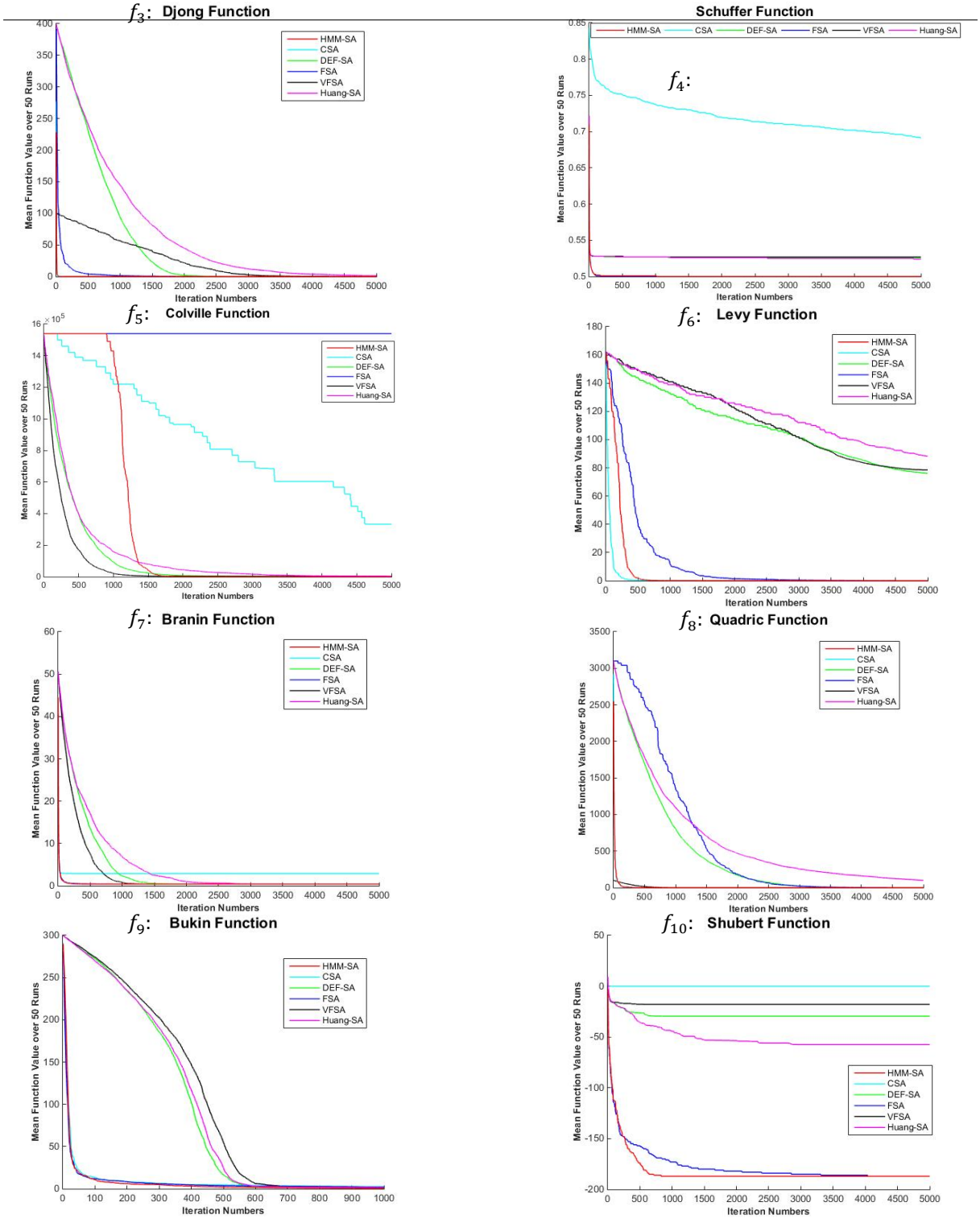


Fig. 2 Comparison of convergence performance of HMM-SA and SA variant's

In the most cases, our approach gives better results than other variants except for functions f_3 and f_8. Classical simulated annealing give a better solution over 50 runs for function f_3, but we noticed that HMM-SA gives better solution if iteration numbers is limited to 2500. Observed from Fig. 3 we can find that the mean function values over 50 runs of HMM-SA converges to a better solutions. There are certain stages that others SA variant's outperforms HMM-SA for many functions' plots. For example, CSA's convergence speed is faster than HMM-SA for function f_6 at the early, the same for function f_5 where convergence speed of VFSA, DEF-SA and Huang-SA exceeds HMM-SA. However, the convergence speed of HMM-SA is faster than others variants at the initial stage for functions $f_2, f_3, f_4, f_7, f_8, f_9, f_{10}$.

5 Conclusion and future research

In this study, we proposed a self-tuning capability of simulated annealing based on Hidden Markov Model. To test the performance of this approach, it were applied to a number of benchmark function selected from literature. This approach allows to controls the cooling of SA during the run, based on sequence of state generated from a set of rules. The HMM parameters are calculated and updated at each cooling step. The Viterbi algorithm is then used to classify the observed sequence as an exploration or exploitation or an escape from local minimum.

The comparisons of the proposed approach and the other variants of simulated annealing demonstrate that the simulated annealing based on HMM classifier is able to find better solutions in reasonable time. Our approach is able to manage time by rapidly decreasing temperature and thus anticipating exploitation state, this lead to a better convergence.

Future research may focus on HMM with continuous observations. This may enhance the performances of HMM classifier. Furthermore, our approach may be compared to SA with fuzzy logic controllers and the application of our method to some optimization problems should be pursued.

References

[1] A. Casotto, F. Romeo, and A. Sangiovanni-Vincentelli (1986). A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells. Proceedings of the IEEE International Conference on Computer-Aided Design, 30-33.

[2] B. Hajek (1988). Cooling schedules for optimal annealing. Mathematical Operations Research, 13(2), 311–329.

[3] C. Elkan (1993). The Paradoxical Success of Fuzzy Logic, AAAI-93 Proceedings,7.

[4] D. Abramson, M. Krishnamoorthy, H. Dang (1996). Simulated Annealing Cooling Schedules for the School Timetabling Problem, submitted to Asia Pacific Journal of Operation Research

[5] E. Aarts, and J. Korst (1989). Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing. John Wiley & Sons.

[6] E. Aarts, F. de Bont, E. Habers, and P. van Laarhoven (1986). Parallel Implementations of the Statistical Cooling Algorithm. Integration, the VLSI Journal, Vol. 4, 209-238.

[7] E. Triki , Y.Collette, P. Siarry (2005). A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. Eur. J. Journal of Operation Research 166, 77–92.

[8] H. A. E Oliveira Jr. and A. Petraglia (2013). Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing. Applied Soft Computing Journal, Vol. 13, no. 11, pp. 4349–4357.

[9] H.H. Szu, R.L. Hartley (1987). Fast simulated annealing, Phys. Lett. A, 122, 157–162.

[10] L. Ingber (1989). Very fast simulated re-annealing. Mathematical and Computer Modelling, 12(8), 967–973.

[11] L. Rabiner (1989). A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257–286.

[12] M. Huang, F. Romeo, and A. Sangiovanni-Vincentelli (1986). An Efficient General Cooling Schedule for Simulated Annealing. In Proceedings of the International Conference on Computer Aided Design, p. 381.

[13] P. Banerjee, and M. Jones (1986). A Parallel Simulated Annealing Algorithm for Standard Cell Placement on a Hypercube Computer. Proceedings of the IEEE International Conference on Computer-Aided Design, 34-37.

[14] P. Kouvelis, W.-C. Chiang, J.A. Fitzsimmons (1992). Simulated Annealing Procedures for Machine Layout Problems in the Presence of Zoning Constraints. Eur J Oper Res 57(22), 203–223.

[15] P.N. Strenski and S. Kirkpatrick (1991), Analysis of finite length annealing schedules. Algorithmica, 6(1-6), 346-366.

[16] R. Battiti, M. Brunato (2008). Reactive Search and Intelligent Optimization, Computer Science Interfaces, Springer.

[17] R. Bellio , S. Ceschia (2016), Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem, Journal of Computers & Operations Research 65, 83–92.

[18] S. Geman, and D. Geman (1984). Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images, IEEE Trans. Pattern Analysis and Machine Intelligence. 6, 721-741.

[19] S. Kirkpatrik (1983). Optimization by simulated annealing. Science, Vol. 220, No. 4598, 671–680.

[20] S. Sun, F. Zhuge, S. A. Napel (2008). Learning enhanced simulated annealing, Applied Intelligence, 2008, Vol. 28, Issue 1, pp 83-99.

[21] S.J. Jeong and K-S. Kim (2009). The efficient search method of simulated annealing using fuzzy logic controller. Expert Systems with Applications 36, 70099-7103.

[22] S.Yang, J. Marcio Machado (2000). A Self-Learning Simulated Annealing Algorithm for Global Optimizations of Electromagnetic Devices, IEEE Transactions On Magnetics, Vol. 36, N°. 4.

Hidden Markov Model classifier for the adaptive ACS-TSP pheromone parameters

Safae Bouzbita[1], Abdellatif El Afia[2], and Rdouan Faizi[3]

[1] ENSIAS - Mohammed V University , Rabat, Morocco
safae.bouzbita@gmail.com
[2] ENSIAS - Mohammed V University , Rabat, Morocco
a.elafia@um5s.net.ma
[3] ENSIAS - Mohammed V University , Rabat, Morocco
rdfaizi@gmail.com

Abstract

The purpose of this work is to enhance the performance of Ant Colony System algorithm applied to the Travelling Salesman Problem (ACS-TSP) by varying dynamically both the local and global pheromone decay parameters based on the Hidden Markov Model algorithm, using two state indicators: Diversity and Iteration that reflect the state of research space in a given moment. The proposed method was tested on several TSP benchmark instances, which compared with the basic ACS, the combination of Fuzzy Logic Controller (FLC) and ACS to prove the efficiency of its performance.

1 Introduction

The Ant Colony System (ACS) is one of the powerful variants of the original Ant System introduced by Dorigo and Gambardella (1997) [1–3], that characterized by the addition of local pheromone updating rule to shuffle the ants' tours by changing dynamically the amount of pheromone $\tau(r, s)$ on the visited edges (r, s). In this way, edges become slightly less desirable and the search becomes more diversified [4–7]. In fact, edges are initialized with an initial amount of pheromone τ_0, then every time an ant k chooses an edge (r_k, s_k), it applies the local pheromone updating rule using the local pheromone decay parameter $\xi \in [0, 1]$ according to this equation:

$$\tau(r_k, s_k) := (1 - \xi)\tau(r_k, s_k) + \xi\tau_0 \tag{1}$$

To make the search more directed, ACS algorithm perform the global pheromone updating rule to the globally best ant's tour L_{best}, that is to say only the best ant k is allowed to modify the amount of pheromone. Global pheromone updating is performed using the global pheromone parameter $\rho \in [0, 1]$ after all ants have terminated their tours according to the following equation:

$$\tau(r_k, s_k) := (1 - \rho)\tau(r_k, s_k) + \frac{\rho}{L_{best}} \tag{2}$$

As it is obvious, the performance of ACS depends strongly to the pheromone information. In most applications of ACS algorithm, the ξ and ρ parameters which are corresponding of the pheromone updating rules are set constant throughout the run of the algorithm. However, adapting ACS parameters can improve the performance of the algorithm at computational time. In this work, instead of using constant values of ξ and ρ parameters, we performed a dynamic adaptation using the Hidden Markov Model (HMM) algorithm.
The remainder of this paper is organized as follows: First, we summarize the related work, then we present the proposed method in section 3, in section 4 we discuss the results and experiments on the Travelling Salesman Problem (TSP). Finally, in section 5 conclusions and future work are presented.

2 Related Work

In the last years, adapting the ACS parameters become a hot topic. Many studies have been conducted to propose improved ACS algorithm by adapting one or several parameters. In this section we review the adaptation of pheromone decay parameters ξ and ρ chronologically.

Pilat and White [8] used a genetic algorithm to evolve the values of ρ, β and q_0. The main idea consist on initializing the algorithm with 4 ants, where each ant is characterized by its own parameters' values, then the two ants with the best fitness are chosen to produce two children by applying the genetic algorithm, after that the two worst ants' parameters replaced by the two produced.

Randall [9] proposed a self-adaptation method using the following equation $p_i = l_i + \frac{w_i}{P}(u_i - l_i)$ to identify suitable values for ρ, β, q_0 and ξ for the TSP and the Quadratic Assignment Problems(QAP). Where, P is a constant the defines the granularity of parameter values, p_i is the value of the i^{th} parameter, l_i is the lower bound value of the i^{th} parameter, w_i is the discretized division chosen for the i^{th} parameter and u_i is the upper bound value of the i^{th} parameter.

Gaertner and Clark in [10] proposed a genetic algorithm for automatically determining the optimal combination of β, ρ and q_0 parameters for a given TSP problem. In this approach the ants communicate with the environment using the genetic process to replace the old population of ants with the new one.

Hao et al. [11] used a particle swarm optimization (PSO) approach for dynamically adapting ρ, β and q_0 parameters, in which each ant is characterized by its own parameter setting and the particle position X_k presents the parameters ρ_k, β_k and q_{0k} of the ant k. If the particle's best solution of the current iteration is better than its previous best solution replace the best previous by the current one.

Ling and Luo [12] proposed the use of an Artificial Fish Swarm Algorithm (AFSA) for adapting ρ, α and Q. As in the proposed method of Hao et al [11]. the state X_k of an Artificial Fish presents the parameters ρ_k, α_k and Q_k of an ant k of the colony, then the adaptation of parameters is done according to AFSA's strategies and formulas by replacing the previous global solution with the current best found one. The main difference between this work and the Hao's [11] is that Ling and Luo use the same parameter setting for all ants.

Gomez et al [13] proposed a PSO method to perform the search over a wide range of available sets of parameters ξ,ρ, α, β, q_0 and the number of ants m. The PSO was used to compute the best fitness of ACS algorithm and return the set of parameters according to this fitness.

Hao et al. [14] presented a tuning method for the ρ parameter depending on the quality of solution built by an artificial ant, according to this equation: $\rho_m = \frac{L_m^{-1}}{(L_m^{-1} + P_m^{-1})}$. Where, L_m is the length of the tour found by the ant m, L_p is the length of the tour constructed based on the pheromone matrix.

Cai and Huang in [15] proposed an automatic parameters configuration to adapt the parameter β and ρ, by using two adaptive strategies. First, a new transition rule with adaptive weight parameter is set, to adjust the relative weight of pheromone trail and heuristic value. Second, a tuning rule for ρ parameter is ran based on the quality of the solution generated by artificial ants.

Melo et al. [16] proposed a multi-colony ACS algorithm, where various autonomous colonies of ants try to solve the same problem simultaneously. Each colony has its parameter settings for α, β , ρ and q_0. The exchanging of information among the colonies is ensured by a migration mechanism. The proposed algorithm also includes a mutation operator that modifies the parameter settings of the worst colony by the value of the same parameter of the best colony.

Cai et al in [17] proposed an adaptive pheromone decay parameter ρ based on the quality of the solutions constructed by ants, using the same equation proposed by Hao [14], the only difference between the two approaches is the demonstration of the convergence of the proposed appraoch.

Kumar et al In [18] proposed an adaptive pheromone values for the 1-ANT variant according to some suggested theorems that depend on the edges contained in the path of the best solution. Also, they discussed how to select the pheromone parameter ρ using some proposed formulas.

Liu-ai and Wen-qing [19] used a genetic algorithm to adapt a combination of four parameters which are: ρ , α , β, Q by balancing exploration and exploitation abilities of the search space. In other words, the genetic algorithm's fitness was built according to the objective function of the ACS algorithm.

Olivas et al in [20] introduced an approach for adapting the ρ parameter in a dynamical manner in rank-based Ant colony variant, using a Fuzzy Logic system which controls the abilities for the diversification and intensification of the search space.

Unlike the Fuzzy Logic, the genetic algorithm and the PSO, HMM algorithm is a learnable stochastic automate that has the ability to learn from the data produced by the process described by the HMM itself. The HMM is widely used for dealing with ranging and sequences from speech recognition, process monitoring, pattern recognition, and computational biology due to its stochastic modeling.
Thereby, the use of the HMM algorithm in our approach consist on learning from the exploitation and exploration in the search space by two chosen indicators to find good solution in an appropriate time.

3 Proposed method

In this section we describe the dynamical adaptation of the pheromone decay parameters ξ and ρ using the Hidden Markov Model (HMM). To perform this adaptation two HMM alternatives were proposed. The first alternative consists in the Baum-Welch algorithm which is considered as a learning method for the best presentation of HMM parameters. The second alternative is the Viterbi algorithm which is used here as a classification method. The main idea of the proposed method is to find at each iteration of the ACS algorithm the most suitable state that best explains the solution found by ants in which, we increase or decrease the values of ξ and ρ according to the found state. Thus, to determine the values of ξ and ρ several tests were done to find the appropriate values.

3.1 Hidden Markov Model

The Hidden Markov Model (HMM) is a powerful statistical tool for modeling sequences of data that can be characterized by an underlying stochastic process that is hidden producing a sequence of observations.The HMM is defined by five elements: (S, V, A, B, π), where

- $S = \{S_1, ..., S_T\}$ is the set of the hidden states
- $V = \{V_1, ..., V_M\}$ is the set of the observation symbols per state
- $A = [a_{ij}]$ is a matrix of transition probabilities from state S_i at time t to state S_j at time t+1,
- $B = [b_{jk}]$ is the emission matrix of observing a symbol V_k from a state S_i
- $\Pi = [\pi_{i\in\{1,T\}}]$ is the initial probability where π_i is the probability of being in the state S_i.

In this work, we define two Hidden Markov Models corresponding respectively to the pheromone decay parameters ξ and ρ, which are characterized by five hidden states corresponding to the values of evaporation parameters ξ and ρ respectively: High (H), Medium High (MH), Medium (M), Medium Low (ML), Low (L), then

$$S = \{H, MH, M, ML, L\}.$$

And the observation sequence is a combination of two parameters named Iteration and Diversity.

$$Iteration = \frac{Current\,iteration}{Total\,of\,iterations};\ Diversity = \frac{1}{m}\sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n}(x_{ij}(t) - \bar{x}_j(t))^2} \tag{3}$$

Where, Current iteration is the number of fulfilled iterations, total of iteration is the total number of iterations for testing the algorithm, m is the population size, i is the number of the ant, n is the total number of dimensions, j is the number of the dimension, x_{ij} is the j dimension of the ant i, $\bar{x}_j$ is the j dimension of the current best ant of the colony.
The diversity function used to measure the degree of dispersion of the ants with respect to the current best ant of the colony. This measure reflects the degree of exploration among the search space which considers the dimensions of the ants to compute the value of variety. Thus, when the diversity of ants is low, we need to use more exploration to find another solutions by increasing

the value of ρ and decreasing the value of ξ and vice versa.

In this work, we divided each of the two indicators into three equal intervals. These are Low (L), Medium (M) and High (H). The combination of these two indicators gives nine possible observations then we have

$$V = \{LL, LM, LH, ML, MM, MH, HL, HM, HH\}.$$

The initial model for ξ parameter is as follows:
The array of the initial state probabilities is defined by equiprobable values:

$$\Pi = [\pi_1, \pi_2, \pi_3, \pi_4, \pi_5] = [\frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}]$$

We define the state transition matrix and the emission matrix respectively as following:

$$A = (a_{ij}) = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix} \quad B = (b_{jk}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix}$$

The initial model for ρ parameter is as follows:
The array of the initial state probabilities is defined as the initial state probabilities for ξ parameter:

$$\Pi = [\pi_1, \pi_2, \pi_3, \pi_4, \pi_5] = [\frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}]$$

The state transition matrix and the emission matrix are defined respectively as following:

$$A = (a_{ij}) = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0.5 \end{bmatrix} \quad B = (b_{jk}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This model was developed according to the concept, that is, a low value of ξ accelerate the convergence speed, while high value of ξ weak the attraction of short edges. In contrast, a low value of ρ allows a long persistence of pheromone. Thus ants can exploit the best solutions while high values of ρ encourage the exploration of search space by forgetting previously attractive solutions, and focusing on new information. After building the model and setting the parameters, we use the baum-welch algorithm to re-estimate the HMM parameters $\lambda = (A, B, \pi)$ in an appropriate way.

3.2 Learning HMM parameters

At the end of each iteration we perform an Online Learning for HMM parameters by the Baum-Welch algorithm. The Baum-Welch algorithm uses the expectation maximization algorithm to find the maximum likelihood estimate of the HMM parameters given a sequence of observed data.
First, we initialize the model $\lambda = (A, B, \pi)$ with best guess values, in our case the values were concluded from the rule base of FLC, then we execute an iterative process to re-estimate the initial, transition and emission probabilities matrices which mostly fit the observed data set.
At first, Baum-Welch algorithm uses the Forward algorithm to find $\alpha_{t,i}$: the probability of the subsequence (o_1,o_2,...,o_t) ending at time t, in state S_i and P(O/λ): the probability of the occurrence of the observation sequence O given the model λ. The second algorithm used by Baum-Welch is the Backward algorithm which returns $\beta_{t,i}$: the probability of observing (o_{t+1},...,o_T) that starts at time (t+1) and ends at time t at state S_i.
To estimate the new probabilities, Baum-Welch algorithm uses two new matrices: ξ and γ where the coefficients $\xi_t(i, j)$ represent the probability to be at state S_i at time t and moving to state

S_j at time (t+1) given the model λ and the set of observation O. The coefficients $\gamma_{t,i}$ represent the probability to be at state S_i at time t given the model λ and the set of observation O.

Algorithm 1: Baum-Welch

Input: O=o_1,o_2,...,o_T, S={H,MH,M,ML,L}, $\lambda = (A, B, \pi)$
Output: Re-estimated λ
repeat

1 **Forward:**

for *i=1 to 5* **do** $\alpha_{1,i} = \pi_i * b_{i,o_1}$;
for *t=1 to T-1* **do**
for *j=1 to 5* **do** $\alpha_{t+1,j} = \sum_{i=1}^{N} \alpha_{t,i} * a_{ij} * b_{j,o_{t+1}}$;
end
P(O/λ)=$\sum_{i=1}^{3} \alpha_{T,i}$

2 **Backward:**

for *i=1 to 5* **do** $\beta_{T,i}$=1;
for *t=T to 1* **do**
for *i=1 to 5* **do** $\beta_{t,i} = \sum_{j=1}^{5} a_{ij} * \beta_{t+1,j} * b_{j,o_{t+1}}$;
end

3 **Update:**

for *t=1 to T* **do**
for *i=1 to 5* **do**
for *j=1 to 5* **do** $\xi_{t,i,j} = \dfrac{\alpha_{t,i} a_{ij} * b_{j,o_{t+1}} * \beta_{t+1,j}}{P(O/\lambda)}$;
end
$\gamma_{t,i} = \dfrac{\alpha_{t,i} * \beta_{t,i}}{P(O/\lambda)}$
end
$\pi_i = \gamma_{1,i},\ a_{ij} = \dfrac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_{t,i}},\ b_{i,k} = \dfrac{\sum_{t=1 \cap o_t = q_k}^{T} \gamma_{t,i}}{\sum_{t=1}^{T} \gamma_{t,i}}$

until *no increase of P(O/λ) or no more iterations are possible to do*;

3.3 HMM Classifier

The Viterbi algorithm is used to find the most likely explanation or the most likely state that generated a particular sequence of observations by finding a maximum over all possible state sequences. This algorithm has the ability to deal with the entire sequences of hidden states from the beginning of the algorithm till the current iteration,and then make a decision base on the whole history, which makes it advantageous compared to other algorithms that depend only on the information of the current iteration. The main idea of the proposed method is to find at each iteration of the ACS algorithm the most suitable hidden state that best explains the solution found by ants in which, we increase or decrease the values of ξ and ρ according to the found state.

Algorithm 2: Viterbi

Input: O={o_1,o_2,...,o_T}, S, A, B, π
Output: The classified state z_t

1 [A, B, π]=Baum-Welch(O={o_1,o_2,..., o_T},S, A, B, π)

2 **Initialization:**

for *i:=1 to 5* **do** $\alpha_{1,i} = \pi_i * b_{i,o_1}, \quad \phi_{1,i} = 0$;

3 **Recursion:**

for *t:=2 to T, j:=1 to 5* **do** $\alpha_{t,j} = max_{i \in S}[\alpha_{t-1,i} * a_{ij} * b_{j,o_t}]$
$\phi_{t,j} = argmax_{i \in S}[\alpha_{t-1,i} * a_{ij}]$;

4 **Termination:**

$p = max_{i \in S}[\alpha_{T,i}], \quad z_T = argmax_{i \in S}[\alpha_{T,i}]$

5 **Reconstruction:**

for $t = T-1, T-2, ..., 1$, **do** $z_t = \phi_{t+1,z_{t+1}}$;

Result: The classified state z_t

3.4 Proposed algorithm

This section gives more details about the hybridization of the ACS and the HMM algorithms. It is divided into three pseudo-codes: For the first algorithm, after creating an HMM model, we proceed with the combination of Iteration and diversity parameters calculated at the end of each iteration to the Baum-Welch algorithm as observation sequence to best fit the observed data of Hidden Markov Model. The combination of the two indicators that belongs to $V = \{LL, LM, LH, ML, MM, MH, HL, HM, HH\}$ is done according to the following classification:

Algorithm 3: Classification of indicators

Input: {Iteration, Diversity, L_{best} , MaxIter}

Output: {Observation: O}

Initialization:

if $0 < Iteration \leq \frac{MaxIter}{3}$ **then** Iteration = L;

else if $\frac{MaxIter}{3} < Iteration \leq \frac{2 * MaxIter}{3}$ **then** Iteration = M;

else Iteration= H;

if $0 < Diversity \leq \frac{1}{3}$ **then** Diversity= L;

else if $\frac{1}{3} < Diversity \leq \frac{2}{3}$ **then** Diversity= M;

else Diversity= H;

Return O=Iteration Diversity;

According to the observation sequence returned by the Algorithm 2, the following algorithm tries to find the suitable state's path that most explain the obtained observation sequence. In our case we take the last state of the found path as the desired state:

Algorithm 4: Classification of state

Input: {state, S}

Output: {ξ , ρ}

if *state=L* **then** $\xi = \frac{5}{6}$ and $\rho = \frac{1}{6}$;

else if *state=ML* **then** $\xi = \frac{4}{6}$ and $\rho = \frac{2}{6}$;

else if *state=M* **then** $\xi = \frac{3}{6}$ and $\rho = \frac{3}{6}$;

else if *state=MH* **then** $\xi = \frac{2}{6}$ and $\rho = \frac{4}{6}$;

else $\xi = \frac{1}{6}$ and $\rho = \frac{5}{6}$;

To better understand the hybridization of the ACS and the HMMM algorithms we made, we propose the next algorithm which resumes everything we have come to say above. First we run ACS algorithm, where ants build their tours by choosing the next node using the pseudo random-proportional action choice rule. At the end of each iteration we build an HMM using the data obtained from calculating the Iteration and the Diversity indicators. Then, we use the Baum-Welch algorithm to re-estimate the parameters of the proposed HMM. After that, the Viterbi algorithm was used as a classifier method to find the appropriate sequence of state that generated a particular sequence of observation, the concerned state is the last one in the path.

Algorithm 5: The pseudocode of our proposed algorithm

Initialization:

Initialize the ant number m,set parameters α,β, ρ, ξ and q_0,
Initialize pheromone trails with $\tau_0 = 1/L_{NN}$

Construction Solution:

while *termination condition not met* **do**
O:={};

Place ants in starting node randomly

for *t:=1 to Maxiter* **do**
foreach *ant k in the population* **do**
if $q < q_0$ **then**
$s_k = argmax_{s_k \in J_k(r_k)}[\tau(r_k, s_k)]^\alpha [\eta(r_k, s_k)]^\beta$

end

else $p_{rs}^k = \frac{[\tau(r_k, s_k)]^\alpha . [\eta(r_k, s_k)]^\beta}{\sum_{u_k \in J_k(r_k)} [\tau(r_k, u_k)]^\alpha [\eta(r_k, u_k)]^\beta}, \quad if \quad s_k \in J_k(r_k)$

determinate state s_k from wheel roulette algorithm;

$\tau(r_k, s_k) := (1 - \xi)\tau(r_k, s_k) + \xi\tau_0$
end
for *k:=1 to m* **do** Compute L_k;

Compute L_{best}

Compute Iteration and Diversity according to (3)

o_t:= algorithm 3 (Iteration, Diversity, L_{best} , MaxIter);

O= {O, o_t};

state=Viterbi(O, S, A, B, π)

Update ξ and ρ values according to Algorithm 4 (state, S)

end
Update the global best solution;

Global pheromone update:

$$\tau(r_k, s_k) := (1 - \rho)\tau(r_k, s_k) + \frac{\rho}{L_{best}}$$

end
Return L_{best}

4 Experimental results and comparison

To test the performance of the proposed adaptive ACS algorithm, we compared it with the standard ACS algorithm on several TSP instances.

4.1 Experiment setup

According to [21] the best known values of ACS algorithm parameters are $\beta = 2, \rho = 0.1, and\, q_0 = 0.9$. We have also compared it with the Fuzzy Logic Controller with the same initial parameters values to give logical comparisons with the proposed method. The initial position of ants is set randomly on all experiments
The TSP benchmark instances used in this study were chosen from the TSPLIB [22] according to the most common used instances in the literature. The algorithm is developed on MATLAB.
Each result in the figure 2 above is a best found length over 30 runs with 1000 iterations and different population sizes (m=10, 20, 30, n/4, n), where n is the size of the problem. The best results for every problem instance are shown in table 2.

TSP	Number of cities	best known solutions
att48	48	10628
berlin52	52	7542
ch130	130	6110
D198	198	15780
eil51	51	426
eil76	76	538
eil101	101	629
kroA100	100	21282
lin105	105	14379
pr226	226	80369

Table 1. Characteristics of TSP benchmark instances

4.2 Comparison on the solution accuracy

Before moving on, we should mention in passing that we have implemented the HMM into the ACS algorithm in three ways: First, we have modified only the local pheromone decay parameter ξ which is named in the table by "LC". Second, we have modified only the global pheromone decay parameter ρ which is named in the table by "GB". Third, we have modified both parameters in the same computation time which is mentioned in the table by "LC+GB". The same procedure was applied to the integration of fuzzy logic (FL) into the ACS algorithm.
For the solution accuracy, we can observe from the table 2 that instances of small sizes have found the same minimum length. On the other hand, the proposed algorithm attains a better solution accuracy comparing to the basic ACS with a constant set of parameters and the combination of FL and ACS for the big size problems.
For the ACSFL algorithm, we can see from the results that varying the ξ parameter gives better results than varying the ρ parameter or varying both parameters. Also, the adaptation of varying the two parameters in the same time gives better results than varying the global parameter ρ in most of the instances.

TSP	ACS	ACSFL			ACSHMM		
		LC	GB	LC+GB	LC	GB	LC+GB
att48	33523.70	33523.70	33523.70	33523.70	33523.70	33523.70	33523.70
berlin52	7544.36	7544.36	7544.36	7544.36	7544.36	7544.36	7544.36
ch130	6176	6200.34	6224.31	6246.39	6235.33	6187.08	6172.50
D198	16206.75	16096.55	16233.42	16228.65	16118.06	16041.18	16099.27
eil51	431.67	428.98	432.72	431.11	430.60	432.73	428.98
eil76	547.39	547.73	553.61	550.60	549.79	547.54	544.36
eil101	653.92	647.11	651.44	657.83	661.21	643.29	647.83
kroA100	21445.83	21294.39	21519.75	21362.97	21285.44	21285.44	21285.44
lin105	14382.99	14410.19	14416.64	14382.99	14382.99	14382.99	14382.99
pr226	81257.96	80632.94	80691	81355.92	81311.20	80410.82	80398.69

Table 2. Summary of results using ACSHMM algorithm for TSP instances

However, For the ACSHMM algorithm there is a contrast in the results, where we obtain better results when varying only the local parameter ξ or varying both parameters ξ and ρ.
In general, it can be noted that the adaptation of ξ and ρ in the same time for the ACSHMM shows better solutions in most of the instances. Also, the adaptation of ξ and ρ parameters separately outperform the original ACS and the FL system in all its variants.

4.3 Comparison on the convergence speed

In this section, we display some figures obtained from running the basic ACS algorithm and the proposed one with the same parameters, that draws the best found solution in each iteration for each algorithm.
From the figures above, we can observe that the proposed algorithm ACSHMM in the case of varying ξ and ρ simultaneously found better solutions than the basic ACS with faster convergence.
For D198 the basic ACS converged earlier than the proposed algorithm but to a bad solution.
For pr226 we can observe a clear difference between the obtained solutions in both the convergence speed and the soution accuracy.

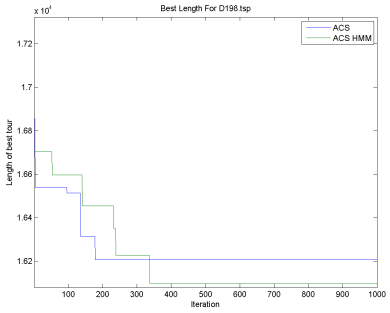


Fig. 1. Sample run on D198.tsp

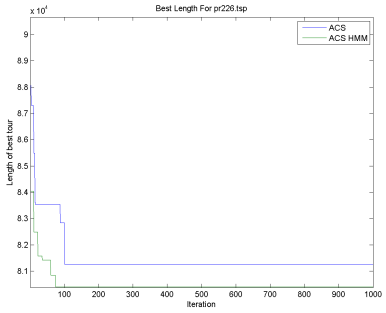


Fig. 2. Sample run on pr226.tsp

4.4 Statistical test

The Z-test was used as a statistical test to compare the methods. And the used parameters for the tests are: the 30 experiments for each instance, the null hypothesis H_0 says that the proposed method returns better results when compared with the other methods and the alternative hypothesis H_1 says that the proposed method returns greater than or equal results on average when compared with the other approaches, the level of significance is 5 percent, and the critical value is $Z_0 = -1.96$, so the rejection region is the set of values lowers than - 1.96. The values in the table 3 are the Z-test values that don't reject the null hypothesis. From the results in table 3 we can assume that our proposed approach can achieve better results with level of significance of 5 percent compared with other methods.

TSP	ACS	ACSFL			ACSHMM		
		LC	GB	LC+GB	LC	GB	LC+GB
att48	0.0089	-0.0038	-0.0120	0.0059	0.0070	-0.0072	0.0096
berlin52	-0.0049	-0.0045	-0.0045	3.7099e-04	-0.0025	-0.0017	-0.0294
ch130	-0.0294	7.1841e-04	-0.0034	3.1709e-04	-0.0037	0.0017	-0.0033
D198	-0.0231	0.0065	-0.0049	0.02	0.0032	0.0023	0.012
eil51	1.4151	0.1147	1.6736	0.5685	-5.7159e-05	6.1038e-05	0.6724
eil76	-3.4981e-06	1.3974e-05	-3.2335e-05	3.7294e-06	-3.7649e-05	9.6346e-06	-1.4947e-05
eil101	-1.6489e-05	-1.5841e-05	-2.4306e-05	-1.9698e-05	-5.2884e-05	1.9537e-05	-1.7094e-05
kroA100	-0.0127	-0.0014	0.0109	0	0.0117	-0.0111	7.6096e-04
lin105	0.0013	0.0173	0.0053	0.0102	0.0108	0.0084	-0.0293
pr226	-0.0033	0.0012	0.0028	-0.0023	-9.2120e-04	0.0045	-0.0029

Table 3. Results of the comparisons using the Z-test

5 Conclusion and Future Work

This paper proposed a new method to control the pheromone parameters ξ and ρ that have a crucial impact on the performance of ACS algorithm using the HMM algorithm. This approach outperforms the basic ACS and FL algorithm in terms of both solution quality and convergence speed. In our future works we specify other indicators that best represent the exploration and exploitation abilities of Ant Colony System search space in order to enhance the control of HMM into ACS.

References

1. Dorigo, Marco, and Luca Maria Gambardella. "Ant colony system: a cooperative learning approach to the traveling salesman problem." Evolutionary Computation, IEEE Transactions on 1.1 (1997): 53-66.
2. Dorigo, Marco, and Christian Blum. "Ant colony optimization theory: A survey." Theoretical computer science 344.2 (2005): 243-278.
3. Dorigo, Marco, and Luca Maria Gambardella. "Ant colonies for the travelling salesman problem." BioSystems 43.2 (1997): 73-81.
4. Erol, Ayşe Hande, Merve Er, and Serol Bulkan. "Optimizing the Ant Colony Optimization Algorithm Using Neural Network for the Traveling Salesman Problem." Actas de la Conferencia Internacional de. 2012.
5. Dorigo, Marco, Mauro Birattari, and Thomas Stützle. "Ant colony optimization." Computational Intelligence Magazine, IEEE 1.4 (2006): 28-39.
6. Dorigo, Marco, and Thomas Stützle. "Ant colony optimization: overview and recent advances." Techreport, IRIDIA, Universite Libre de Bruxelles (2009).
7. Gilmour, Stephen, and Mark Dras. "Understanding the pheromone system within ant colony optimization." AI 2005: Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2005. 786-789.
8. Pilat, Marcin L., and Tony White. "Using genetic algorithms to optimize ACS-TSP." Ant Algorithms. Springer Berlin Heidelberg, 2002. 282-287.
9. M. Randall, "Near Parameter Free Ant Colony Optimisation," Workshop on Ant Algorithms 2004, pp. 374-381.
10. Gaertner, Dorian, and Keith L. Clark. "On Optimal Parameters for Ant Colony Optimization Algorithms." IC-AI. 2005.
11. Hao Z., Cai R., Huang H.: An adaptive parameter control strategy for ACO. In: Proceedings of the International Conference on Machine Learning and Cybernetics, IEEE Press, pp. 203–206 (2006)
12. Ling W., Luo H.: An adaptive parameter control strategy for ant colony optimization. In: CIS'07: Proceedings of the 2007 International Conference on Computational Intelligence and Security, IEEE Computer Society, Washington, DC, pp. 142–146 (2007)
13. Gómez-Cabrero, David, Carmen Armero, and D. Nalin Ranasinghe. "The Travelling Salesman's Problem: A self-adapting PSO-ACS algorithm." Industrial and Information Systems, 2007. ICIIS 2007. International Conference on. IEEE, 2007.
14. Hao, Zhifeng, et al. "An ACO algorithm with adaptive volatility rate of pheromone trail." Computational Science–ICCS 2007. Springer Berlin Heidelberg, 2007. 1167-1170.
15. Cai, Z., and Huang, H. (2008, December). Ant colony optimization algorithm based on adaptive weight and volatility parameters. In Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on (Vol. 2, pp. 75-79). IEEE.
16. Melo, Leonor, Francisco Pereira, and Ernesto Costa. "MC-ANT: a multi-colony ant algorithm." Artifical Evolution. Springer Berlin Heidelberg, 2009. 25-36.
17. Zhaoquan, C. A. I., et al. "Ant Colony Optimization Based on Adaptive Volatility Rate of Pheromone Trail" International Journal of Communications, Network and System Sciences 2.08 (2009): 792.
18. Kumar, Prasanna, and G. S. Raghavendra. "A Note on the Parameter of Evaporation in the Ant Colony Optimization Algorithm." International Mathematical Forum. Vol. 6. No. 34. 2011.
19. Liu-ai, Wu, and Fan Wen-Qing. "A Parameter Model of Genetic Algorithm Regulating Ant Colony Algorithm." e-Business Engineering (ICEBE), 2012 IEEE Ninth International Conference on. IEEE, 2012.
20. F. Olivas, F. Valdez and O. Castillo (2014) "Ant Colony Optimization with parameter adaptation using fuzzy logic for TSP problems", International Seminar on Computational Intelligence 2014.
21. Stützle, T., López-Ibánez, M., Pellegrini, P., Maur, M., de Oca, M. M., Birattari, M., and Dorigo, M. (2012). Parameter adaptation in ant colony optimization. In Autonomous search (pp. 191-215). Springer Berlin Heidelberg.
22. Reinelt, G. "Tsplib discrete and combinatorial optimization." (1995).

Metaheuristics for supply chain and logistics optimization

Location and Selective Routing Problem with Profits for Green Supply Chains

N. Aras[1], M.T. Tekin[1], and D. Aksen[2]

[1] Industrial Eng. Dept., Boğaziçi University, İstanbul, Turkey
arasn@boun.edu.tr, mttekin@hotmail.com
[2] College of Administrative Sciences and Economics, Koç University, İstanbul, Turkey
daksen@ku.edu.tr

1 Introduction

Environmentally conscious manufacturing, waste reduction and product recovery have emerged as alternative means of coping with the significant societal problem of environmental sustainability. One of the key concerns of the companies involved in product recovery is used product acquisition (collection). There are only a few papers which incorporate the core acquisition explicitly in reverse logistics network design problems. An uncapacitated collection center location problem (CCLP) for incentive- and distance-dependent returns is analyzed in [1]. The p-median version of the same CCLP is considered in [2] under the pickup collection policy in which dedicated vehicles depart from a collection center (CC), visit a single customer zone, and bring the collected cores back to the CC. Note that routing decisions are not included in this model. Aras et al. [4] consider an extension of the CCLP in which routing of the vehicles in incorporated into the model without making decisions about CC locations. The firm is not obliged to visit all dealers; vehicles are dispatched to a dealer only if it is profitable to do so. The problem then becomes an extension of the classical multi-depot vehicle routing problem (MDVRP) in which each visit to a dealer is associated with a gross profit and an acquisition price to be paid to take the cores back. Aras et al. [3] extend this last work by relaxing the assumption that a number of uncapacitated collection centers have already been installed. Hence, the investigated problem becomes a variant of the well-known location-routing problem (LRP) in which the acquisition price paid by the firm to take back cores is also a decision variable. The new problem is referred to as the location and selective routing problem with profits (LSRPP). Since the LRP is known to be $\mathcal{NP}$-hard [10], and it is a special case of the LSRPP in which all dealers are visited and there is no acquisition price involved, the LSRPP is also $\mathcal{NP}$-hard.

In the present paper, we continue to study on the LSRPP, but consider both uncapacitated and capacitated CCs. Furthermore, we provide a more effective mathematical model which is based on the flow variables suggested by Gavish-Graves in their single-commodity flow models [6]. These flow variables represent the amount of cores that are carried by a vehicle from dealer i to dealer j. Thus, rather than using Miller-Tucker-Zemlin (MTZ) subtour elimination constraints [9], we write these constraints by the help of flow variables type. Despite the improvements in the mathematical model, large instances cannot be solved effectively, therefore we devise very effective and efficient Tabu search heuristics to solve the LSRPP.

2 Location and Selective Routing Problem with Profits

In this section we present a mathematical programming formulation of the LSRPP. The context in which the problem is briefly given here. A company engaged in reverse logistics wants to locate facilities with limited capacity which will serve as CCs for the cores to be collected from dealers. The locations of the dealers as well as the amount of cores at each dealer are known. Moreover, each dealer has a reservation price (minimum amount of money he would be willing to receive) per core, and (s)he will only return cores if the offered acquisition price is less than this reservation price. It is important to note that the acquisition price offered by the firm is the same for all dealerships since a pricing policy discriminating with respect to dealers would not be perceived as a fair attitude. Each dealer can only be visited by a single vehicle, hence split pickups are not permitted. All vehicles are identical with respect to load capacity and speed. They must start and end their routes at the same CC without visiting any other CC in the route. The objective of the

firm is to maximize its profit by determining the locations of the CCs, the dealers to be visited, the acquisition price offered for each unit of core collected from the dealers, the number of vehicles allocated to each opened CC and the route of each vehicle. The source of the revenue is the cost savings that results from using the components of the cores in remanufacturing like-new products.

We first define the index sets, and then present the parameters and decision variables of the model developed for the LSRPP. LSRPP is then obtained as

Sets:

$\mathcal{ID}$: set of dealers,
$\mathcal{IC}$: set of potential CC locations,
$\mathcal{I}$: set of dealers and potential CC locations (the union is referred to as sites),

Parameters:

a_i : the number of cores at dealer $i \in \mathcal{ID}$,
c_v : unit vehicle operating cost,
c_d : cost per unit distance traveled,
d_{ij} : distance between sites $i \in \mathcal{I}$ and $j \in \mathcal{I}$,
f_k : the fixed cost of opening a CC at site $k \in \mathcal{IC}$,
p_i : reservation price of dealer i per core,
q : vehicle capacity,
r : the revenue from each core,
s_k : capacity of a CC to be opened at site $k \in \mathcal{IC}$,

Decision variables:

Y_{ik} : binary variable, which is one if dealer $i \in \mathcal{ID}$ is assigned to CC $k \in \mathcal{IC}$,
F_{ij} : continuous variable denoting the amount of cores that are carried by a vehicle from site $i \in \mathcal{I}$ to site $j \in \mathcal{I}$,
X_{ij} : binary variable, which is one if site $j \in \mathcal{I}$ is visited after site $i \in \mathcal{I}$,
Z_k : binary variable, which is one if a CC is located as site $k \in \mathcal{IC}$,
W : continuous variable denoting the the acquisition price for each core collected.

$$\max \sum_{i\in\mathcal{ID}} \sum_{k\in\mathcal{IC}} a_i Y_{ik}(r-W) - \sum_{k\in\mathcal{IC}} f_k Z_k - c_v \sum_{k\in\mathcal{IC}} \sum_{i\in\mathcal{ID}} X_{ki} - c_d \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}:j\neq i:} d_{ij} X_{ij} \tag{1}$$

subject to

$$q \sum_{i\in\mathcal{ID}} X_{ki} \geq \sum_{i\in\mathcal{ID}} a_i Y_{ik} \qquad k \in \mathcal{IC} \tag{2}$$

$$\sum_{i\in\mathcal{I}} X_{ij} = \sum_{k\in\mathcal{IC}} Y_{jk} \qquad j \in \mathcal{ID} \tag{3}$$

$$\sum_{i\in\mathcal{I}} X_{ji} = \sum_{k\in\mathcal{IC}} Y_{jk} \qquad j \in \mathcal{ID} \tag{4}$$

$$\sum_{k\in\mathcal{IC}} \sum_{l\in\mathcal{IC}} X_{kl} = 0 \tag{5}$$

$$\sum_{k\in\mathcal{IC}} Y_{ik} \leq 1 \qquad i \in \mathcal{ID} \tag{6}$$

$$X_{ij} + Y_{ik} - Y_{jk} \leq 1 \qquad i \neq j \in \mathcal{ID}, k \in \mathcal{IC} \tag{7}$$

$$X_{ij} + Y_{jk} - Y_{ik} \leq 1 \qquad i \neq j \in \mathcal{ID}, k \in \mathcal{IC} \tag{8}$$

$$\sum_{i} a_i Y_{ik} \leq s_k \qquad k \in \mathcal{IC} \tag{9}$$

$$Y_{ik} \leq Z_k \qquad i \in \mathcal{ID}, k \in \mathcal{IC} \tag{10}$$

$$X_{ik} \leq Y_{ik} \qquad i \in \mathcal{ID}, k \in \mathcal{IC} \tag{11}$$

$$X_{ki} \leq Y_{ik} \qquad i \in \mathcal{ID}, k \in \mathcal{IC} \tag{12}$$

$$W \geq r_i \sum_{k\in\mathcal{IC}} Y_{ik} \qquad i \in ID \tag{13}$$

$$F_{ij} = \sum_{j\in\mathcal{I}:j\neq i} F_{ji} + a_i \sum_{k} Y_{ik} \qquad i \in \mathcal{ID}, j \in \mathcal{I} \tag{14}$$

$$F_{ij} \leq (q - a_j X_{ij}) \qquad i, j \in \mathcal{I}, i \neq j \tag{15}$$

$$F_{ij} \geq a_i X_{ij} \qquad i,j \in \mathcal{I}, i \neq j \qquad (16)$$

$$X_{ij} \in \{0,1\}, F_{ij} \geq 0 \qquad i \in \mathcal{I}, j \in \mathcal{I} \qquad (17)$$

$$Y_{ik}, Z_k \in \{0,1\} \qquad i \in \mathcal{ID}, k \in \mathcal{IC} \qquad (18)$$

$$W \geq 0 \qquad (19)$$

The objective function, which represents the profit from the collection of cores, consists of four terms. The first term is the total revenue, the second one is the total fixed cost of installing and operating CCs, the third one is related to the fixed cost of operating the fleet of vehicles, while the last one stands for the traveling cost of the vehicles. As can be realized, the first term is nonlinear due to the product of Y_{ik} and W variables, which can be linearized by defining an auxiliary variable $U_{ik} = Y_{ik}W$ and introducing the following sets of constraints: $U_{ik} \geq 0, i \in \mathcal{ID}, k \in \mathcal{IC}$ and $U_{ik} \geq W - \max_{j \in \mathcal{ID}} \{p_j\}(1 - Y_{ik}), i \in \mathcal{ID}, k \in \mathcal{IC}$

Constraint (2) makes sure that the total number of vehicles departing from a CC k is sufficient to carry all the collected cores from dealers assigned to this CC. Constraints (3) and (4) a vehicle must visit a dealer assigned to a CC for the collection of cores. Constraint (5) eliminates the possibility of a vehicle to go directly from a CC to another one. Constraint (6) ensures that a dealer is either assigned to one CC only or is not visited for core collection. Constraints (7) and (8) guarantee the assignment of two dealers to the same CC if they are visited consecutively by a vehicle. They also imply that a vehicle cannot travel from one dealer to another whenever these dealers are assigned to different CCs. Constraint (9) implies that cores collected from dealers assigned to the same CC cannot exceed the capacity of the CC. Constraints (10) are used to assign a dealer to an open CC only. Constraints (11) say that if a dealer is visited just before a CC, then that dealer must be assigned to this CC. Constraints (12) are similar in nature to constraints (11). Constraints (13) simply make sure that the acquisition price must be at least equal to the reservation price of the visited dealers. Constraints (14) are the flow balance constraints written for each dealer $i \in \mathcal{ID}$. If dealer i is visited, then the load on the vehicle must increase by the number of cores collected at that dealer. Constraints (15) ensure that if a vehicle goes from a dealer or CC site i to another site j, then the amount of cores F_{ij} carried from i to j cannot be larger than $(q - a_j)$. In other words, the remaining capacity of the vehicle after leaving site i must allow loading of the cores at site j. Since $a_j = 0$ for $j \in \mathcal{IC}$, these constraints simplify to $F_{ij} \leq q$ when site j is a CC. Constraints (16) ensures that if a vehicle travels from site i to site j, then the amount of core transported must be at least equal to the number of cores collected at site i. When location i is a CC, then a trivial nonnegativity constraint is obtained ($F_{ij} \geq 0$). The set of constraints (14)–(16) eliminates the subtours. Constraints (17)–(19) are the binary and non-negativity restrictions on the decision variables.

3 SOLUTION PROCEDURE

Since LSRPP is $\mathcal{NP}$-hard, large instances can only be solved by metaheuristic procedures within reasonable times. Therefore, we develop a tabu search (TS) heuristic for this problem. Two types of neighborhoods are used to explore the solution space: Change-Tour and Inter-Tour-Exchange. The former removes a dealer from its current tour by utilizing α-GENI unstringing procedure and adds that dealer to another tour. The target tour may belong to the same CC or another CC. This neighborhood also tries to add the removed dealer to a CC as a new tour. Note that GENI is a generalized insertion procedure due to [5] and α is a proximity measure developed by [8] to improve the performance of the Lin-Kernighan TSP heuristic. It is a better indicator than the direct distances between nodes while determining the successive nodes in an optimal TSP tour. Inter-Tour-Exchange move selects a chain of dealers from a tour and inserts this chain as a new tour to a CC. The predecessor and the successor of the chain are connected to each other. Similar to the previous neighborhood, this neighborhood considers all CCs. The tabu condition for all neighborhoods are the same, whenever the tour of a dealer is changed, the dealer cannot be removed from its tour during the tabu tenure, unless the aspiration condition is satisfied.

The results obtained on randomly generated test instances shows that the TS heuristic is very effective and efficient. Except for very small instances, TS outperforms CPLEX and GUROBI in finding a better solution.

References

1. Aras, N., D. Aksen, "Locating collection centers for distance and incentive-dependent returns", *International Journal Production Economics*, 11(2), 316–333, 2008.
2. Aras, N., D. Aksen, A.G. Tanuğur, "Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles", *European Journal of Operational Research*, 191(3), 1223–1240, 2008.
3. Aras, N., D. Aksen, M.T. Tekin, "Location and selective routing problem with pricing for the collection of used products, 40th International Conference on Computers and Industrial Engineering (CIE), pp. 1–6, 2010
4. Aras, N., D. Aksen, M.T. Tekin, "Selective multi-depot vehicle routing problem with pricing", *Transportation Research Part C*, 19, 866–884, 2011.
5. Gendreau, M. A. Hertz, G. Laporte. "New insertion and postoptimization procedures for the travelling salesman problem", *Operations Research*, 40, 1086–1094, 1992.
6. Gavish, B., S.C. Graves, "The travelling salesman problem and related problems. In: Working Paper GR-078-78, Operations Research Center, Massachusetts Institute of Technology, 1978.
7. Guide, V.D.R., R. Teunter, L.N. van Wassenhove, "Matching demand and supply to maximize profits from remanufacturing", *Manufacturing and Service Operations Management*, Vol. 5, No. 4, pp. 303–316, 2003.
8. Helsgaun, K. "An effective implementation of the Lin-Kernighan traveling salesman heuristic", *European Journal of Operational Research*, 126, 106–130, 2000.
9. Miller, C.E., A.W. Tucker, R.A. Zemlin. "Integer programming formulation of traveling salesman problems", *Journal of the Association for Computing Machinery* 7(4), 326–329, 1960.
10. Nagy, G., S. Salhi, "Location-routing: issues, models and methods", *European Journal of Operational Research*, 177(2), 649–672, 2007.

Transportation Research Part B 67 (2014) 344–356

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem

John Willmer Escobar [a,c], Rodrigo Linfati [b], Maria G. Baldoquin [a], Paolo Toth [c,*]

[a] Departamento de Ingeniería Civil e Industrial, Pontificia Universidad Javeriana, Cali, Colombia
[b] Departamento de Ingeniería Industrial, Universidad del Bío-Bío, Concepción, Chile
[c] DEI Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi", University of Bologna, Bologna, Italy

ARTICLE INFO

Article history:
Received 5 June 2013
Received in revised form 27 May 2014
Accepted 27 May 2014

Keywords:
Capacitated Location Routing Problem
Granular Tabu Search
Variable Neighborhood Search

ABSTRACT

This paper proposes a new heuristic algorithm for the *Capacitated Location-Routing Problem* (CLRP), called Granular Variable Tabu Neighborhood Search (GVTNS). This heuristic includes a Granular Tabu Search within a Variable Neighborhood Search algorithm. The proposed algorithm is experimentally compared on the benchmark instances from the literature with several of the most effective heuristics proposed for the solution of the CLRP, by taking into account the CPU time and the quality of the solutions obtained. The computational results show that GVTNS is able to obtain good average solutions in short CPU times, and to improve five best known solutions from the literature. The main contribution of this paper is to show a successful new heuristic for the CLRP, combining two known heuristic approaches to improve the global performance of the proposed algorithm for what concerns both the quality of the solutions and the computing times required to find them.

1. Introduction

The *Capacitated Location-Routing Problem* (CLRP) is a strategic problem of the supply chain management. The basic hierarchical structure of the CLRP is a supply chain involving two echelons: depots and customers. The CLRP is an NP-hard problem, since it is a generalization of the two well known NP-hard problems: the *Capacitated Facility Location Problem* (CFLP) and the *Multi Depot Vehicle Routing Problem* (MDVRP). Mathematical formulations for the location-routing problems have been proposed by Perl and Daskin (1985), Laporte et al. (1986), Hansen et al. (1994), Prins et al. (2007). The most effective exact approaches for the CLRP (see e.g. Baldacci et al. (2011), Belenguer et al. (2011), and Contardo et al. (2014a)) are able to solve to proven optimality only small-medium size instances. Several heuristic algorithms have been proposed to solve larger instances of the problem. The most effective ones are shortly mentioned in the following.

A two-phase metaheuristic approach, which iterates between the location and the routing phases has been proposed by Tuzun and Burke (1999). A greedy randomized adaptive search procedure (GRASP) with a path relinking methodology has been proposed by Prins et al. (2006b). The same authors (see Prins et al. (2006a)) developed a memetic algorithm with population management (MA|PM).

Prins et al. (2007) proposed a new iterative two phase method. In the location phase, a Lagrangian relaxation method is used to solve the CFLP by grouping the customers into clusters called "super customers". In the routing phase, a Granular

* Corresponding author. Tel.: +39 051 2093028; fax: +39 051 2093973.
E-mail addresses: johnwillmer.escobar2@unibo.it (J.W. Escobar), rlinfati@ubiobio.cl (R. Linfati), mgulnara@javerianacali.edu.co (M.G. Baldoquin), paolo.toth@unibo.it (P. Toth).

http://dx.doi.org/10.1016/j.trb.2014.05.014
0191-2615/

Tabu Search (GTS) algorithm with one neighborhood has been used to solve the corresponding MDVRP. For more details on the GTS approach see Toth and Vigo (2003) and, for a successful application of the GTS framework for the solution of difficult routing problems, see, e.g., Kirchler and Wolfler Calvo (2013). Cluster based methods for the CLRP were proposed by Barreto et al. (2007).

More recently, Duhamel et al. (2010) developed a hybridized GRASP with an evolutionary local search (ELS) procedure. Yu et al. (2010) proposed a Simulated Annealing (SA) heuristic based on three random neighborhoods. Pirkwieser and Raidl (2010) proposed a Variable Neighborhood Search (VNS) (for further details see Mladenović and Hansen (1997)) coupled with ILP-based very large neighborhood searches to solve the (periodic) location-routing problem. An adaptive large neighborhood algorithm for the Two-Echelon Vehicle Routing Problem (2E-VRP), which is also able to solve the CLRP, has been introduced by Hemmelmayr et al. (2012). A GRASP with an ILP-based metaheuristic and a multiple ant colony optimization method have been proposed by Contardo et al. (2014b), and by Ting and Chen (2013), respectively. Finally, a *Two-Phase Hybrid heuristic* (2-Phase HGTS) has been presented by Escobar et al. (2013). In this hybrid heuristic algorithm, after a Construction phase (first phase), a modified *Granular Tabu Search* (GTS) approach, with different diversification strategies, is applied during the Improvement phase (second phase). In addition, a random perturbation procedure is considered to prevent the algorithm from remaining in a local optimum for a given number of iterations.

In this work, we propose a new heuristic algorithm, and computationally compare it with the most effective heuristics proposed for the CLRP. The new algorithm, called *Granular Variable Tabu Neighborhood Search* (GVTNS), uses a hybrid initialization procedure and the most successful neighborhood structures proposed in the literature for the CLRP. GVTNS exploits the systematic changes of neighborhood structures known for the CLRP and the neighborhood topologies considered in the *Variable Neighborhood Search* (VNS) scheme to guide a trajectory local search procedure according to the GTS approach. The proposed algorithm is a novel metaheuristic approach which combines VNS with GTS techniques for getting good results within short computing times. While a combination between VNS and Tabu Search (TS) has been proposed in the literature (see e.g. Moreno Pérez et al. (2003) and Repoussis et al. (2006)), no attempt has been proposed for combining a GTS technique within a VNS scheme. The basic VNS scheme sometimes meets difficulties to escape from local optima. A possible way to overcome this drawback is to combine the VNS scheme with the GTS approach which has no such difficulties, since infeasible solutions are allowed, and the memory technique prevents cycling, allowing the algorithm to escape from local optima.

The paper is organized as follows. The Capacitated Location Routing Problem (CLRP) is described in Section 2. Section 3 presents the general framework used by the proposed algorithm GVTNS, whose description is given in Section 4. A computational comparative study on benchmark instances from the literature is provided in Section 5. Finally, Section 6 contains concluding remarks.

2. Problem description

The *Capacitated Location-Routing Problem* (CLRP) can be defined as follows: let $G = (V, E)$ be an undirected graph, where V is a set of nodes which is partitioned into a subset $I = \{1, \ldots, m\}$ of potential *depots* and a subset $J = \{1, \ldots, n\}$ of *customers*. Each potential depot $i \in I$ has a capacity w_i and an opening cost o_i. Each customer $j \in J$ has a nonnegative demand d_j which must be fulfilled by a depot. An unlimited set of identical vehicles, each with capacity q and fixed cost f, is available at each depot $i \in I$. Each edge $(i, j) \in E$ has an associated traveling cost c_{ij}. The goal of the CLRP is to determine the depots to be opened and the routes to be performed to fulfill the demand of the customers. Each route must start and finish at the same depot, the global demand of each route must not exceed the vehicle capacity q, and the global demand of the routes assigned to a depot i must not exceed its capacity w_i. The objective function of the CLRP is given by the sum of the costs of the open depots, of the costs of the traveled edges, and of the fixed costs associated with the used vehicles. A multi-objective version of the problem is considered in Perugia et al. (2011).

3. General framework

3.1. Granular search space

The granular search approach, proposed in Toth and Vigo (2003), is based on the utilization of a sparse graph containing the edges incident to the depots, the edges belonging to the best solutions found so far, and the edges whose cost is smaller than a *granularity threshold* $\vartheta = \beta\bar{z}$, where $\bar{z}$ is the average cost of the edges in the best solution found so far, and β is a *sparsification factor* which is dynamically updated during the search. In particular, the search starts by initializing β to a small value β_0. After $N_s \times n$ iterations the value of β is increased to the value β_n, and $N_r \times n$ additional iterations are performed by considering as current solution the best feasible solution found so far. Finally, the *sparsification factor* β is reset to its original value β_0 and the search continues. β_0 , β_n, N_s , and N_r are given parameters.

The main idea of the *granularity* approach is to obtain high quality solutions within short computing times. To evaluate this significant effect, in Section 5 a computational study of algorithm GVTNS is performed, by executing it with and without the granular search approach.

3.2. Neighborhood structures

Algorithm GVTNS allows the solutions to be infeasible with respect to the depot and the vehicle capacities. Given a solution S (feasible or infeasible), we assign to $f(S)$ a value equal to the sum of the setup costs of the open depots, of the traveling costs of the edges belonging to the routes traversed by S, and of the fixed costs of the vehicles used in S. In addition, $p_d(S)$ is a penalty term obtained by multiplying the global over depot capacity of the solution S times a dynamically updated penalty factor α_d, and $p_r(S)$ is a penalty term obtained by multiplying the global over vehicle capacity of the solution S times a dynamically updated penalty factor α_r. In particular, $\alpha_d = \gamma_d \times f(S_0)$ and $\alpha_r = \gamma_r \times f(S_0)$, where S_0 is the initial feasible solution, and the initial values of γ_d and γ_r are given parameters. Let us introduce $F(S) = f(S) + p_d(S) + p_r(S)$ as the objective function for any solution S. Note that for each feasible solution S, we have $F(S) = f(S)$. It is useful to relate the penalty factors to the value of $f(S_0)$ because the order of magnitude of this value substantially changes by considering different instances.

In particular, if no infeasible solutions with respect to the depot capacity have been found over N_{fact} iterations, then the value of γ_d is set to $max\{\gamma_{min}, \gamma_d \times \delta_{red}\}$, where $\delta_{red} < 1$. On the other hand, if no feasible solutions with respect to the depot capacity have been found during N_{fact} iterations, then the value of γ_d is set to $min\{\gamma_{max}, \gamma_d \times \delta_{inc}\}$, where $\delta_{inc} > 1$. A similar procedure is applied to update the value of γ_r. Note that the values of γ_d and γ_r are set within the range $[\gamma_{min}\gamma_{max}]$. $N_{fact}, \delta_{red}, \delta_{inc}, \gamma_{min}$ and γ_{max} are given parameters.

A similar approach was proposed by Gendreau et al. (1994) for the classical *Capacitated Vehicle Routing Problem* (CVRP). In the algorithm proposed in that paper, if the last N_{fact} solutions were all feasible (respectively, infeasible) with respect to the vehicle capacity, then the value of the penalty factor α_r (initially equal to 1) is set to 0.5 α_r (respectively, to $2\alpha_r$). The main difference of our approach with respect to that proposed in Gendreau et al. (1994), is that we use dynamically updated penalty factors related with the objective function value of the initial solution, instead of using penalty factors defined within a fixed interval. We have experimentally seen that the utilization of this dynamic updating procedure within the proposed algorithm generally produces good feasible solutions within short computing times.

Algorithm GVTNS uses *intra-route* moves (performed in the same route) and *inter-route* moves (performed between two routes assigned to the same depot or to different depots) corresponding to the well known neighborhood structures N_k ($k = 1, \ldots, 5$) utilized in Escobar et al. (2013) and described in the following:

- *Insertion.* A customer is removed from its current position and inserted either in a different position of the same route or in a different route (belonging to the same depot or to a different depot).
- *Swap.* Two customers (in the same route or in different routes, belonging to the same depot or to different depots) exchange their position.
- *Two-opt.* We use a modified version of the well-known 2-opt move used for the *Traveling Salesman Problem* (TSP). If the two selected edges are in the same route or in routes belonging to the same depot, the move is equivalent, respectively, to the intra route and inter-route 2-opt moves for the CVRP. Otherwise, i.e. if the two edges are in routes belonging to different depots, an additional move with the edges incident to the depots is performed to ensure that each route starts and finishes at the same depot. Fig. 1. shows an example of this move.

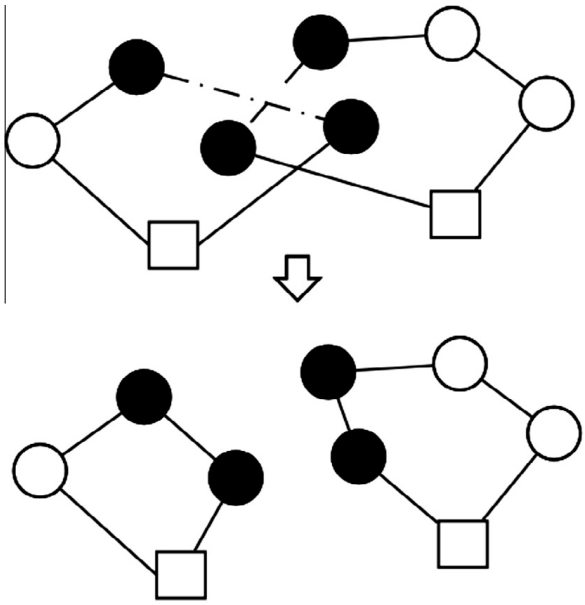


Fig. 1. Example of *Two-opt* move by exchanging edges incident to the depots.

- *Double-insertion*. Two consecutive customers are removed from their current position and inserted in the same route or in a different route (belonging to the same depot or to a different depot) by keeping the edge connecting them.
- *Double-swap*. This move is an extension of the Swap move obtained by considering two pairs of consecutive customers. The edge connecting each pair of customers is kept. The move is performed between pairs of customers in two different routes (belonging to the same depot or to different depots).

A move is performed only if all the new edges inserted in the solution are in the "granular" search space.

3.3. Initial solution procedure

The initial solution S_0 is constructed by using a hybrid heuristic, proposed in Escobar et al. (2013) and based on a cluster approach, which is able to find good initial feasible solutions within short computing times. The following steps are executed:

- *Step 1*. Construct a giant TSP tour containing all the customers, by using the well known Lin–Kernighan heuristic (LKH) (for further details see Lin and Kernighan (1973) and Helsgaun (2000)).
- *Step 2*. Starting from an initial customer j^*, split the giant TSP tour into several clusters composed of consecutive customers so that, for each cluster, the vehicle capacity constraint is satisfied.
- *Step 3*. For each depot i and each cluster g, a TSP tour is obtained by using procedure LKH to evaluate the traveling cost for visiting i and the customers belonging to g.
- *Step 4*. Assign the depots to the clusters by solving an ILP model for the *Single Source Capacitated Facility Location* problem (for further details see Barcelo and Casanovas (1984) and Klincewicz and Luss (1986)).
- *Step 5*. Apply a *Splitting Procedure* to reduce the traveling cost by adding new routes, and by assigning them to different depots. For each depot, the current solution is improved by the VRPH procedure for the CVRP proposed in Groer et al. (2010) and modified by Escobar et al. (2013). Each iteration of the splitting procedure considers a different route having the largest cost edges. This route is split into two subsets of customers by removing the two largest cost edges. The first subset of customers, which is still connected to the depot, replaces the original route. While the second subset of customers, which has been disconnected from the depot, is assigned to a new depot generating a new route. For the new route a TSP tour is obtained by using procedure LKH.

The initial procedure is repeated n times, by considering each customer as initial customer to split the giant tour by following its trajectory, and filling as much as possible all the vehicles. The first initialization procedure based on the construction of a giant tour (route first - cluster second approach) was proposed by Beasley (1983) for the solution of the capacitated VRP. The giant tour is then split into routes by solving a shortest path problem on an appropriate acyclic auxiliary directed graph. Several generalizations of this approach have been proposed in the literature (see, e.g., Vidal et al. (2014) for an effective initialization procedure for the solution of the Multi-Depot Vehicle Fleet Mix Problem). None of these initialization procedures chooses simultaneously the depots to be opened and the subsets of customers to be assigned to each open depot by solving to optimality a small-size ILP model.

4. Description of the Granular Variable Tabu Neighborhood Search heuristic algorithm (GVTNS)

Algorithm GVTNS combines the potentiality of the systematic changes of the neighborhood structures of the Variable Neighborhood Search (VNS) approach, proposed in Mladenović and Hansen (1997) and Hansen and Mladenović (2003), and the efficient Granular Tabu Search (GTS) approach, introduced in Toth and Vigo (2003). In the proposed algorithm, the VNS technique controls the neighborhood changes, while the GTS technique guides the search process by using the neighborhood structures described in Section 3.2, and the efficient search space detailed in Section 3.1.

After constructing the initial solution S_0, the VNS procedure iterates through different neighborhood structures to improve the best feasible solution (S^*) found so far. The algorithm starts by setting $S^* = \overline{S} = \widehat{S} = S_0$, where $\overline{S}$ is the current (feasible or infeasible) solution, and $\widehat{S}$ is the current feasible solution. The following steps are then repeated until a stopping criterion (number of iterations or computing time) is reached:

1. Select a neighborhood from the neighborhoods structures $N_k(k = 1, \ldots, 5)$ described in Section 3.2. The neighborhood structures are explored in a cyclic sequential order (Sequential VNS) by beginning with the neighborhood $N_1 = \textit{Insertion}$. This procedure is proposed in Mladenović and Hansen (1997);
2. Local search: apply a Granular Tabu Search (GTS) procedure in the selected neighborhood $N_k\left(\overline{S}\right)$ until a local minimum S' is found;
3. If $F(S') \leqslant F\left(\overline{S}\right)$, set $\overline{S} := S'$;
4. If S' is feasible and $F(S') \leqslant F\left(\widehat{S}\right)$, set $\widehat{S} := S'$;

5. Every $N_g \times n$ iterations apply the third diversification strategy used by Algorithm 2-Phase HGTS proposed by Escobar et al. (2013), where N_g is a given parameter which controls the number of iterations. This diversification strategy selects the best feasible solution with respect to the depot capacity. Then, for each depot, the customers assigned to the depot are considered, and an initial feasible solution for the corresponding CVRP is built by using the savings procedure proposed in Groer et al. (2010). Then this solution is improved by using the CVRP procedures *vrp_sa, vrp_rtr* proposed in Groer et al. (2010). Note that these procedures are able to obtain, for the considered depot, feasible solutions with respect to the vehicle capacity constraint.

Finally, the best feasible solution found so far S^* is kept. The GTS procedure explores the solution space by moving, at each iteration, from a solution $\overline{S}$ to the best solution S in the neighborhood $N(\overline{S})$. The best possible move is selected by considering the move in $N(\overline{S})$ producing the smallest value of the objective function $F(S)$, and by applying the following tabu aspiration criterion: if the value of the objective function $F(S)$ of the new solution S is not greater than the cost of the best solution found so far, the move producing S is performed even if it corresponds to a *tabu move*. As usual, a move is considered tabu if it tries to reinsert an edge removed in one of the previous moves. The tabu tenure t for each move performed is an integer random value uniformly distributed in the interval $[t_{min}, t_{max}]$, where t_{min} and t_{max} are given parameters.

5. Computational experiments

Algorithm GVTNS is a deterministic algorithm, and, for each instance, a single run, performing $7n$ iterations, has been executed. The implementation details and the results are discussed in the following subsections.

5.1. Implementation details

Algorithm GVTNS has been implemented in C++, and the computational experiments have been performed on an Intel Core 2 Duo (only one core is used) CPU (2.00 GHz) under Linux Ubuntu 11.04 with 2 GB of memory. The algorithm has been evaluated by considering 79 benchmark instances from the literature. The complete set of instances considers three data subsets proposed by Tuzun and Burke (1999), Prins et al. (2004) (called "Prodhon Instances" in the following), and Barreto (2004). In all the subsets, the customers and the depots are represented by points in the plane. Consequently, the traveling cost of an edge is the Euclidean distance, multiplied by 100 and rounded up to the next integer (Prins et al., 2004), or calculated as a double-precision real number (Tuzun and Burke (1999) and Barreto (2004)).

The first data subset was proposed by Tuzun and Burke (1999), and contains 36 instances with uncapacitated depots. The number of customers is $n = 100, 150$ and 200. The number of potential depots is either 10 or 20. The vehicle capacity is set to 150. The second data subset (containing the Prodhon instances) was introduced by Prins et al. (2004), and considers 30 instances with capacity constraints on routes and depots. The number of customers is $n = 20, 50, 100$ and 200. The number of potential depots is either 5 or 10. The vehicle capacity is either 70 or 150. Finally, the third data subset is proposed by Barreto (2004), and considers 13 instances obtained by modifying some classical CVRP instances by adding new depots with capacities and fixed costs. The number of customers ranges from 21 to 150, and the number of potential depots from 5 to 10.

5.2. Comparison of algorithms

We compare the performance of algorithm GVTNS, described in Section 4, with some of the most effective heuristic algorithms proposed in the literature for the solution of the CLRP: GRASP + ELS of Duhamel et al. (2010), SALRP of Yu et al. (2010), VNS + ILP of Pirkwieser and Raidl (2010), ALNS of Hemmelmayr et al. (2012), GRASP + ILP of Contardo et al. (2014b), 2-Phase HGTS of Escobar et al. (2013), and MACO of Ting and Chen (2013).

In Tables 1–5, the following notation is used:

Table 1
Summarized results of Gap PBKS, NBKS and NIBS for algorithm GVTNS with and without the "granular" search approach.

Set	Size	Initial solution Average Gap PBKS	GVTNS with the granular approach Average Gap PBKS	GVTNS without the granular approach Average Gap PBKS	Average CPU Time
Tuzun-Burke	36	1.88	0.69	1.01	592
Prodhon	30	1.46	0.32	0.76	265
Barreto	13	2.04	0.66	1.11	160
Total	79				
Global Avg.		**1.75**	**0.55**	**0.93**	**397**
NBKS		**0**	**28**	**15**	
NIBS		**0**	**5**	**1**	

Table 2
Summarized best results for all the algorithms on the complete data set.

Set	Size	GRASP + ELS		SALRP		VNS + ILP			ALNS - 500 K			ALNS - 5000 K		GRASP + ILP			2-Phase HGTS		MACO		GVTNS	
		Gap Best PBKS	CPU time	Gap PBKS	CPU time	Gap Best PBKS	Gap Avg. PBKS	CPU time*	Gap Best PBKS	Gap Avg. PBKS	CPU time	Gap Best PBKS	CPU time	Gap Best PBKS	Gap Avg. PBKS	CPU time**	Gap PBKS	CPU time	Gap Best PBKS	CPU time**	Gap PBKS	CPU time
Tuzun–Burke	36	1.23	607	1.42	826	-	-	-	0.37	0.83	830	0.11	8103	0.11	0.49	2590	1.08	392	1.17	202	0.69	201
Prodhon	30	1.06	258	0.41	422	0.08	0.90	7	0.39	0.67	451	0.21	4221	0.07	0.27	1163	0.52	176	0.35	191	0.32	91
Barreto	13	0.07	188	0.29	161	-	-	-	0.15	0.24	177	0.05	1772	0.13	0.62	264	0.78	105	0.06	49	0.66	53
Total halfline	79																					
Global Avg.		**0.97**	**405**	**0.85**	**564**	-	-	-	**0.34**	**0.67**	**579**	**0.14**	**5587**	**0.10**	**0.43**	**1665**	**0.81**	**263**	**0.67**	**173**	**0.55**	**135**
Total NBKS		**28**		**24**		**16**	**2**		**29**	**17**		**53**		**39**	**15**		**22**		**26**		**28**	
Total NIBS		**0**		**0**		**2**	**0**		**4**	**0**		**16**		**16**	**0**		**1**		**0**		**5**	
CPU		**Core 2 Quad (2.83 Ghz)**		**Core 2 Quad (2.66 Ghz)**		**Core2 Quad Q9550 (2.83 Ghz)**			**AMD Opteron 275 (2.20 Ghz)**			**AMD Opteron 275 (2.20 Ghz)**		**Intel Xeon E5462 (3.00 Ghz)**			**Core 2 Duo (2.00 Ghz)**		**Athlon XP 2500+ (1.83 Ghz)**		**Core 2 Duo (2.00 Ghz)**	
CPU index		**4373**		**4046**		**4049**			**1234**			**1234**		**9586**			**1398**		**374**		**1398**	

* For each instance: average CPU time over 30 runs.
** For each instance: average CPU time over 10 runs.

Table 3
Best results for all algorithms on Tuzun-Burke Instances.

Instance	PBKS	GRASP+ELS			SALRP			ALNS - 500K					ALNS - 5000K			GRASP+ILP					2-Phase HGTS			MACO			GVTNS		
		Best Cost	Gap Best PBKS	CPU time	Cost	Gap PBKS	CPU time	Best Cost	Gap Best PBKS	Avg. Cost	Gap Avg. PBKS	CPU time	Best Cost	Gap Best PBKS	CPU time	Best Cost	Gap Best PBKS	Avg. Cost	Gap Avg. PBKS	CPU time*	Cost	Gap PBKS	CPU time	Best Cost	Gap Best PBKS	CPU time*	Cost	Gap PBKS	CPU time
111112	**1467.68**	1473.36	0.39	233	1477.24	0.65	369	**1467.68**	0.00	1475.67	0.54	275	**1467.68**	0.00	-	1469.40	0.12	1475.50	0.53	198	1479.21	0.79	152	1489.68	1.50	71	1479.21	0.79	84
111122	**1449.20**	**1449.20**	0.00	9	1470.96	1.50	274	1452.14	0.20	1464.72	1.07	321	**1449.20**	0.00	-	**1449.20**	0.00	1452.00	0.19	580	1486.27	2.56	239	1453.89	0.32	46	1485.28	2.49	126
111212	**1394.80**	1396.59	0.13	112	1408.65	0.99	231	1394.93	0.01	1400.49	0.41	244	**1394.80**	0.00	-	1394.90	0.01	1405.80	0.79	220	1407.26	0.89	120	1407.78	0.93	61	1402.59	0.56	74
111222	**1432.29**	**1432.29**	0.00	114	**1432.29**	0.00	420	1433.42	0.08	1441.21	0.62	376	**1432.29**	0.00	-	1432.30	0.00	1440.60	0.58	755	1474.01	2.91	146	1433.42	0.08	54	1463.23	2.16	99
112112	**1167.16**	**1167.16**	0.00	27	1177.14	0.86	348	1167.53	0.03	1173.04	0.50	489	**1167.16**	0.00	-	1169.10	0.17	1176.20	0.77	278	**1167.16**	0.00	232	1208.04	3.50	80	**1167.16**	0.00	83
112122	**1102.24**	**1102.24**	0.00	259	1110.36	0.74	342	**1102.24**	0.00	1102.34	0.01	373	**1102.24**	0.00	-	1102.40	0.01	1103.60	0.12	634	**1102.24**	0.00	224	**1102.24**	0.00	65	**1102.24**	0.00	105
112212	**791.66**	792.03	0.05	5	**791.66**	0.00	360	**791.66**	0.00	791.83	0.02	739	**791.66**	0.00	-	791.70	0.01	795.80	0.52	227	**791.66**	0.00	201	792.90	0.16	95	**791.66**	0.00	96
112222	**728.30**	**728.30**	0.00	48	731.95	0.50	418	**728.30**	0.00	728.32	0.00	384	**728.30**	0.00	-	**728.30**	0.00	728.50	0.03	550	**728.30**	0.00	254	**728.30**	0.00	65	**728.30**	0.00	126
113112	**1238.49**	1240.39	0.15	55	**1238.49**	0.00	300	1238.70	0.02	1240.31	0.15	357	**1238.49**	0.00	-	**1238.49**	0.00	1239.60	0.09	286	**1238.49**	0.00	160	1265.27	2.16	77	**1238.49**	0.00	82
113122	**1245.30**	1246.00	0.06	233	1247.28	0.16	428	1246.52	0.10	1248.17	0.23	445	1245.31	0.00	-	**1245.30**	0.00	1246.30	0.08	646	1251.22	0.48	237	1256.95	0.94	50	1247.27	0.16	127
113212	**902.26**	902.30	0.00	249	**902.26**	0.00	291	**902.26**	0.00	**902.26**	0.00	321	**902.26**	0.00	-	902.30	0.00	902.80	0.06	231	**902.26**	0.00	135	**902.26**	0.00	61	**902.26**	0.00	71
113222	**1018.29**	**1018.29**	0.00	196	1024.02	0.56	316	**1018.29**	0.00	1018.56	0.03	386	**1018.29**	0.00	-	**1018.29**	0.00	**1018.29**	0.00	749	**1018.29**	0.00	157	**1018.29**	0.00	69	**1018.29**	0.00	85
131112	**1899.90**	1944.57	2.35	518	1953.85	2.84	743	1922.70	1.20	1939.52	2.09	504	1914.41	0.76	-	**1899.90**	0.00	1924.10	1.27	1640	1961.75	3.26	485	1945.43	2.40	227	1933.67	1.78	179
131122	**1823.53**	1864.24	2.23	705	1899.05	4.14	835	1847.93	1.34	1857.29	1.85	635	**1823.53**	0.00	-	1825.30	0.10	1831.00	0.41	3612	1856.51	1.81	298	1853.22	1.63	101	1852.14	1.57	173
131212	**1964.30**	1992.41	1.43	727	2057.53	4.75	456	1975.83	0.59	2009.44	2.30	664	1975.83	0.59	-	**1964.30**	0.00	1969.30	0.25	1275	2012.69	2.46	406	1991.44	1.38	201	1983.09	0.96	184
131222	**1792.80**	1835.25	2.37	415	1801.39	0.48	833	1806.31	0.75	1838.51	2.55	485	1796.45	0.20	-	**1792.80**	0.00	1800.30	0.42	3099	1803.01	0.57	302	1812.34	1.09	141	1803.01	0.57	175
132112	**1444.73**	1453.78	0.63	103	1453.30	0.59	750	1447.43	0.19	1449.15	0.31	1049	1444.73	0.00	-	1446.80	0.14	1450.40	0.39	871	1445.25	0.04	449	1499.05	3.76	206	**1443.32**	−0.10	186
132122	**1434.63**	1444.17	0.66	662	1455.50	1.45	828	1445.32	0.75	1446.91	0.86	805	**1434.63**	0.00	-	1443.90	0.65	1447.20	0.88	2738	1452.07	1.22	493	1446.63	0.84	163	1441.43	0.47	210
132212	**1204.42**	1219.86	1.28	459	1206.24	0.15	752	1204.98	0.05	1205.83	0.12	2197	**1204.42**	0.00	-	1204.80	0.03	1205.90	0.12	2082	**1204.42**	0.00	270	1204.76	0.03	218	**1204.42**	0.00	128
132222	**931.28**	945.81	1.56	224	934.62	0.36	842	931.49	0.02	933.14	0.20	982	**931.28**	0.00	-	931.30	0.00	931.90	0.07	3734	931.49	0.02	335	931.73	0.05	150	**931.28**	0.00	177
133112	**1694.18**	1712.11	1.06	271	1720.81	1.57	742	1694.64	0.03	1700.39	0.37	1046	**1694.18**	0.00	-	1695.90	0.10	1703.80	0.57	938	1705.36	0.66	444	1724.02	1.76	226	1701.34	0.42	182
133122	**1392.01**	1402.94	0.79	524	1415.85	1.71	833	1400.50	0.61	1403.50	0.83	925	**1392.01**	0.00	-	1398.00	0.43	1401.50	0.68	2751	1416.74	1.78	342	1401.05	0.65	123	1416.74	1.78	175
133212	**1198.28**	1214.82	1.38	251	1216.84	1.55	756	1198.67	0.03	1199.27	0.08	1375	**1198.28**	0.00	-	1198.60	0.03	1199.60	0.11	1010	1234.83	3.05	526	1217.29	1.59	241	1213.87	1.30	207
133222	**1151.80**	1155.96	0.36	375	1159.12	0.64	837	1152.01	0.02	1154.36	0.22	911	**1151.80**	0.00	-	1157.30	0.48	1158.70	0.60	3560	1156.05	0.37	380	1158.03	0.54	130	**1151.80**	0.00	208
121112	**2243.40**	2295.90	2.34	655	2324.10	3.60	1328	2265.15	0.97	2278.27	1.55	944	2251.93	0.38	-	**2243.40**	0.00	2251.30	0.35	2805	2265.59	0.99	522	2304.67	2.73	461	2258.02	0.65	315
121122	**2138.40**	2203.57	3.05	432	2258.16	5.60	1455	2183.05	2.09	2192.61	2.54	847	2159.93	1.01	-	**2138.40**	0.00	2154.90	0.77	5680	2166.43	1.31	603	2187.65	2.30	231	2166.20	1.30	300
121212	**2209.30**	2246.39	1.68	1566	2260.30	2.31	1319	2233.55	1.10	2247.75	1.74	907	2220.01	0.48	-	**2209.30**	0.00	2226.10	0.76	3004	2249.40	1.82	527	2231.46	1.00	428	2239.65	1.37	287
121222	**2225.10**	2265.53	1.82	2192	2326.53	4.56	1428	2230.94	0.26	2263.20	1.71	860	2230.94	0.26	-	**2225.10**	0.00	2241.70	0.75	6143	2237.81	0.57	558	2275.70	2.27	234	2236.73	0.52	351
122112	**2073.73**	2106.47	1.58	1521	2112.65	1.88	1320	2082.60	0.43	2093.78	0.97	1606	**2073.73**	0.00	-	2077.80	0.20	2093.80	0.97	3462	2121.93	2.32	522	2098.56	1.20	570	2103.82	1.45	278
122122	**1692.17**	1779.05	5.13	618	1722.99	1.82	1400	1710.67	1.09	1732.00	2.35	941	**1692.17**	0.00	-	1694.80	0.16	1704.40	0.72	8547	1749.10	3.36	691	1711.25	1.13	277	1717.92	1.52	433
122212	**1453.18**	1474.25	1.45	514	1469.10	1.10	1299	1458.55	0.37	1462.15	0.62	1861	**1453.18**	0.00	-	1465.40	0.84	1467.80	1.01	3471	1473.27	1.38	724	1472.93	1.36	544	1469.45	1.12	318
122222	**1082.59**	1085.69	0.29	1243	1088.64	0.56	1429	1085.29	0.25	1086.08	0.32	812	1082.74	0.01	-	1082.90	0.03	1086.00	0.31	5292	1082.59	0.00	616	1087.57	0.46	317	**1082.46**	−0.01	349
123112	**1954.70**	2004.33	2.54	1451	1994.16	2.02	1318	1964.75	0.51	1971.01	0.83	968	1960.30	0.29	-	**1954.70**	0.00	1986.70	1.64	3865	1984.77	1.54	542	1978.74	1.23	387	1969.38	0.75	261
123122	**1926.64**	1964.40	1.96	1273	1932.05	0.28	1412	**1926.64**	0.00	1952.31	1.33	740	**1926.64**	0.00	-	1931.10	0.23	1936.20	0.50	9367	1958.98	1.68	617	1959.71	1.72	230	1935.74	0.47	344
123212	**1762.03**	1778.80	0.95	1398	1779.10	0.97	1314	1762.09	0.00	1764.16	0.12	2055	**1762.03**	0.00	-	1763.10	0.06	1766.20	0.24	3766	1778.41	0.93	697	1782.94	1.19	406	1776.90	0.84	349
123222	**1390.87**	1453.82	4.53	2202	1396.42	0.40	1427	1393.06	0.16	1395.38	0.32	1038	1391.68	0.06	-	1392.00	0.08	1392.70	0.13	5157	**1390.87**	0.00	518	1392.70	0.13	269	1391.50	0.05	317
Average		**1522.01**	**1.23**	**607**	**1526.41**	**1.42**	**826**	**1507.44**	**0.37**	**1515.64**	**0.83**	**830**	**1502.90**	**0.11**	**8103**	**1502.18**	**0.11**	**1508.79**	**0.49**	**2590**	**1519.05**	**1.08**	**392**	**1520.22**	**1.17**	**202**	**1512.50**	**0.69**	**201**
NBKS		**6**			**4**			**7**		**1**			**24**			**13**		**1**			**9**			**4**			**12**		
NIBS		**0**			**0**			**2**		**0**			**12**			**9**		**0**			**1**			**0**			**2**		

* For each instance: average CPU time over 10 runs.

Table 4
Best results for all algorithms on Prodhon Instances.

Instance	PBKS	GRASP + ELS			SALRP			VNS + ILP					ALNS - 500 K					ALNS - 5000 K			GRASP + ILP					2-Phase HGTS			MACO			GVTNS		
		Best Cost	Gap Best PBKS	CPU time	Cost	Gap PBKS	CPU time	Best Cost	Gap Best PBKS	CPU time	Avg. Cost	Gap Avg. PBKS	CPU time*	Best Cost	Gap Best PBKS	Avg. Cost	Gap Avg. PBKS	CPU time	Best Cost	Gap Best PBKS	CPU time	Best Cost	Gap Best PBKS	Avg. Cost	Gap Avg. PBKS	CPU time**	Cost	Gap PBKS	CPU time	Best Cost	Gap Best PBKS	CPU time**	Cost	Gap PBKS	CPU time
20–5-1a	**54793**	**54793**	0.00	0	**54793**	0.00	20	**54793**	0.00	-	54890	0.18	1	**54793**	0.00	**54793**	0.00	39	**54793**	0.00	-	**54793**	0.00	**54793**	0.00	2	**54793**	0.00	3	**54793**	0.00	4	**54793**	0.00	2
20–5-1b	**39104**	**39104**	0.00	0	**39104**	0.00	15	**39104**	0.00	-	**39104**	0.00	1	**39104**	0.00	**39104**	0.00	54	**39104**	0.00	-	**39104**	0.00	**39104**	0.00	3	**39104**	0.00	4	**39104**	0.00	5	**39104**	0.00	3
20–5-2a	**48908**	**48908**	0.00	0	**48908**	0.00	19	**48908**	0.00	-	48934	0.05	1	**48908**	0.00	**48908**	0.00	38	**48908**	0.00	-	**48908**	0.00	**48908**	0.00	1	48945	0.08	3	**48908**	0.00	4	48945	0.08	2
20–5-2b	**37542**	**37542**	0.00	0	**37542**	0.00	15	**37542**	0.00	-	**37542**	0.00	1	**37542**	0.00	**37542**	0.00	67	**37542**	0.00	-	**37542**	0.00	**37542**	0.00	3	**37542**	0.00	4	**37542**	0.00	5	**37542**	0.00	3
50–5-1a	**90111**	**90111**	0.00	3	**90111**	0.00	75	**90111**	0.00	-	90389	0.31	2	**90111**	0.00	**90111**	0.00	101	**90111**	0.00	-	**90111**	0.00	**90111**	0.00	15	90402	0.32	27	**90111**	0.00	25	**90111**	0.00	13
50–5-1b	**63242**	**63242**	0.00	0	**63242**	0.00	58	**63242**	0.00	-	63472	0.36	3	**63242**	0.00	**63242**	0.00	65	**63242**	0.00	-	**63242**	0.00	63281	0.06	18	64073	1.31	27	**63242**	0.00	21	**63242**	0.00	9
50–5-2a	**88298**	88643	0.39	11	**88298**	0.00	95	**88298**	0.00	-	89859	1.77	2	88443	0.16	88576	0.31	99	**88298**	0.00	-	**88298**	0.00	88333	0.04	18	89342	1.18	23	**88298**	0.00	24	89342	1.18	12
50–5-2b	**67308**	**67308**	0.00	16	67340	0.05	59	**67308**	0.00	-	68013	1.05	4	67340	0.05	67448	0.21	200	**67308**	0.00	-	67373	0.10	67436	0.19	22	68479	1.74	21	**67308**	0.00	20	67951	0.96	10
50–5-2bis	**84055**	**84055**	0.00	0	**84055**	0.00	75	**84055**	0.00	-	84208	0.18	3	**84055**	0.00	84119	0.08	107	**84055**	0.00	-	**84055**	0.00	**84055**	0.00	21	**84055**	0.00	23	**84055**	0.00	25	84126	0.08	8
50–5-2bbis	**51822**	**51822**	0.00	11	**51822**	0.00	66	**51822**	0.00	-	52131	0.60	5	**51822**	0.00	51840	0.03	98	**51822**	0.00	-	51883	0.12	51898	0.15	27	52087	0.51	29	**51822**	0.00	17	52213	0.75	9
50–5-3a	**86203**	**86203**	0.00	0	86456	0.29	74	**86203**	0.00	-	86728	0.61	2	**86203**	0.00	86262	0.07	101	**86203**	0.00	-	**86203**	0.00	**86203**	0.00	17	**86203**	0.00	66	**86203**	0.00	33	**86203**	0.00	18
50–5-3b	**61830**	**61830**	0.00	0	62700	1.41	58	**61830**	0.00	-	62905	1.74	3	**61830**	0.00	**61830**	0.00	137	**61830**	0.00	-	**61830**	0.00	61853	0.04	23	**61830**	0.00	38	**61830**	0.00	26	61885	0.09	20
100–5-1a	**275457**	276960	0.55	148	277035	0.57	349	275813	0.13	-	278292	1.03	5	275636	0.06	276364	0.33	520	275524	0.02	-	**275457**	0.00	275628	0.06	220	276186	0.26	157	276220	0.28	117	276137	0.25	75
100–5-1b	**213704**	215854	1.01	68	216002	1.08	269	213973	0.13	-	216286	1.21	8	214735	0.48	215059	0.63	1190	**213704**	0.00	-	214056	0.16	214785	0.51	230	214892	0.56	136	214323	0.29	135	216154	1.15	59
100–5-2a	**193671**	194267	0.31	212	194124	0.23	349	**193671**	0.00	-	195022	0.70	4	193752	0.04	193903	0.12	463	**193671**	0.00	-	193708	0.02	194054	0.20	122	194625	0.49	145	194441	0.40	238	193896	0.12	76
100–5-2b	**157095**	157375	0.18	125	157150	0.04	212	157157	0.04	-	158217	0.71	5	**157095**	0.00	157157	0.04	859	**157095**	0.00	-	157178	0.05	157311	0.14	100	157319	0.14	193	157222	0.08	144	157180	0.05	82
100–5-3a	**200160**	200345	0.09	141	200242	0.04	250	**200160**	0.00	-	201748	0.79	4	200305	0.07	200496	0.17	454	200246	0.04	-	200339	0.09	200394	0.12	97	201086	0.46	163	201038	0.44	179	200777	0.31	69
100–5-3b	**152441**	152528	0.06	221	152467	0.02	197	152466	0.02	-	154917	1.62	4	**152441**	0.00	152900	0.30	684	**152441**	0.00	-	152466	0.02	152814	0.24	100	153663	0.80	168	152722	0.18	152	153435	0.65	68
100–10-1a	**287892**	301418	4.70	48	291043	1.09	270	288540	0.23	-	291775	1.35	9	296877	3.12	299982	4.20	210	292868	1.73	-	287892	0.00	292657	1.66	2622	289755	0.65	277	291134	1.13	105	**287864**	−0.01	203
100–10-1b	**232230**	269594	16.09	186	234210	0.85	203	232230	0.00	-	238059	2.51	10	235849	1.56	240829	3.70	188	233146	0.39	-	234080	0.80	236026	1.63	1067	238002	2.49	152	235348	1.34	82	232599	0.16	117
100–10-2a	**243677**	243778	0.04	260	245813	0.88	261	**243677**	0.00	-	245614	0.79	6	244740	0.44	245548	0.77	136	243829	0.06	-	243695	0.01	243851	0.07	236	245768	0.86	92	245263	0.65	123	245484	0.74	52
100–10-2b	**203988**	**203988**	0.00	139	205312	0.65	199	**203988**	0.00	-	205719	0.85	7	204016	0.01	204494	0.25	261	**203988**	0.00	-	**203988**	0.00	204253	0.13	259	204252	0.13	99	205524	0.75	85	204252	0.13	42
100–10-3a	**250882**	253511	1.05	164	**250882**	0.00	338	251128	0.10	-	255140	1.70	7	253801	1.16	254882	1.59	202	253722	1.13	-	252927	0.82	253610	1.09	723	254716	1.53	125	254302	1.36	113	254558	1.47	82
100–10-3b	**204601**	205087	0.24	203	205009	0.20	240	204706	0.05	-	207410	1.37	7	205609	0.49	206175	0.77	224	**204601**	0.00	-	204664	0.03	205110	0.25	584	205837	0.60	144	204786	0.09	79	205824	0.60	78
200–10-1a	**475327**	486467	2.34	1521	481002	1.19	1428	478349	0.64	-	481142	1.22	18	480883	1.17	483205	1.66	752	478951	0.76	-	**475327**	0.00	477656	0.49	3960	476778	0.31	671	478843	0.74	942	477009	0.35	320
200–10-1b	**377327**	382329	1.33	359	383586	1.66	1336	378631	0.35	-	381516	1.11	16	378961	0.43	380538	0.85	1346	378065	0.20	-	**377327**	0.00	378656	0.35	4006	378289	0.25	476	378351	0.27	562	377716	0.10	239
200–10-2a	**449291**	452276	0.66	112	450848	0.35	1796	449571	0.06	-	452374	0.69	17	450451	0.26	451750	0.55	1201	450377	0.24	-	449291	0.00	449797	0.11	4943	449951	0.15	483	451457	0.48	704	**449006**	−0.06	231
200–10-2b	**374575**	376027	0.39	1610	376674	0.56	1245	375129	0.15	-	376836	0.60	16	374751	0.05	376112	0.41	1349	374751	0.05	-	**374575**	0.00	374996	0.11	3486	374961	0.10	530	374972	0.11	404	374717	0.04	290
200–10-3a	**469870**	478380	1.81	1596	473875	0.85	1776	471024	0.25	-	475345	1.17	18	475373	1.17	479366	2.02	1251	474087	0.90	-	**469870**	0.00	471272	0.30	4075	472321	0.52	624	475155	1.12	879	471978	0.45	330
200–10-3b	**363103**	365166	0.57	591	363701	0.16	1326	363907	0.22	-	365705	0.72	15	366902	1.05	366902	1.05	1137	366416	0.91	-	363103	0.00	363581	0.13	7888	363252	0.04	389	365401	0.63	491	**362827**	−0.08	214
Average		**199630**	**1.06**	**258**	**197778**	**0.41**	**422**	**196911**	**0.08**	**-**	**198643**	**0.90**	**7**	**197852**	**0.39**	**198648**	**0.67**	**451**	**197357**	**0.21**	**4221**	**196776**	**0.07**	**197332**	**0.27**	**1163**	**197617**	**0.52**	**176**	**197657**	**0.35**	**191**	**197229**	**0.32**	**91**
NBKS		**12**			**10**			**16**			**2**			**12**		**7**			**18**			**16**		**7**			**6**			**12**			**9**		
NIBS		**0**			**0**			**2**			**0**			**2**		**0**			**4**			**5**		**0**			**0**			**0**			**3**		

* For each instance: average CPU time over 30 runs.
** For each instance: average CPU time over 10 runs.

Table 5
Best results for all algorithms on Barreto Instances.

Instance	PBKS	GRASP + ELS				SALRP			ALNS - 500 K					ALNS - 5000 K				GRASP + ILP					2-Phase HGTS			MACO			GVTNS		
		Best Cost	Gap Best PBKS	CPU time		Cost	Gap PBKS	CPU time	Best Cost	Gap Best PBKS	Avg. Cost	Gap Avg. PBKS	CPU time	Best Cost	Gap Best PBKS	CPU time		Best Cost	Gap Best PBKS	Avg. Cost	Gap Avg. PBKS	CPU time*	Cost	Gap PBKS	CPU time	Best Cost	Gap Best PBKS	CPU time*	Cost	Gap PBKS	CPU time
Christofides69-50x5	**565.6**	**565.6**	0.00	8		**565.6**	0.00	53	**565.6**	0.00	**565.6**	0.00	73	**565.6**	0.00	-		569.5	0.69	584.9	3.41	18	580.4	2.62	45	**565.6**	0.00	29	580.4	2.62	22
Christofides69-75x10	**844.6**	850.8	0.73	86		848.9	0.51	127	853.5	1.05	854.9	1.22	207	848.9	0.51	-		**844.6**	0.00	847.1	0.30	88	848.9	0.51	94	844.9	0.03	59	853.8	1.09	45
Christofides69-100x10	**833.4**	**833.4**	0.00	127		838.3	0.59	331	**833.4**	0.00	835.4	0.24	403	**833.4**	0.00	-		840.7	0.88	850.0	1.99	492	838.6	0.62	234	836.8	0.40	84	837.1	0.44	111
Daskin95-88x8	**355.8**	**355.8**	0.00	130		**355.8**	0.00	577	**355.8**	0.00	**355.8**	0.00	250	**355.8**	0.00	-		**355.8**	0.00	356.1	0.08	210	362.0	1.74	148	**355.8**	0.00	100	361.6	1.63	97
Daskin95-150x10	**43952.3**	43963.6	0.03	1697		45109.4	2.63	323	44309.0	0.81	44497.2	1.24	613	44004.9	0.12	-		**43952.3**	0.00	44263.5	0.71	1842	44578.9	1.43	456	44131.0	0.41	167	44578.9	1.43	199
Gaskell67-21x5	**424.9**	**424.9**	0.00	0		**424.9**	0.00	18	**424.9**	0.00	**424.9**	0.00	25	**424.9**	0.00	-		**424.9**	0.00	**424.9**	0.00	2	**424.9**	0.00	6	**424.9**	0.00	6	**424.9**	0.00	4
Gaskell67-22x5	**585.1**	**585.1**	0.00	15		**585.1**	0.00	17	**585.1**	0.00	**585.1**	0.00	21	**585.1**	0.00	-		**585.1**	0.00	**585.1**	0.00	3	**585.1**	0.00	9	**585.1**	0.00	5	**585.1**	0.00	6
Gaskell67-29x5	**512.1**	**512.1**	0.00	9		**512.1**	0.00	24	**512.1**	0.00	**512.1**	0.00	40	**512.1**	0.00	-		**512.1**	0.00	**512.1**	0.00	5	**512.1**	0.00	11	**512.1**	0.00	9	**512.1**	0.00	7
Gaskell67-32x5	**562.2**	**562.2**	0.00	18		**562.2**	0.00	27	**562.2**	0.00	**562.2**	0.00	58	**562.2**	0.00	-		**562.2**	0.00	**562.2**	0.00	6	**562.2**	0.00	40	**562.2**	0.00	13	**562.2**	0.00	20
Gaskell67-32x5	**504.3**	**504.3**	0.00	34		**504.3**	0.00	25	**504.3**	0.00	**504.3**	0.00	55	**504.3**	0.00	-		**504.3**	0.00	**504.3**	0.00	8	**504.3**	0.00	22	**504.3**	0.00	10	**504.3**	0.00	15
Gaskell67-36x5	**460.4**	**460.4**	0.00	0		**460.4**	0.00	32	**460.4**	0.00	**460.4**	0.00	61	**460.4**	0.00	-		**460.4**	0.00	**460.4**	0.00	9	**460.4**	0.00	39	**460.4**	0.00	13	**460.4**	0.00	22
Min92-27x5	**3062.0**	**3062.0**	0.00	35		**3062.0**	0.00	23	**3062.0**	0.00	**3062.0**	0.00	38	**3062.0**	0.00	-		**3062.0**	0.00	**3062.0**	0.00	4	**3062.0**	0.00	11	**3062.0**	0.00	9	**3062.0**	0.00	7
Min92-134x8	**5709.0**	5719.3	0.18	280		**5709.0**	0.00	522	5713.0	0.07	5732.6	0.41	460	**5709.0**	0.00	-		5719.3	0.18	5801.9	1.63	750	5890.6	3.18	252	**5709.0**	0.00	137	5789.0	1.40	134
Average		**4492.27**	**0.07**	**188**		**4579.85**	**0.29**	**161**	**4518.56**	**0.15**	**4534.81**	**0.24**	**177**	**4494.51**	**0.05**	**1772**		**4491.78**	**0.13**	**4524.19**	**0.62**	**264**	**4554.65**	**0.78**	**105**	**4504.16**	**0.06**	**49**	**4547.06**	**0.66**	**53**
NBKS		**10**				**10**			**10**		**9**			**11**				**10**		**7**			**7**			**10**			**7**		
NIBS		**0**				**0**			**0**		**0**			**0**				**2**		**0**			**0**			**0**			**0**		

* For each instance: average CPU time over 10 runs.

Instance	instance name
Cost	solution cost obtained by the corresponding algorithm in one single run
Best Cost	best solution cost found by the corresponding algorithm over the executed runs
Avg. Cost	average solution cost found by the corresponding algorithm over the executed runs
PBKS	cost of the previous best-known solution given by the minimum cost among those found by algorithms GRASP + ELS, SALRP, VNS + ILP, ALNS-500 K, ALNS-5000 K, GRASP + ILP, 2-Phase HGTS, and MACO
BKS	cost of the best known solution = min {PBKS, solution cost found by GVTNS}
NBKS	number of best results (BKS) obtained by the corresponding algorithm
NIBS	number of instances for which the corresponding algorithm is the only one which found BKS
CPU	CPU used by the corresponding algorithm
CPU index	Passmark performance test for the corresponding CPU
CPU time	running time in seconds on the CPU used by the corresponding algorithm;
Gap PBKS	percentage gap of the solution cost found by the corresponding algorithm in one single run with respect to PBKS;
Gap Best PBKS	percentage gap of the best solution cost found by the corresponding algorithm over the executed runs with respect to PBKS;
Gap Avg. PBKS	percentage gap of the average solution cost found by the corresponding algorithm over the executed runs with respect to PBKS

In addition, for each instance, the solution costs which are equal to the corresponding BKS are reported in bold. Whenever the considered algorithm is the only one which found the corresponding BKS value, the reported cost is underlined. Finally, the CPU index of a CPU is given by the Passmark performance test (for further details see (PassMark, 2012)). This is a well known benchmark test focused on CPU and memory performance. A higher value of the CPU index indicates that the corresponding CPU is faster.

5.3. Parameter settings

A suitable set of parameters and stop conditions, whose values are based on extensive computational tests on the benchmark instances, was selected for algorithm GVTNS. Initially, the values of all the parameters were set equal to the corresponding values used in Algorithm 2-Phase HGTS (see Escobar et al. (2013)), which generally obtained, for this latter algorithm, good results. Then, for each parameter in turn (with the other parameters fixed to the corresponding current values), two parameter values, different from the current one C have been considered: one value S, smaller than C with $S = C - \sigma$, and the other one L, larger than C with $L = C + \sigma$, where $\sigma = 0.1$ (initial value of C). For each of these two parameter values, algorithm GVTNS was executed on all the 79 benchmark instances, the value of the associated performance index Gap PBKS (see Section 5.2) was evaluated, and the parameter value (S, C or L) corresponding to the minimum performance index value was selected as the current parameter value. In case the parameter value was changed with respect to its current value, the above step was repeated until no change occurred. We observed that the most critical parameters are $\beta_{\mathbf{o}}$, $\beta_{\mathbf{n}}$, $\gamma_{\mathbf{d}}$, $\gamma_{\mathbf{r}}$: Parameters β_o and β_n control the value of β, i.e. the graph sparsification, and hence the number of moves to be performed and the associated computing time. Larger values of β allow the algorithm to perform several moves before to obtain a local optimum, while smaller values lead to shorter computing times. Parameters γ_d and γ_r are related to the penalization scheme used to deal with infeasible solutions. In general, smaller values of these parameters allow the algorithm to consider more moves (although many of them can lead to infeasible solutions), while larger values produce a more conservative behavior of the algorithm, allowing it to obtain quickly feasible solutions. As a consequence, to obtain a good performance of algorithm GVTNS, for what concerns both the quality of the solution found and the computing time, it is critical to define appropriate values for all the previously mentioned parameters.

The values of the selected parameters are reported in the following:

Parameter	Selected value
β_o	1.80
β_n	2.40
N_s	2.00
N_r	1.00
N_{fact}	10
γ_d	0.0075
γ_r	0.0050

(*continued on next page*)

γ_{min}	$1/F(S_0)$
γ_{max}	0.0400
δ_{red}	0.30
δ_{inc}	2.00
N_g	1.50
t_{min}	3
t_{max}	10

These values have been utilized for the execution of all the considered instances. In addition, the Splitting Procedure (see Step 5 of the initialization procedure described in Section 3.3) is executed for 3 iterations.

5.4. Evaluation of the effect of the granularity

We evaluate the impact of the granularity approach on the performance of algorithm GVTNS by executing it with and without the granularity approach, and by fixing, for each instance, the same maximum CPU time as stopping criterion. In particular, the CPU time for each instance has been defined as the CPU time spent by algorithm GVTNS, using the granularity approach and the parameter values detailed in Sub Section 5.3, to solve the given instance.

The results corresponding to the execution of the algorithm with and without the granular search approach are summarized in Table 1. The results show that the granular search approach significantly improves the performance of algorithm GVTNS.

5.5. Comparison of the most effective algorithms

In Tables 2–5, we compare algorithm GVTNS with the most effective heuristics proposed for the solution of the CLRP, i.e., as previously mentioned, algorithms GRASP + ELS of Duhamel et al. (2010), SALRP of Yu et al. (2010), VNS + ILP of Pirkwieser and Raidl (2010), ALNS of Hemmelmayr et al. (2012), GRASP + ILP of Contardo et al. (2014b), 2-Phase HGTS of Escobar et al. (2013) and MACO of Ting and Chen (2013). In the tables, we report the results as presented in the corresponding papers.

The recent algorithms GRASP + PR of Prins et al. (2007), MA|PM of Prins et al. (2006b) and LRGTS of Prins et al. (2006a) are dominated by algorithm GRASP + ELS, proposed by Duhamel et al. (2010), and the corresponding results are not reported in Tables 2–5.

For each instance, algorithm GRASP + ELS has been executed five times and only the best solution found over the five runs is reported. In addition, it is to note that the CPU time reported for each instance represents the time required to find the best solution within the corresponding run. The results reported for algorithm SALRP correspond to a single run of the algorithm. Algorithm VNS + ILP has been computationally evaluated only on the second data subset (Prodhon instances). The results reported for algorithm VNS + ILP correspond, for each instance, to the average cost found and to the average CPU time over thirty runs, and to the best cost found by considering several versions of the algorithm with thirty runs for each version (the corresponding CPU time has not been reported). For algorithm ALNS, for each instance, the best and the average costs over five runs for 500 K iterations (ALNS – 500 K), as well as the best costs over five runs for 5000 K iterations (ALNS – 5000 K), are reported. The CPU time reported for each instance corresponds to the total running time of the corresponding algorithm (only the global average CPU time for each data subset has been reported for ALNS – 5000 K). Algorithms GRASP + ILP and MACO have been executed for ten runs. The results reported for algorithm GRASP + ILP correspond, for each instance, to the best and to the average costs found, and to the average CPU time over the ten runs. The results reported for algorithm MACO correspond, for each instance, to the best cost found and to the average CPU time over the ten runs. Finally, the results reported for Algorithms 2-Phase HGTS and GVTNS correspond to a single run of the algorithm.

Table 2 shows a summary of the results found by the algorithms on the complete data set, while Tables 3–5 show the detailed results for the three considered data sets. For what concerns a comparison among the reported CPU times, it is necessary to take into account the different speeds of the CPUs used in the computational experiments. In addition, for the algorithms reporting average values of the CPU times, i.e. algorithms GRASP + ILP and MACO which execute ten runs for each instance, the CPU times corresponding to the best found costs should be multiplied times the number of executed runs. As previously mentioned, the CPU times corresponding to the best costs found by algorithm VNS + ILP are not reported in Pirkwieser and Raidl (2010), but they seem to be large.

As shown in Table 2, for what concerns the global average value of Gap PBKS, algorithm GVTNS obtains better results than those obtained by algorithms GRASP + ELS, SALRP, 2-Phase HGTS and MACO. Note that the average and best results on the complete data set are not available for algorithm VNS + ILP, therefore a global comparison with this algorithm cannot be performed. In addition, by considering the global average value of the gaps corresponding to the average costs computed over several runs (Gap Avg. PBKS), Table 2 shows that algorithm GVTNS obtains results better than those obtained (in comparable CPU times) by algorithm ALNS – 500 K, and slightly worse than those obtained (in much larger CPU times) by algorithm GRASP + ILP. The best results on the global average value of Gap Best PBKS are obtained, with very large CPU times, by algo-

rithms GRASP + ILP and ALNS-5000 K. By taking into account the big difference of the corresponding CPU times, it is difficult to make a direct comparison of the quality of the solutions found by algorithm GVTNS with respect to the best results reported for algorithms GRASP + ILP and ALNS-5000 K.

For what concerns the number NBKS of the best known solutions found and the number NIBS of instances for which the corresponding algorithm is the only one which finds the best known solution, algorithms ALNS-5000 K and GRASP + ILP are again the best ones, while algorithms ALNS-500 K (Best solution) and GVTNS have comparable behaviors (although the former algorithm has larger CPU times). Finally, it is to note that algorithm GVTNS is able to find, within short CPU times, 28 best known solutions and to improve the previous best known solution for 5 instances.

As for the global CPU time, algorithm GVTNS is faster than the previous published algorithms which are able to find the best results in terms of average gaps and number of best known solutions. Algorithm VNS + ILP takes smaller average CPU times, but the corresponding average solution quality is clearly worse than that of the proposed method. Algorithm MACO seems to require smaller CPU times than algorithm GVTNS, but since only the average computing times over ten runs are reported for the former algorithm, instead of the complete running times for executing the ten runs, a comparison between the two algorithms may be biased.

6. Concluding remarks

We have proposed a new metaheuristic algorithm, called GVTNS and based on the Granular Tabu Search and the Variable Neighborhood Search approaches, for the solution of the CLRP. The computational experiments emphasize the importance of the granular search approach, by showing that it significantly improves the performance of algorithms GVTNS. We have also compared the performance of algorithm GVTNS with that of the most recent effective published heuristics for the CLRP on a set of benchmarking instances from the literature. The results show the effectiveness of algorithm GVTNS, which is able to improve some best known results within short computing times.

Acknowledgments

This work has been partially supported by MIUR (Ministero Istruzione, Università e Ricerca), Italy, by Pontificia Universidad Javeriana, Cali, Colombia and by Universidad del Bío-Bío, Concepción, Chile. This support is gratefully acknowledged. We also wish to thank three anonymous reviewers for their helpful suggestions.

References

Baldacci, R., Mingozzi, A., Calvo, R., 2011. An exact method for the capacitated location-routing problem. Operations Research 59 (5), 1284–1296.
Barcelo, J., Casanovas, J., 1984. A heuristic lagrangean algorithm for the capacitated plant location problem. European Journal of Operational Research 15 (2), 212–226.
Barreto, S., 2004. Análise e modelização de problemas de localização-distribuição (analysis and modelling of location-routing problems) (Ph.D. thesis), University of Aveiro, pp. 3810–4193.
Barreto, S., Ferreira, C., Paixao, J., Santos, B., 2007. Using clustering analysis in a capacitated location-routing problem. European Journal of Operational Research 179 (3), 968–977.
Beasley, J.E., 1983. Route firstcluster second methods for vehicle routing. Omega 11 (4), 403–408.
Belenguer, J., Benavent, E., Prins, C., Prodhon, C., Wolfler-Calvo, R., 2011. A branch-and-cut method for the capacitated location-routing problem. Computers & Operations Research 38 (6), 931–941.
Contardo, C., Cordeau, J., Gendron, B., 2014a. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. INFORMS Journal on Computing 26 (1), 88–102.
Contardo, C., Cordeau, J., Gendron, B., 2014b. A GRASP + ILP-based metaheuristic for the capacitated location-routing problem. Journal of Heuristics 20 (1), 1–38.
Duhamel, C., Lacomme, P., Prins, C., Prodhon, C., 2010. A GRASP × ELS approach for the capacitated location-routing problem. Computers & Operations Research 37 (11), 1912–1923.
Escobar, J., Linfati, R., Toth, P., 2013. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. Computers & Operations Research 40 (1), 70–79.
Gendreau, M., Hertz, A., Laporte, G., 1994. A tabu search heuristic for the vehicle routing problem. Management Science 40 (10), 1276–1290.
Groer, C., Golden, B., Wasil, E., 2010. A library of local search heuristics for the vehicle routing problem. Mathematical Programming Computation 2 (2), 79–101.
Hansen, P., Hegedahl, B., Hjortkjaer, S., Obel, B., 1994. A heuristic solution to the warehouse location-routing problem. European Journal of Operational Research 76 (1), 111–127.
Hansen, P., Mladenović, N., 2003. Variable neighborhood search. In: Handbook of Metaheuristics. Springer, pp. 145–184.
Helsgaun, K., 2000. An effective implementation of the Lin–Kernighan traveling salesman heuristic. European Journal of Operational Research 126 (1), 106–130.
Hemmelmayr, V., Cordeau, J., Crainic, T., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Computers & Operations Research 39 (12), 3215–3228.
Kirchler, D., Wolfler Calvo, R., 2013. A granular tabu search algorithm for the dial-a-ride problem. Transportation Research Part B: Methodological 56, 120–135.
Klincewicz, J., Luss, H., 1986. A lagrangian relaxation heuristic for capacitated facility location with single-source constraints. Journal of the Operational Research Society 37 (5), 495–500.
Laporte, G., Nobert, Y., Arpin, D., 1986. An exact algorithm for solving a capacitated location-routing problem. Annals of Operations Research 6 (9), 291–310.
Lin, S., Kernighan, B., 1973. An effective heuristic algorithm for the traveling-salesman problem. Operations Research 21 (2), 498–516.
Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Computers & Operations Research 24 (11), 1097–1100.
Moreno Pérez, J., Marcos Moreno-Vega, J., Rodríguez Martín, I., 2003. Variable neighborhood tabu search and its application to the median cycle problem. European Journal of Operational Research 151 (2), 365–378.

PassMark, 2012. PassMark Performance Test. <http://www.passmark.com> (Online; accessed 28-Jun-2012).
Perl, J., Daskin, M., 1985. A warehouse location-routing problem. Transportation Research Part B: Methodological 19 (5), 381–396.
Perugia, A., Moccia, L., Cordeau, J.-F., Laporte, G., 2011. Designing a home-to-work bus service in a metropolitan area. Transportation Research Part B: Methodological 45 (10), 1710–1726.
Pirkwieser, S., Raidl, G., 2010. Variable neighborhood search coupled with ilp-based very large neighborhood searches for the (periodic) location-routing problem. In: Proceedings of the International Conference on Hybrid Metaheuristics, pp. 174–189.
Prins, C., Prodhon, C., Ruiz, A., Soriano, P., Wolfler-Calvo, R., 2007. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. Transportation Science 41 (4), 470–483.
Prins, C., Prodhon, C., Wolfler-Calvo, R., 2004. Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. In: MOSIM (4éme Conférence Francophone de Modélisation et Simluation, Nantes, France, vol. 4. pp. 1115–1122.
Prins, C., Prodhon, C., Wolfler-Calvo, R., 2006a. A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. Lecture Notes in Computer Science 3906, 183–194.
Prins, C., Prodhon, C., Wolfler-Calvo, R., 2006b. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR: A Quarterly Journal of Operations Research 4 (3), 221–238.
Repoussis, P., Paraskevopoulos, D., Tarantilis, C., Ioannou, G., 2006. A reactive greedy randomized variable neighborhood tabu search for the vehicle routing problem with time windows. In: Proceedings of the International Conference on Hybrid Metaheuristics, vol. 4030, pp. 124–138.
Ting, C., Chen, C., 2013. A multiple ant colony optimization algorithm for the capacitated location routing problem. International Journal of Production Economics 141 (1), 34–44.
Toth, P., Vigo, D., 2003. The granular tabu search and its application to the vehicle-routing problem. INFORMS Journal on Computing 15 (4), 333–346.
Tuzun, D., Burke, L., 1999. A two-phase tabu search approach to the location routing problem. European Journal of Operational Research 116 (1), 87–99.
Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2014. Implicit depot assignments and rotations in vehicle routing heuristics. European Journal of Operational Research, http://dx.doi.org/10.1016/j.ejor.2013.12.044..
Yu, V., Lin, S., Lee, W., Ting, C., 2010. A simulated annealing heuristic for the capacitated location routing problem. Computers & Industrial Engineering 58 (2), 288–299.

Heuristic methods to solve a production planning and scheduling problem in bulk ports

Gustavo Campos Menezes[1,2], Geraldo Robson Mateus[1] and Martín Gómez Ravetti[3]

[1] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627 ,CEP 31270-010 ,Belo Horizonte ,MG, Brazil.
gcm@dcc.ufmg,br, mateus@dcc.ufmg.br

[2] Departamento de Eletroeletrônica e Computação, Centro Federal de Educação Tecnológica de Minas Gerais, Avenida Doutor Antônio Chagas Diniz 655 ,CEP 32210-160, Cidade Industrial, Contagem, MG, Brazil.

[3] Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 ,CEP 31270-901 , Belo Horizonte, MG, Brazil.
martin.ravetti@dep.ufmg.br

Abstract. This paper describes a real production planning and scheduling problem found in bulk cargo terminals. In this work, the planning and scheduling problem is defined, a mathematical programming model is proposed, and solutions are obtained through the use of heuristics and a commercial optimization package. Computational results show that the proposed approach is able to produce good solutions for large instances (compatible with real cases) for which optimization packages are not able to provide solutions.

1 Introduction

Along with the growth of international trade have come concerns regarding the meeting of deadlines, cost reduction, and the environment. These factors push producers, customers, port operators, ship owners, and other actors in international trade to find minimal cost solutions. In this scenario, the port terminal is a fundamental part of this process. The optimization of its activities, particularly the reduction of cargo transit time and the reliability of operations, is directly reflected in time savings, reductions of fines, and, consequently, profit increases for the terminal operators and their customers.

The problem investigated in this paper can be defined as follows: lets consider a variety of products arriving at a port terminal (supply), they need to be transferred to meet the demand in a set of pier or to a local storage area. To make this transfer, products need a feasible route of equipment. The planning problem must take decisions regarding when to move the material and where to move it, and the scheduling problem deals with making the planning feasible, that is, determining a route of equipment to be use at each time period. Different routes have different capacities. They may share equipment creating conflicts when used during the same time period. In the remainder of this article, this problem will be called the Product Flow Planning and Scheduling Problem, or $PFPSP$.

The main contributions of this article are related to the use of a hierarchical framework to solve a production planning, stockyard allocation and scheduling problems for the delivery of products. The methodology uses a combination of heuristics and mathematical formulations.

The remainder of this article is structured as follows: Section 2 presents the literature review. Section 3 defines the problem. Section 4 presents some sets and variables used in the mathematical formulation. Section 5 discusses the solution strategy applied. Section 6 is dedicated to computational experiments and the manuscript ends with conclusions.

2 Literature Review

The challenges arising from integrating production and logistic operations promote research focusing on productivity increase, cost reduction, and the generation of value in endeavors. Various studies in this line of research, such as those by ([3], [20], [24], [8] and [21]), stress the need for integrated solutions that take into account the problems associated with the three levels: strategic,

tactical, and operational. These studies in particular address themes associated with the supply chain involving the mine-railway-port system.

The main problem study in this article involves the flow of products between reception, stockyard and ships. The primary contributions from the literature are related to the product flow in bulk cargo terminals (iron ore, coal, grains). The references highlighted below are related to mathematical models and exact and heuristic algorithms for problems in this sector.

Bilgen et. all [4], study the problem of blending and allocating ships for grain transportation. Conradie et. al [12] address the optimization of the flow of products (in this case coal) between mines and factory. Byung [17], study the allocation of products in the stockyard. Barros et. all [1] develop an integer linear programming model for the problem of allocating berths in conjunction with the storage conditions of the stockyard. Boland et. all [7] address the problem of managing coal stockpiles in Australia. In the study, it is necessary to choose which equipment will be used for transporting goods to be piled in the stockyard (preferably near the berth where the ship will be loaded), and how to synchronize the whole process. Singh [24] present a mixed-integer programming model for the problem of planning the capacity expansion of the coal production chain in Australia. Finally, [22] proposes an integrated model for the integrated berth allocation and yard assignment problem in bulk ports.

Although these research works address various important aspects of the challenges found in bulk cargo terminals, we did not find articles investigating the integration of product flow and scheduling routes. Such problematic is very usual and must be solved in several bulk terminals. However, various efforts have been made to solve Planning and Scheduling problems, such as in [13], [23], [9], [14] and [10]. More recently, [19], [11] and [25] also address integrated problems.

3 Problem Description

The infrastructure of a port terminal is directly associated with the type of cargo to be handled. Fundamentally, port terminals can be categorized into container and bulk terminals. Generally speaking, research efforts have been greatest regarding solutions for container terminals.

A port terminal for bulk cargo essentially consists of the following components: piers and berths (where ships dock to load and unload cargo) and one or several storage areas (product stockyards). The connection of the port with the land can be through various modes of transport (railway, highway, pipeline) that facilitate the disposal of products to customers or the receipt of products from suppliers (mines, agricultural producers, power plants) to load onto ships.

The flow of products between ships, stockyards, and the transport system is performed using specific equipment: conveyor belts, trucks, ore stackers, and reclaimers, among others. The availability and capacities of the equipment and the storage limits of the stockyards are factors that define the capacity of the terminal to fulfill the shipping contracts. Ship loading/unloading delays at the terminal (demurrage) or even loading/unloading ahead of schedule (dispatch) result in a fine, for the terminal operator in the event of demurrage or for the ship owner in the event of dispatch

To better understand the planning and scheduling problem consider the following scenario. There is a set of supply nodes or reception subsystem, where products are available for transportation, stockyards and demand nodes (berths). Specialized equipment with predefined capacities is used to transport the products within the network. An equipment route between nodes has a given capacity and handle one product at a time. Figure 1 provides a schematic representation of the problem.

The number of routes is limited and they may share equipment. Thus, if two different products are assigned to routes sharing equipment, these routes must be active at non-overlapping intervals. Figure 2 shows a case where two routes (routes 1 and 2) share the same equipment.

In real cases, various types of equipment can be used for cargo transportation, e.g., car dumpers, conveyor belts, stackers, reclaimers, shiploaders, trains, silos and trucks. Similarly, various products can be handled, e.g., iron ore, coal, fertilizers, corn, soy, and peanuts.

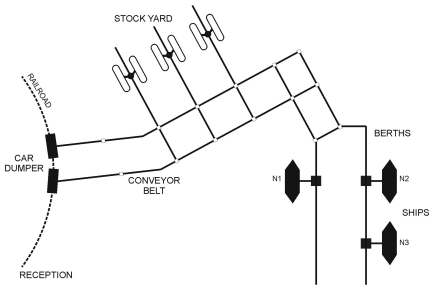


Fig. 1. Reception, stockyards and berths.

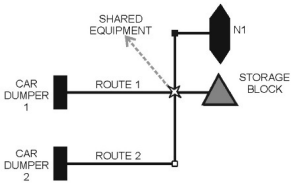


Fig. 2. Routes with shared equipments.

4 PFPSP Formulation

The PFPSP mathematical model was initially proposed in [18]. All production is planned for a given time horizon, divided into T periods. Product supply is related to the arrival of a train, and demand is related to the arrival of a ship. Routes are classified into three types: routes x that transport products from the Reception to the Stockyard, routes y from the Reception to the Ships, and routes z from the Stockyard to the Ships. The main challenges are the allocation of products in the stockyard and the allocation and scheduling of routes that fulfill demands and supplies. The sets, parameters, variables, and equations of the model are described next.

Table 1. Set definition and Input parameters for the PFPSP model

Set	Description	Parameter	Description
T	Set of periods.	a_{pt}	Supply of product p at the beginning of period t.
P	Set of products.	d_{npt}	Demand of product p at a ship moored at berth n at the beginning of period t.
S	Set of storage sub-areas.	K	High-value constant.
R^x	Set of routes (Reception/Stockyard).	l^s_{pt}	Storage capacity of subarea s for product p in period t.
R^x_s	Subset of routes x that reach subarea s.	b^m	Capacity of equipment m (in ton/hour).
R^y	Set of routes (Reception/Piers).	j^m_t	Available time (in hours) for the use of equipment m in period t.
R^z	Set of routes (Stockyard/Pier).	H_t	Total time in period t (in hours).
R^z_s	Subset of routes z from subarea s.		
M	Set of equipment.	c^{rx}	Capacity (in tons/hour) of route $r \in R^x$.
R^x_m	Subset of routes x that use equipment m.	c^{ry}	Capacity (in tons/hour) of route $r \in R^y$.
R^y_m	Subset of routes y that use equipment m.	c^{rz}	Capacity (in tons/hour) of route $r \in R^z$.
R^z_m	Subset of routes z that use equipment m.	β_{npt}	Penalty for not meeting the demand of a ship moored at berth n with product p in period t.
R	Set of all available routes. $R = R^x \cup R^y \cup R^z$.	α_{pt}	Penalty for not meeting the supply at the reception of product p in period t.
N	Set of available mooring berths.	$\gamma^s_{pp't}$	Preparation cost associated with replacing product p by product p' in subarea s at period t.
E	Pairs of routes that share at least one piece of equipment to transport products.	$\lambda_{pp'}$	Cost associated with the loss of income by replacing product p by product p' to meet the demand of product p. When $p = p'$, $\lambda_{pp'} = 0$.

Table 1 defines the sets used for PFPSP modeling and gives the input parameters of the model, which define the capacity limits for periods, storage subareas, equipment, routes, and costs associated with the objective function.

Table 2 shows all decision variables used in PFPSP modeling. These variables are associated with the allocation of storage subareas and the allocation and scheduling of routes and for returning which demands were not met in each period.

Table 2. Variable Definition

Variable	Description
f^s_{pt}	Has unit value when subarea s is allocated for product p in period t.
$Sf^s_{pp't}$	Has a value of 1 when product p has been replaced with product p' at period t. This replacement can occur only when the amount of product p in subarea s has been exhausted in the preceding period $t-1$.
x^r_{pt}	Time taken in period t to transport product p from the reception to the stockyard using route $r \in R^x$
$y^r_{pp't}$, $z^r_{pp't}$	Time taken to transport product p' to meet the demand of product p in period t using route r from sets R^y and R^z. When p' is equal to p, the product delivered is the same as was requested.
e^s_{pt}	Amount of product p stored at subarea s in period t.
IR_{pt}	Represents the amount of product p in the Reception subsystem that was not delivered at the end of period t.
IP_{npt}	Represents the amount of product p that was not delivered at mooring berth n at the end of period t.
$t^r_{pp't}$	Start time for the processing operation using route $r \in R$ in period t. For each variable x^r_{pt}, $y^r_{pp't}$ and $z^r_{pp't}$, there is one start time ($t^r_{pp't}$). When $t^r_{pp't}$ is associated with x^r_{pt}, p and p' are the same product.
$u^r_{pp't}$	Binary variable. It has a value of 1 if the product p' used to meet the demand of product p uses the route r from set R in period t. For all $r \in R^x$, $p = p'$.
$\theta^{rr'}_{pp'\hat{p}\hat{p}'t}$	Binary variable. It has a value of 1 if the product p or p' (used to meet p) precedes the product $\hat{p}$ or $\hat{p}'$ (used to meet $\hat{p}$) in the conflicting routes r,r' $\in E$ in period t.

Follow how the objective function and each constraint of PFPSP are formulated.

The objective function (1) seeks to minimize the penalty of not meeting the supply of products at the Reception subsystem, the penalty of not meeting the demand of ships, the cost of product allocation in the stockyard and the cost of using routes to transport products.

$$\begin{aligned}\min f(z) = &\sum_{p\in P}\sum_{t\in T}\alpha_{pt}IR_{pt} + \sum_{n\in N}\sum_{p\in P}\sum_{t\in T}\beta_{npt}IP_{npt} \\ &+\sum_{s\in S}\sum_{p\in P}\sum_{p'\in(P\cup 0)}\sum_{t\in T}\gamma^s_{pp't}Sf^s_{pp't} + \sum_{p\in P}\sum_{p'\in P}\sum_{t\in T}\lambda_{pp'}(\sum_{r\in R^y}c^{ry}y^r_{pp't} + \sum_{r\in R^z}c^{rz}z^r_{pp't}) \\ &+\sum_{p\in P}\sum_{t\in T}\sum_{r\in R^x}\sigma^r(c^{rx}x^r_{pt}) + \sum_{p\in P}\sum_{p'\in P}\sum_{t\in T}\sum_{r\in R^y}\sigma^r(c^{ry}y^r_{pp't}) + \sum_{p\in P}\sum_{p'\in P}\sum_{t\in T}\sum_{r\in R^z}\sigma^r(c^{rz}z^r_{pp't})\end{aligned} \tag{1}$$

$$\sum_{r\in R^x}c^{rx}x^r_{pt} + \sum_{r\in R^y}c^{ry}(\sum_{p'\in P}y^r_{p'pt}) - IR_{p(t-1)} + IR_{pt} = a_{pt}, \qquad \forall p\in P, \forall t\in T. \tag{2}$$

$$\sum_{r\in R^z}c^{rz}(\sum_{p'\in P}z^r_{pp't}) + \sum_{r\in R^y}c^{ry}(\sum_{p'\in P}y^r_{pp't}) - IP_{np(t-1)} + IP_{npt} = d_{npt}, \qquad \forall n\in N, \forall p\in P, \forall t\in T. \tag{3}$$

$$e^s_{p(t+1)} = e^s_{pt} + \sum_{r\in R^x_s}c^{rx}x^r_{pt} - \sum_{r\in R^z_s}c^{rz}(\sum_{p'\in P}z^r_{p'pt}), \qquad \forall s\in S, \forall p\in P, \forall t\in T. \tag{4}$$

$$e^s_{pt} \le l^s_{pt}, \qquad \forall s\in S, \forall p\in P, \forall t\in T. \tag{5}$$

$$\sum_{p\in P}(\sum_{r\in R^x_m}c^{rx}x^r_{pt} + \sum_{r\in R^z_m}c^{rz}(\sum_{p'\in P}z^r_{pp't}) + \sum_{r\in R^y_m}c^{ry}(\sum_{p'\in P}y^r_{pp't})) \le j^m_t b^m, \qquad \forall m\in M, \forall t\in T. \tag{6}$$

$$\sum_{p\in(P\cup 0)} f^s_{pt} = 1, \qquad \forall s \in S, \forall t \in T. \tag{7}$$

$$Sf^s_{pp't} \geq f^s_{p,(t-1)} + f^s_{p't} - 1, \qquad \forall s \in S, \forall t \in T, \forall p \in P, \forall p' \in (P \cup 0), p \neq p'. \tag{8}$$

$$l^s_{pt}\ f^s_{pt} - e^s_{pt} \geq 0, \qquad \forall s \in S, \forall p \in P, \forall t \in T. \tag{9}$$

$$l^s_{pt}\ f^s_{pt} - \sum_{r\in R^x_s} c^{rx} x^r_{pt} \geq 0, \qquad \forall s \in S, \forall p \in P, \forall t \in T \tag{10}$$

$$x^r_{pt} \leq H_t\ u^r_{pp't}, \qquad \forall p, p' \in P, p = p', \forall t \in T, \forall r \in R^x \tag{11}$$

$$y^r_{pp't} \leq H_t\ u^r_{pp't}, \qquad \forall p, p' \in P, \forall t \in T, \forall r \in R^y \tag{12}$$

$$z^r_{pp't} \leq H_t\ u^r_{pp't}, \qquad \forall p, p' \in P, \forall t \in T, \forall r \in R^z \tag{13}$$

$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geq t^r_{pp't} + x^r_{pt} \tag{14}$$
$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geq t^{r'}_{\hat{p}\hat{p}'t} + x^{r'}_{\hat{p}t} \tag{15}$$
$$\forall (r,r') \in E, (r,r') \in R^x, \forall p = p' \in P, \forall \hat{p} = \hat{p}' \in P, \forall t \in T$$

$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geq t^r_{pp't} + x^r_{pt} \tag{16}$$
$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geq t^{r'}_{\hat{p}\hat{p}'t} + y^{r'}_{\hat{p}\hat{p}'t} \tag{17}$$
$$\forall (r,r') \in E, r \in R^x, r' \in R^y, \forall p = p' \in P, \forall \hat{p}, \hat{p}' \in P, \forall t \in T$$

$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geq t^r_{pp't} + x^r_{pt} \tag{18}$$
$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geq t^{r'}_{\hat{p}\hat{p}'t} + z^{r'}_{\hat{p}\hat{p}'t} \tag{19}$$
$$\forall (r,r') \in E, r \in R^x, r' \in R^z, \forall p = p' \in P, \forall \hat{p}, \hat{p}' \in P, \forall t \in T$$

$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geq t^r_{pp't} + y^r_{pp't} \tag{20}$$
$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geq t^{r'}_{\hat{p}\hat{p}'t} + y^{r'}_{\hat{p}\hat{p}'t} \tag{21}$$
$$\forall (r,r') \in E, (r,r') \in R^y, \forall p, p', \hat{p}, \hat{p}' \in P, \forall t \in T$$

$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geq t^r_{pp't} + z^r_{pp't} \tag{22}$$
$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geq t^{r'}_{\hat{p}\hat{p}'t} + z^{r'}_{\hat{p}\hat{p}'t} \tag{23}$$
$$\forall (r,r') \in E, (r,r') \in R^z, \forall p, p', \hat{p}, \hat{p}' \in P, \forall t \in T$$

$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(1-\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^{r'}_{\hat{p}\hat{p}'t} \geq t^r_{pp't} + z^r_{pp't} \tag{24}$$
$$H_t(1-u^r_{pp't}) + H_t(1-u^{r'}_{\hat{p}\hat{p}'t}) + H_t(\theta^{rr'}_{pp'\hat{p}\hat{p}'t}) + t^r_{pp't} \geq t^{r'}_{\hat{p}\hat{p}'t} + y^{r'}_{\hat{p}\hat{p}'t} \tag{25}$$
$$\forall (r,r') \in E, r \in R^z, r' \in R^y, \forall p, p', \hat{p}, \hat{p}' \in P, \forall t \in T$$

$$t^r_{pp't} + x^r_{pt} \leq \sum_{i=1}^{t} H_i \tag{26}$$

$$t^r_{pp't} \geq \sum_{i=1}^{t-1} H_i \tag{27}$$
$$\forall r \in R^x, \forall p = p' \in P, \forall t \in T$$

$$t^r_{pp't} + y^r_{pp't} \leq \sum_{i=1}^{t} H_i \tag{28}$$

$$t^r_{pp't} \geq \sum_{i=1}^{t-1} H_i \tag{29}$$

$$\forall r \in R^y, \forall p, p' \in P, \forall t \in T$$

$$t^r_{pp't} + z^r_{pp't} \leq \sum_{i=1}^{t} H_i \tag{30}$$

$$t^r_{pp't} \geq \sum_{i=1}^{t-1} H_i \tag{31}$$

$$\forall r \in R^z, \forall p, p' \in P, \forall t \in T$$

$$x^r_{pt} \geq 0, \quad \forall r \in R^x, \forall p \in P, \forall t \in T. \tag{32}$$

$$y^r_{pp't} \geq 0, \quad \forall r \in R^y, \forall p, p' \in P, \forall t \in T. \tag{33}$$

$$z^r_{pp't} \geq 0, \quad \forall r \in R^z, \forall p, p' \in P, \forall t \in T. \tag{34}$$

$$e^s_{pt} \geq 0, \quad \forall s \in S, \forall p \in P, \forall t \in T. \tag{35}$$

$$IR_{pt} \geq 0, \quad \forall p \in P, \forall t \in T. \tag{36}$$

$$IP_{npt} \geq 0, \quad \forall n \in N, \forall p \in P, \forall t \in T. \tag{37}$$

$$f^s_{pt} \in \{0,1\}, \quad \forall s \in S, \forall p \in P, \forall t \in T. \tag{38}$$

$$0 \leq Sf^s_{pp't} \leq 1, \quad \forall s \in S, \forall p \in P, \forall p' \in (P \cup 0), p \neq p', \forall t \in T. \tag{39}$$

$$t^r_{pp't} \geq 0, \quad \forall r \in R, \forall p, p' \in P, \forall t \in T. \tag{40}$$

$$u^r_{pp't} \in \{0,1\}, \quad \forall r \in R, \forall p, p' \in P, \forall t \in T. \tag{41}$$

$$\theta^{rr'}_{pp'\hat{p}\hat{p}'t} \in \{0,1\}, \quad \forall r, r' \in E, \forall p, p', \hat{p}, \hat{p}' \in P, \forall t \in T. \tag{42}$$

Constraints (2) formulate the meeting of the supply at the Reception subsystem. As previously stated, meeting the supply at this subsystem consists of unloading the trains, and therefore, these constraints guarantee that unmet supplies are updated. The unmet supplies IR_{p0} at period zero are an input data of the problem. Meeting the demand at the Pier subsystem, i.e., loading cargo onto the ships, is imposed by constraints (3). The unmet demands (IP_{np0}) at period zero are an input data of the problem.

Constraints (4) guarantee that stocks are kept up-to-date at each subarea. Constraints (5) define the storage capacity of each subarea. Constraints (6) ensure that no equipment will have its capacity exceeded. Constraints (6) allow two routes that share the same equipment to be used simultaneously as long as they do not exceed its capacity. Constraints (7) control the stockyard allocation. Once a subarea is allocated for a product, it cannot be used for any other product in the same period. When a subarea is empty, the product 0 is allocated to it, i.e., $f^s_{0t} = 1$. Constraints (8) control the replacement of products in a subarea. If $Sf^s_{pp't} = 1$, then product p has been replaced with p' at period t. The requirement that there can be only one product p stockpiled at a subarea s in period t if the stockyard allocation decision variable is valued 1 is enforced by constraints (9) and (10). If a route r at period t is used ($u^r_{pp't} = 1$) to carry a product p or p', constraints (11, 12 and 13) guarantee that its availability and capacity (measured in hours) are met.

Equations (14) and (15) define disjunctive constraints for each pair of conflicting routes ($r, r' \in E$) and $(r, r') \in R^x$. They also establish the order of products p and $\hat{p}$ sharing equipment. If $\theta^{rr'}_{pp'\hat{p}\hat{p}t} = 1$, (15) is redundant, and (14) ensures that product p or (p') precedes $\hat{p}$ or ($\hat{p}'$) and that the start time of $\hat{p}$ is greater than the start time of p; if $\theta^{rr'}_{pp'\hat{p}\hat{p}t} = 0$, $\hat{p}$ precedes p. The same is true for all the other pairs of conflicting routes ($r, r' \in E$) such that $r \in R^x$ and $r' \in R^y$ constraints (16,17); $r \in R^x$ and $r' \in R^z$ constraints (18,19); $r, r' \in R^y$ constraints (20,21); $r, r' \in R^z$ constraints (22,23), and $r \in R^z$ and $r' \in R^y$ constraints (24,25). Constraints (26,27), (28,29) and (30,31) ensure that a product cannot be introduced before or after the period in which it is established in the production plan. Finally, constraints (32 to 42) define the nonnegativity and integrality constraints.

5 Solution approach

To solving the PFPSP problem, we adopt a hierarchical approach where production planning and scheduling are solved separately. Figure 3 presents the solution strategy.

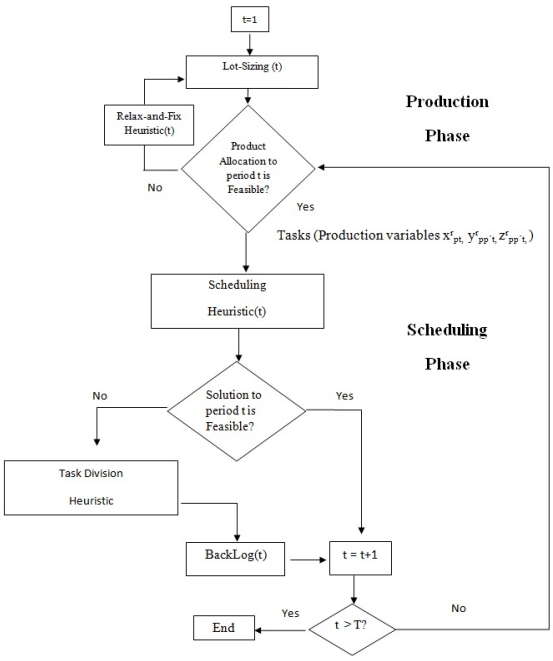


Fig. 3. Solution strategy

In this approach, production planning and scheduling are solved period by period. In the production phase, a relaxed version of the PFPSP is solved through a commercial solver. In this version, scheduling and integrality constraints are relaxed. In the remainder of this article, this relaxed problem will be called relaxed PFPSP. The production variables (x^r_{pt}, $y^r_{pp't}$ and $z^r_{pp't}$) are sent to the scheduling phase to select and schedule the routes for a given period t. The scheduling phase defines the start time and the end time for each task, considering the sharing of equipment among the selected routes.

If a feasible schedule is not found, we transfer activities to the following period (backlog). This transfer is carried out as follows: If only part of the task is unfeasible in the period t, a new task is created in t + 1, considering its length only part unprocessed in t. If the task completely violated the period t, the task is completely sent to the period t + 1. In case of finding a feasible schedule, we just move towards the next period. The Figure 4 illustrates this procedure: Task 6 has been completed after the time period. Therefore, a part of your processing time will be transferred to the following period and scheduled with other tasks.

5.1 Production Phase

At this phase, the method solves a capacitated Lot sizing. The problem allocates products into sub-areas, with the aim of selecting the best sub-area for each product. While the Lot sizing is easily solved by a commercial solver, the allocation of products considers several integer variables, making the problem harder to be solved. Therefore we use a relax-and-fix strategy to efficiently solve the allocation problem.

The relax-and-fix heuristic works by fixing decision variables in a sequence until reaching a feasible solution. In our case, we first deal with the number of products allocated in each sub-area, variable f^s_{pt}, selecting the variable with more allocated products. The relaxed PFPSP is solved again, and the process is repeated until reaching a feasible set of variables f^s_{pt}. In a similar fashion,

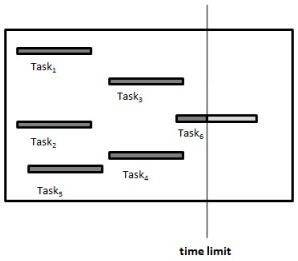


Fig. 4. Transferring tasks (Backlog)

the algorithm works fixing fractional variables. In an iterative procedure, the variable of the subarea with the largest fractional value is set to one. By the end of the production phase, an integer solution is found (considering the allocation problem and lot sizing) for each period of the time horizon. After that, the scheduling problem is solved in the Scheduling Phase.

5.2 Scheduling Phase

The PFPSP production variables define the product type, the quantity and the route used to transport the products. The scheduling problem consists of establishing the start and end times for these tasks, considering incompatibility restrictions. Preemption is not allowed and the objective is to minimize the makespan. Hereafter, this scheduling problem is called the scheduling problem with incompatibility jobs (SPIJ).

The SPIJ is not a new problem, many solutions have been proposed in the literature, such as scheduling with incompatible jobs [6], scheduling jobs using an agreement graph [2], and multicoloring and job-scheduling [5]. Previous works as [6], [16], have shown that the SPIJ and several variations are NP-Hard.

5.3 SPIJ Heuristic

To efficiently find good solutions, a greedy randomized search procedure (GRASP) was implemented. GRASP is an iterative algorithm proposed by Feo and Resende [15] that basically consists of two phases, greedy construction and local search. The greedy construction phase builds a feasible solution s, whereas the local search investigates the neighborhood of s, searching for better solutions. The main phases of the heuristic are described in the following.

```
1: procedure GRASP-SCHEDULING()                                    ▷
2:     jobs = Production variables (relaxed PFPSP);
3:     for i=1 do MaxIteration
4:         Solution = GreedyRandomizedConstruction(Seed);
5:         Solution = LocalSearch(Solution);
6:         Solution = UpdateSolution(Solution, BestSolution);
7:     end for
8:     Return BestSolution;
9: end procedure
```

At each iteration of procedure *GreedyRandomizedConstruction*, the algorithm considers the jobs extracted from planning not yet scheduled, as the list of candidate elements. A greedy solution for the SPIJ is constructed as follows: Select randomly a job i from list of candidate elements at random. Next, define the lowest start time for the job, keeping already scheduled jobs that conflict with i without overlapping. Once all jobs are scheduled, a feasible solution for the SPIJ is provided.

The local search consists in exchanging the order of jobs found in the greedy construction phase. Two neighborhoods are explored. The first consists of exchanging the first and the last job of the sequence, then the second and the penultimate, etc. The second neighborhood explores exchanges between job pairs, i.e., the first and the second are exchanged with the last and the penultimate, following the same sequence of the first neighborhood.

6 Computational Experiments

The experiments are performed based on an real product flow problem in an iron ore port terminal in Brazil, recognized as one of the largest worldwide. The basic parameters are the number of periods, the products and the routes. In general, they work with seven periods of one day or fourteen periods of twelve hours. Table 3 highlights the main parameters used to create the instances. The parameters α_{pt}, β_{npt}, $\gamma^s_{p,p',t}$, $\lambda_{pp'}$ and σ^r are part of the PFPSP formulation.

Table 3. Data used to generate the instances

Parameter	Description
Stockyard	The product storage area is divided into four stockyards.
Delivery	Two berths: two ships can be loaded simultaneously at berth 1.
Equipment	Five car dumpers, four ore reclaimers, three stackers/reclaimers (equipment that performs both tasks), and eight stackers.
α_{pt}	2 (two monetary units)
β_{npt}	10 (ten monetary units for the berth number one), 50 (fifty monetary units for the berth number two).
$\gamma^s_{p,p',t}$	10 (ten monetary units)
$\lambda_{pp'}$	Based on the following formula: 0.01 (monetary unit) $*\lvert p-p'\rvert$, where $\lvert p-p'\rvert$ represents the quality deviation between the product p and p'.
σ^r	Based on the following formula: 0.01 (monetary unit) $*$ length of route r.

In the iron ore port terminal, the demand nodes are those where the ship moors to receive the products. For the experiments, three demand nodes are considered: two moored ships in berth 1 and one ship in berth 2. Likewise, the supply nodes represent the points where the wagons unload the products. These points are related to the car dumpers. For the experiments, five car dumpers were considered. In this system, various products and quantities can arrive at different periods.

The priority in a port terminal is to meet the demand, so the penalty of not meeting it β_{npt}, is usually higher than the non-fulfillment of the supply (α_{pt} parameter). Even among the berths, the penalty is differentiated (β_{npt} value in Table 3). In our particular case, berth 2 meets larger ships so the priority is always to meet the ships of this berth. In the experiments, the cost of exchange products in the stockyard $\gamma^s_{pp't}$ is the same for any pair.

6.1 Instances

In this experiment, instances are created considering the following features: initial empty stock and initial stock with 30% of capacity, products supplied equal to those demanded and instances where the supplied and demanded products are different (when the product type is switched to meet the demand) and the cost of the exchange is calculated based on the parameter $\lambda_{pp'}$ from Table 3.

The first, second and third columns of the Tables, contain the number of the instance, the type and the name. For example, instance $8P5Prod$ corresponds to a planning horizon of eight periods and five different products being handled. Column Z_{LB} provides the lower limit for the PFPSP obtained by its linear relaxation. Columns Z_{UB} and Z_{BB} provide the upper and lower bounds (best bounds) obtained with the branch-and-cut algorithm of the CPLEX solver. Regarding the hierarchical procedure, column Z_{UB_1} provide the upper bound obtained for the hierarchical solution. Finally, t_{LB}, t_{UB} and t_{UB1} are the elapsed computational time to obtain the values of Z_{LB}, Z_{UB}, Z_{UB_1} respectively, expressed in seconds.

The results shown in Table 4 indicate that solving the PFPSP in optimization packages is not feasible. The solver was able to produce solutions only for half (nine) of them. In the rest, because insufficient memory, it was not possible to obtain even an upper bound. With the heuristic, it is possible to obtain solutions for all instances, all supplies and demands were met, and all tasks are scheduled respecting the duration of each period. In cases where the scheduling was unfeasible, the tasks were transferred to the next period (backlog) and scheduled.

In the experiments described in Table 5, it was necessary to manipulate more products and routes (as excess supply must remain at the stockyard); therefore, the optimization package was not able to find a feasible solution for any case. Similar to the results found in Table 4, with the hierarchical approach (Table 5), it is possible to obtain solutions for all instances,

Table 4. 110 routes - capacity ranging from 8,000 to 12,000 t/h - supply equal demand

			CPLEX					Heuristic Approach	
Number	Type	Instance	Z_{LB}	t_{LB}	Z_{UB}	t_{UB}	Z_{BB}	Z_{UB_1}	T_{UB_1}
1		4P5Prod	138.16	5	142.14	205	142.13	143.6	12
2		4P10Prod	140.4	34	143.25	785	143.25	143.6	13
3	Empty stock	8P5Prod	276.32	10	281.62	683	281.62	287.2	36
4	Products	8P10Prod	279.7	1065	-	-	-	287.2	42
5	equal	10P5Prod	345.4	12	351.96	1159	351.931	359	58
6		10P10Prod	348.67	2089	-	-	-	359	63
7		4P5Prod	838.16	5	842.14	168	842.14	843.6	12
8		4P10Prod	2688.16	5674	-	-	-	2693.6	17
9	Empty stock	8P5Prod	1476.32	13	1481.67	1202	1481.56	1487.2	38
10	Different	8P10Prod	6226.32	9871	-	-	-	6237.2	60
11	products	10P5Prod	1945.4	33	1952.96	1977	1952.78	1959	60
12		10P10Prod	7345.4	2381	-	-	-	7359	83
13		4P5Prod	197.35	5.00	200.74	289.00	200.74	237.99	22
14		4P10Prod	204.22	1236.00	206.64	11.003.00	206.64	237.48	25
15	Stock in 30%	8P5Prod	-	-	-	-	-	286.76	43
16	Different	8P10Prod	-	-	-	-	-	300.30	63
17	products	10P5Prod	-	-	-	-	-	376.2	61
18		10P10Prod	-	-	-	-	-	370.4	65

Table 5. 110 routes - capacity ranging from 8,000 to 12,000 t/h - supply greater than demand

			CPLEX					Heuristic Approach	
Number	Type	Instance	Z_{LB}	t_{LB}	Z_{UB}	t_{UB}	Z_{BB}	Z_{UB_1}	T_{UB_1}
19		4P5Prod	148.3	4	-	-	-	163.39	51
20		4P10Prod	150.4	34	-	-	-	154.53	36
21	Empty stock	8P5Prod	297.50	9	-	-	-	316.75	116
22	Products	8P10Prod	305.46	879	-	-	-	323.45	91
23	equal	10P5Prod	372.82	12	-	-	-	391.65	170
24		10P10Prod	380.9	1683	-	-	-	419.44	118
25		4P5Prod	651.66	5	-	-	-	675.40	78
26		4P10Prod	1876.34	854	-	-	-	2063.2	57
27	Empty stock	8P5Prod	710.14	10	-	-	-	773.48	168
28	Different	8P10Prod	2541.04	9422	-	-	-	2635.61	163
29	products	10P5Prod	860.48	12	-	-	-	882.46	200
30		10P10Prod	-	-	-	-	-	3121.53	208
31		4P5Prod	-	-	-	-	-	151.47	49
32		4P10Prod	-	-	-	-	-	152.68	39
33	Stock in 30%	8P5Prod	-	-	-	-	-	335.83	91
34	Different	8P10Prod	-	-	-	-	-	310.55	89
35	products	10P5Prod	-	-	-	-	-	398.93	134
36		10P10Prod	-	-	-	-	-	388.99	133

7 Conclusions

We consider in this work an integrated problem of planning and scheduling. The mathematical model developed, can be used to represent various problems related to the transportation of products and stock conditions. Regarding the heuristic approach, the method is more efficient in producing a feasible solution than the solver. Furthermore, it is possible to solve medium and large instances, that with optimization packages is computationally unfeasible. Future works are planned to deal in an integrated manner with the PFPSP problem and new methods to efficiently solve the SPIJ.

Acknowledgements

This research is supported by the following institutions: VALE, Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

References

1. V. H. Barros, T. S. Costa, A. C. M. Oliveira, and L. A. N. Lorena. Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Computers & Industrial Engineering*, 60:606 – 613, 2011.
2. M. Bendraouche and M. Boudhar. Scheduling jobs on identical machines with agreement graph. *Computers & Operations Research*, 39(2):382 – 390, 2012.

3. K. Bichou and R. Gray. A logistics and supply chain management approach to port performance measurement. *Maritime Policy & Management*, 31(1):47–67, January 2004.
4. B. Bilgen and I. Ozkarahan. A mixed-integer linear programming model for bulk grain blending and shipping. *International Journal of Production Economics*, 107(2):555 – 571, 2007.
5. I. Blchliger and N. Zufferey. Multi-coloring and job-scheduling with assignment and incompatibility costs. *Annals of Operations Research*, 211(1):83–101, 2013.
6. H. L. Bodlaender, K. Jansen, and G. J. Woeginger. Scheduling with incompatible jobs. *Discrete Applied Mathematics*, 55(3):219 – 232, 1994.
7. N. Boland, D. Gulezynski, and M. Savelsbergh. A stockyard planning problem. *EURO Journal on Transportation and Logistics*, 1(3):197–236, 2012.
8. N. Boland and M. Savelsbergh. Optimizing the hunter valley coal chain. In H. Gurnani, A. Mehrotra, and S. Ray, editors, *Supply Chain Disruptions*, pages 275–302. Springer London, 2012.
9. M. Byrne and M. Bakir. Production planning using a hybrid simulation. pages 305 – 311, 1999.
10. M. Byrne and M. Hossain. Production planning: An improved hybrid approach. *International Journal of Production Economics*, 93?94:225 – 229, 2005. Proceedings of the Twelfth International Symposium on Inventories.
11. B. A. Calfa, A. Agarwal, I. E. Grossmann, and J. M. Wassick. Hybrid bilevel-lagrangean decomposition scheme for the integration of planning and scheduling of a network of batch plants. *Industrial & Engineering Chemistry Research*, 52(5):2152–2167, 2013.
12. D. Conradie, L. E. Morison, and J. W. Joubert. Scheduling at coal handling facilities using simulated annealing. *Mathematical Methods of Operations Research*, 68(2):277–293, 2008.
13. S. Dauzere-Peres and J.-B. Lasserre. Integration of lotsizing and scheduling decisions in a job-shop. *European Journal of Operational Research*, 75(2):413–426, 1994.
14. S. Dauzere-Peres and J.-B. Lasserre. On the importance of sequencing decisions in production planning and scheduling. *International Transactions in Operational Research*, 9(6):779–793, 2002.
15. T. A. Feo and M. G. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2):67 – 71, 1989.
16. R. Gandhi, M. Halldrsson, G. Kortsarz, and H. Shachnai. Improved bounds for sum multicoloring and scheduling dependent jobs with minsum criteria. In G. Persiano and R. Solis-Oba, editors, *Approximation and Online Algorithms*, volume 3351 of *Lecture Notes in Computer Science*, pages 68–82. Springer Berlin Heidelberg, 2005.
17. B. Kim, B. Koo, and B. S. Park. A raw material storage yard allocation problem for a large-scale steelworks. *The International Journal of Advanced Manufacturing Technology*, 41(9-10):880–884, 2009.
18. G. C. Menezes and G. R. Mateus. Integrated planning and scheduling in bulk cargo terminal. Technical report, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brasil, 2013.
19. H. Meyr and M. Mann. A decomposition approach for the general lotsizing and scheduling problem for parallel production lines. *European Journal of Operational Research*, 229(3):718 – 731, 2013.
20. A. M. Newman, R. Rubio, R. Caro, A. Weintraub, and K. Eurek. A review of operations research in mine planning. *Interfaces*, 40(3):222–245, 2010.
21. B. S. Pimentel, G. R. Mateus, and F. A. Almeida. Stochastic capacity planning and dynamic network design. *International Journal of Production Economics*, 145(1):139 – 149, 2013.
22. T. Robenek, N. Umang, and M. Bierlaire. A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. *European Journal of Operational Research*, 235(2):399 – 411, 2014.
23. R. Sikora, D. Chhajed, and M. J. Shaw. Integrating the lot-sizing and sequencing decisions for scheduling a capacitated flow line. *Computers & Industrial Engineering*, 30(4):659 – 679, 1996.
24. G. Singh, D. Sier, A. T. Ernst, O. Gavriliouk, R. Oyston, T. Giles, and P. Welgama. A mixed integer programming model for long term capacity expansion planning: A case study from the hunter valley coal chain. *European Journal of Operational Research*, 220(1):210 – 224, 2012.
25. C. Wolosewicz, S. Dauzere-Peres, and R. Aggoune. A Lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem. *European Journal of Operational Research*, 244(1):3–12, 2015.

A new taxonomy for the Green Vehicle Routing Problem

E. Marrekchi[1], W. Besbes[2] and D. Dhouib[3]

Keywords: Green Vehicle Routing Problem, Rich Green Vehicle Routing Problem, Taxonomy

1 Introduction

With the prominence of the globalization, distances between the nodes in the distribution network have grown considerably in the last decades. The Department of Transport from the UK Government has mentioned that road transport accounts for around 22% of UK CO2 emissions and that it emits 44% of the UK's nitrogen oxides and up to 75% or more of nitrogen oxides in conurbations *Sbihi and Eglese (2007a)*. Green logistics has emerged as the new agenda item in the supply chain management to minimize the costs related to the environmental issues. The Green Vehicle Routing Problem (GVRP) is the new weapon against the environmental threats of the roads freight transport. In the last decade, an increasing academic attention was addressed to the green issues and new variants of the GVRP to include more complex constraints and objectives, stimulated by the real-life complex GVRPs.

This paper represents some aspects of a real-world application optimization criteria and constraints leading to Rich Green Vehicle Routing Problem. There is no precise definition to the Rich GVRP but a relevant taxonomy can help to distinguish the GVRPs from the Rich GVRPs.

This paper is organized as follows. Section 2 presents briefly the definition of the GVRP based on the definition of the VRP, its variants and a related classification. Section 3 describes the taxonomy and introduces the key characteristics considered. In Section 4, a detailed analysis of the taxonomy results are discussed and some future researches are presented. Section 5 concludes this paper.

2 The Green Vehicle Routing Problem

The GVRP is the extension of the classical Vehicle Routing Problem consisting of routing a set of vehicles (homogeneous or heterogeneous) visiting a set of customers such that the total distance traveled by the vehicles is minimized. *Lin et al. (2013)* detailed the different definitions of the VRP and its variants (approximatively 19 variants). In 1959, *Dantzig and Ramser (1959)* introduced to the first time the VRP and a novel variant which is the Capacitated VRP (CVRP). The CVRP consists of determining the routes of a fleet of vehicles constrained by a capacity to serve a set of customers. Seven years later, *Cooke and Halsey (1966)* introduced the Time Dependent VRP (TDVRP) with the distinctive characteristic of the dependency of the distance between points or on the time of the day. Since that time till nowadays, several researches are being conducted to study the VRP and its variants like the Pickup and Delivery VRP (VRPPD) (*Wilson and Weissberg(1967)*), Multi-Depot VRP(*Tillman(1969)*), Stochastic VRP (*Tillman(1969)*), VRP with Time Windows (*Russell(1977)*), Fleet Size and Mix VRP(*Golden et al.(1984)*), Multi Echelon VRP, Dynamic VRP…..

In 2007, *Sbihi and* Eglese (2007a, 2007b) applied the Time-dependent VRP as an approach to deal with the minimization of emissions during traveling and it was the first introduction of the concept of GVRP. The Green VRP extends the definition of the VRP by the focus on energy consumption. The same variants of the VRP have being tackled the last decade with the addition of the green component. According to *Lin et al. (2013)*, the GVRP is subdivided into three main categories: (i) The VRP in Reverse Logistics; (ii) The Green-VRP; and (iii) The Pollution Routing Problem (PRP).

The Environmental Protection Agency (EPA) stated that in the United States in 2009, only the quarter of 2.17 million tons of electronic products requiring end-of-life management were collected for recycling and disposal. Thus, "a process which encompasses the logistics activities all the way from used products no

[1] LOGIQ, Institut Superieur de Gestion Industrielle, Sfax, Tunisia. emna.marrekchi.b@gmail.com
[2] Taif University, Taif, KSA. *walid.besbes@gmail.com*
[3] Institut Superieur de Gestion Industrielle, Sfax, Tunisia. dialadhouib@yahoo.fr

longer required by the user to products again usable in a market" *Fleischmann et al. (1997)* should be implemented. Few studies on reverse logistics network design have considered environmental factors. *Wang et al. (2011)* modeled the environmental issues as minimization of carbon emissions. *Duque et al. (2007)* considered the environmental issues as constraints on the amount of carbon emissions.
The Green-VRP is the Vehicle Routing Problems concerning energy consumption. *Xiao et al. (2012)* stated that fuel cost accounts for a large part of the total cost of the transportation costs and can reduce the GHG emissions significantly, *Erdogan and Miller (2012)*. Therefore, FC is an important index in the Green-VRP. Factors influencing fuel consumption have been studied by *Ardekani et al. (1996), Bigazzi and Bertini (2009) and Demir et al. (2011)*. These factors belong to five main categories: vehicle, environment, traffic, driver and operations. *Eglese and Black (2010)* argue that speed is an important factor when estimating emissions. *Demir et al. (2011)* added other factors such as load weight and distribution, engine type and size, vehicle design and road gradient. *Demir et al. (2014a)* presented a detailed description of the FC's models used in the literature. They subdivided the models into two main categories: Microscopic and Macroscopic models. Macroscopic models use average aggregate network parameters to estimate network-wide emission rates, while the microscopic models estimate the instantaneous vehicle FC and emission rates at a more detailed level. Methodology for calculating transportation Emissions and Energy consumption (MEET) by *Hickman et al.(1999)* for heavy goods vehicles and Computer Program to calculate Emissions from Road Transport (COPERT) by *Kouridis et al. (2010)*. Comprehensive Modal Emissions Model (CMEM) presented by *Scora and Barth (2006)*, *Barth et al. (2005)* and *Barth and Boriboonsomsin (2008)* for heavy-goods vehicles is the main used model in the GVRP field on the grounds that its ease application and that it encompasses detailed vehicle specific parameters for the estimations. An Instantaneous Fuel Consumption Model (IFCM) (1985), A Four-Mode Elemental Fuel Consumption Model (FMEFCM) (1985), An Average Speed Fuel Consumption Model (ASFCM) are three of the list of the microscopic models described by *Demir et al. (2014a)*. In this paper, the authors detailed the applications of the mentioned models at the operational level of planning through the literature.
A relevant study on the PRP, aiming at choosing a vehicle dispatching scheme with less pollution through CO2 emission, was conducted by *Bektaș T and Laporte G (2011)* in which they developed a comprehensive objective function that accounts not only for traveled distance, but also for GHG emissions, fuel, travel times and their costs. *Demir et al. (2012)* added to their proposed PRP the constraint of time windows.

3 GVRP Taxonomy

Creating taxonomy is an efficient and effective way of consolidating knowledge, *Reisman (1992).* From the taxonomy, not only effective and efficient storage of the existing body of literature is drawn, but also knowledge is built and expanded. Neither surveys nor classifications of the GVRP have already considered the richness component. It is, to the best of our knowledge, the first survey that tackles the Rich GVRP. This taxonomy aims to highlight the different facets of richness encountered in the literature to distinguish Rich GVRPs from GVRPs. For this we have selected papers dealing with the GVRP published since 2007 and we based our taxonomy on the approach of *Lahyani et al. (2015)*. As a result, 41 papers published in different journals and conferences are studied trying to be the most exhaustive as possible. However, we apologize for any not deliberate exclude of relevant paper.

3.1) Taxonomy

In this section we describe the taxonomy and its main characteristics. The taxonomy was iteratively built as we go deep in the literature. Characteristics were designed to present central concepts of Green Routing that are linked to real-life industrial applications. It is obvious that not all the details related to the topic would be considered, but relevant features would be kept.
To construct the taxonomy, the problem characteristics are subdivided into two main levels: Strategic & Tactical decisions and Operational decisions. Under each of these two classes, the most important adopted attributes are listed, based on which the problem is classified as rich. Strategic and Tactical decisions do not affect the daily transport activities but they affect the routing plan significantly. They include decisions

related to the number of depots, the planning horizon, the frequencies of visits to the customers and the operations' type.
The operational decisions describe the distribution planning with the emissions' calculation model, the characteristics related to the used vehicle and specific constraints for the vehicle, driver and time.

3.1.1) Problem characteristics

In this section we describe the related sublevels of the Strategic and Tactical decisions and Operational decisions presented in Table 1. We briefly define each sublevel with some relevant paper.

Table 1 : A Taxonomy for the Rich GVRP

1) Strategic and tactical decisions	**2) Operational decisions**
1.1) Number of depots	2.1) Emissions models
1.1.1) Single	2.1.1) Microscopic models
1.1.2)Multiple	2.1.1.1) CMEM
1.2) Planning period	2.1.1.2)Others
1.2.1)Single period	2.1.2) Macroscopic models
1.2.2) Multi-period	2.1.2.1)COPERT
1.3) Multiple use of vehicles	2.1.2.2) MEET
1.3.1) Single trip	2.1.2.3) Others
1.3.2)Multi-trip	2.1.3) Others
1.4) Operation type	2.2) Vehicles
1.4.1) Pickup or delivery	2.2.1) Type of the fleet
1.4.2) Pickup and delivery	2.2.1.1) Homogeneous
	2.2.1.2) Heterogeneous
	2.2.2) Size of the fleet
	2.2.2.1) Fixed
	2.2.2.2) Unlimited
	2.2.3) Capacity constraints
	2.2.4) Technology
	2.2.4.1) Traditional
	2.2.4.2) Alternative Fuel Vehicle
	2.3) Time Constraints
	2.3.1) Time Windows' Restrictions
	2.3.2) Idle time allowance
	2.3.3) Tardiness penalty' Application
	2.3.4) Time dependency
	2.4) Speed optimization
	2.5) Drivers' wage consideration
	2.6) Objective function
	2.5.1) Single objective
	2.5.2) Multi-objective

3.1.1.1) Depots: in the first attempts to solve the GVRP, the single depot was the common used characteristics. However, this case seems to be very restrictive for real-life applications where customers have to be assigned to the appropriate depot regarding its location and capacity which affect the overall costs. Some papers deal with the multi-depots GVRP, for instance, *Taha et al. (2014)*, *Li et al. (2015)*, *Yang et al. (2015)*, *Zare-Reisabadi and Mirmohammadi (2015)*, *Tajik et al. (2014)*, *Suzuki (2011)*, *Fagerholt et al. (2010)*, *Bauer et al. (2010)* and *Schneider et al. (2012)*.
3.1.1.2) Planning period: the horizon planning can be either of single-period or multi-period. The single-period case is practically not considered in the GVRSP and also in the classical VRP because of the fact that

it does not reflect real-life cases. In fact, at each period of the planning period (the most used is the day), decisions regarding which customers are served in this period and which orders are postponed to the next period are done. All papers treated in this survey are multi-periodic GVRP.

3.1.1.3) Multiple use of vehicles: in this component, the same vehicle may perform several trips during the planning horizon. From an optimization view, it is obvious that a vehicle have to be used for at least one time and why not several times. For this, all papers dealing with the GVRPs do not restrict the single use.

3.1.1.4) Operation type: two main classes of problems can be depicted at this level: problems where goods are either delivered or picked up and problems where goods are picked up and delivered to customers. The first class is the simple case of any routing problem where products are loaded in the depot and then delivered to the customers, or picked up from customers and unloaded at the depot. This case concerns the majority of the GVRPs formulations.

The second case is the simultaneous load of goods from the depot to the customers (delivery) and unloads of goods the customers locations to the depot (Pickup). This particular case was recently treated by *Huang et al. (2012)*, *Tajik et al. (2014)* and *Zhang et al. (2015)*.

3.1.1.5) Emissions' Models: as the treated problem is the GVRP, so the emissions are the central point. As mentioned in the previous section, the Emissions are closely linked to the FC. In this sublevel, the main used models of the emissions' calculations are considered for the papers' classifications. Microscopic models estimate the instantaneous vehicle FC and emission rates at a detailed level by taking into account the small details related to the vehicle, engine, road… *Kara et al. (2007)* developed a model that uses different vehicle characteristics to minimize the energy consumption like load, friction force, coefficient of friction… CMEM is the most used model for the calculations of emissions. From Table 2, 22 papers among 33 have used the CMEM , *Bektaș and Laporte (2011)*, *Xiao and Konak (2015)*, *Demir et al. (2012)*, *Soysal et al. (2015)*… while the others are divided between different macroscopic models and some other microscopic. The macroscopic models have been well used regarding their simple application. *Jabali et al. (2012)* used the MEET model. Some authors relate the FC and emissions to some factors like speed *Figliozzi (2010)*, to distance and load *Huang et al. (2012)*... *Jabir et al. (2015)* used a simple expression to compute CO2 emissions depending on the volume of FC, diesel fuel price, average price per unit weight of CO2, weight of CO2 emission per liter consumption of diesel and some other factors. *Kwon et al. (2013)* used a fuel based method to determine the effective carbon price in a short-term-cap-and-trade framework.

Some other authors did not use the existing models and develop special function adapted to their problems like *Zhang et al. (2015)* who developed an expression that measures the FC based on the distance and the speed. *Suzuki (2011) and Fagherholt et al. (2010)* used a simple function of FC related to the speed. Some others did not use functions but implicitly consider that FC is minimized through some factors such that distance for *Qian and Eglese (2016)* and *Li et al. (2015)*. *Taha et al. (2014)* and *Erdogan S, Miller-Hooks (2012)* considered that the use of alternative fuel vehicles and sources contribute to the carbon dioxide emissions. *Ericsson et al. (2006)* estimated the potential for reducing FC and thus the emissions of CO2 through a navigation system where optimization of route choice is based on the lowest total FC instead of the traditional shortest distance or time.

3.1.1.6) Vehicles: the evident component in any routing problem is the mean of transport: vehicles. For this, many characteristics are related to vehicles. Two main classes of the fleet of vehicles are considered in the literature: Homogeneous and heterogeneous fleet. Homogeneous fleet composed of identical vehicles is the main used case in the GVRP. Heterogeneous fleet is in major cases related to the technology characteristic which is the Alternative Fuel Vehicles (AFV): vehicles running on a fuel other than traditional petroleum fuels (petrol or Diesel fuel) (e.g. electric car, hybrid electric vehicles, solar powered). This type of vehicles is a new trend and recently tackled in the GVRP context by *Erdogan and Miller-Hooks (2012)*, *Taha et al. (2014)*, *Koç and Karaoglan (2016)*… The fleet size may be fixed or unlimited. In the majority of the papers, the size is fixed. Capacity constraint, which is the major pillar of the Capacitated GVRP and VRP, is in 99% considered.

3.1.1.7) Time constraints: the main time constraint studied is the time window (TW) which imposes that the service must start and end within a given time window at each customer. Two types of time windows are considered: Hard time window and soft time Window. In the hard TW, the vehicle is allowed to arrive

before the expected TW and hence waits until the customer becomes available. Late arrive is not allowed. This idle time was tackled in some papers in the context of GVRP, *Xiao and Konak (2015a)*, *Xiao and Konak (2015b)*, *Franceschetti et al. (2013)*, *Demir et al. (2012)*, *Jabir et al. (2015)*… Soft TW allows the services starting after the allowed TW but with penalty application for late. The penalty application was firstly applied by *Xiao and Konak (2015a)* in the context of GVRP and few papers deal with it, *Zare-Reisabadi (2015)* and *Tajik et al. (2014)*.
In the time constraints, traffic congestion has been a new studied characteristic in the GVRP called Time Dependent GVRP: the consideration of periods of congestion (peak hours) in the scheduling. The main idea of *Franceschetti et al. (2013)* was that congestion affects the speed and hence the CO2 emissions.
3.1.1.8) Speed optimization: recent researches tackle the speed and consider it as a major factor to reduce the CO2 emissions and the FC. In 2007, *Palmer (2007)* investigated the role of speed in reducing CO2 emissions under various congestion scenarios and TW settings. *Jabali et al. (2012)* presented a GVRP in which they estimate the amount of CO2 based on a nonlinear function of speed and present analysis to find the optimal speed with respect to emissions. *Bektaș and Laporte G (2011)* considered the speed among the factors of the amount CO2 emitted. They studied the role of speed on the energy-efficient solution and demonstrate the effect of speed under TW' restrictions as it minimizes the CO2 emissions.
3.1.1.9) Drivers' wage consideration: many authors conclude that the drivers' wage has an important effect on the overall cost. Recent formulations take this component into account to minimize the overall cost.
3.*1.1.10) Objective function:* few papers consider mono-objective GVRPs and the majority treats multiple objectives. The most common objectives include minimizing some or all of these criteria: the total traveled distance, the total spent time, the total tour cost, the CO2 emitted quantities, the arrival time and the drivers' wage. For such problems, adequate formulations and algorithms must ensure the solving and the approximation given that the problem of the GVRP is NP-hard, *Archetti et al. (2015)*.

4 Taxonomy Analysis

In this taxonomy, 41 papers are analyzed in deep in the tables 2 and 3 aside the mentioned papers in previous sections. The difference between the last two tables is that Table 2 considers papers containing mathematical formulations contrarily to Table 3. Each paper is classified according to the attributes defined in Table 1. When the attribute is found in the corresponding paper, an 'X' is ticked in the associated cell. The first column headed *Number* provides the corresponding number assigned to the paper in the references' list. The column *Real Case Study* indicates, if ticked, that the corresponding paper includes a set of instances based on real-life application. The column *Rich* is ticked if the treated problem contains at least two main characteristics of the GVRP and contains different operational pertinent constraints. The last column *Methods* indicates the type of designed methods.
In Table 2 and Table 3 the majority of attributes are present in one paper at least. Some columns are left empty without any tick. It is intentionally so to show the untreated attributes through the literature. For the strategic & tactical decisions, the columns not ticked (1.2.1, 1.3.1, …) represents the attributes not studied regarding the fact that they do not reflect real situations. Heterogeneous fleet, unlimited size of the fleet of vehicles, the application of tardiness penalty and time dependency seem to be not having too much interest.
The papers belong to a time period from 2007 to 2016. First deductions are based on the division of this horizon into three periods of 3 years length: [2007-2010[, [2010-2013[and [2013-2016]. In this 41 papers, 5% were published from 2007 to 2010, 39% from 2010 to 2013 and 56% the last 3 years. 23 papers were published in the last year (2015-2016) among them 14 papers in 2015 and 2016. This division demonstrates the growing attention addressed to the GVRPs. In fact, from 2010, intensive researchers were conducted to tackle this new VRP variant regarding the new environmental issues and regulations. As the VRP is known to be NP-hard, GVRP which is a more complicated case with more constraints and objectives is also NP-hard as demonstrated through the literature. For this, the first initiatives in the GVRP were conducted on small size instances with the application of exact methods using commercial solvers. This method accounts for 56% of the papers and was depicted in the period of [2007-2010]. Since 2011, great attention has been addressed to heuristics and metaheuristics to solve bigger instances' size of the GVRPs. 48% of the papers

treated metaheuristics and this in the time period [2014, 2016]. As seen and mentioned above, a great focus has been addressed to these approximate methods. Genetic algorithms, simulated annealing and tabu search are the most used metaheuristics. Local search and Clark and Wright Savings are the most treated heuristics, and the remaining is dedicated to special heuristics to specific problems.
For the strategic and tactical decisions' attributes, the most adopted are the single depot, multi-period, multi trip and an operation of pickup or delivery. The Green VRPPD is studied seldom due to the complicated network and data even if it is often encountered in real-life industrial situations. CMEM is the most emissions' microscopic model used with a percentage of 22 due to the exhaustive consideration of several characteristics of the vehicle, the road… The vehicles' attributes are obviously intensively studied through the literature with around 98% of the papers considering the capacity constraint which is the main pillar of the CVRP and Capacitated GVRP. This percentage is justified by the huge attention given to this variants and the attempts of authors to be the nearest from the real-life cases. Heterogeneous fleets are receiving growing attention going with the new energy sources and technologies. Only 10% of papers consider heterogeneous fleet dating essentially for [2014-2016].
Like the customers' satisfaction is the main goal of modern organizations, time windows expected must be satisfied. 50% of the considered papers deal with this constraint. The quarter of these papers treats the idle time allowance' constraint simultaneously with the TW: if TW is expected, waiting is preferable to minimize the service tardiness and so to respect the TW.
Tardiness penalty is a recent concept in the GVRPs. It was introduced by *Xiao and Konak (2015a)* and applied in only four papers even if it is an efficient mean to respect the deadlines.
Speed is having intensive attention in the GVRPs as it contributes well in the CO_2 emissions. Forty seven percent of papers considers it and tries to reach optimal values. The same remark is conducted for the drivers' wage seen as a fundamental component of the overall cost.

Table 2: Selected papers devoted to mathematical formulations

Number	Paper	1	1.1.1	1.1.2	1.2.1	1.2.2	1.3.1	1.3.2	1.4.1	1.4.2	2	2.1.1.1	2.1.1.2	2.1.2.1	2.1.2.2	2.1.2.3	2.1.3	2.2.1.1	2.2.1.2	2.2.2.1	2.2.2.2	2.2.3	2.2.4.1	2.2.4.2	2.3.1	2.3.2	2.3.3	2.3.4	2.4	2.5.1	2.5.2	2.6	Real case study	Rich	Methods
5	Bauer et al. (2010)			X		X		X	X				X					X		X		X	X		X				X			X	X	Exact solution using commercial solver
6	Bektaş & Laporte (2011)		X			X		X	X			X						X		X		X	X		X				X		X	X	X	X	Exact solution using commercial solver
11	Demir et al. (2012)		X			X		X	X			X						X		X		X	X		X	X			X		X	X		X	Adaptive Large Neighborhood Search Algorithm
13	Demir et al (2014b)		X			X		X	X			X						X		X		X	X		X	X			X		X	X		X	Adaptive Large Neighborhood Search
16	Erdogan & Miller-Hooks (2012)		X			X		X	X								X	X			X	X		X	X				X	X				X	Modified Clark and Wright savings heuristic
19	Fagerholt et al (2010)			X		X		X	X								X	X		X		X	X		X	X				X	X			X	Exact solution using commercial solver
20	Figliozzi M (2010)		X			X		X	X							X		X		X		X	X		X	X		X	X		X			X	Heuristic for the emissions' minimization
22	Franceschetti et al. (2013)		X			X		X	X			X						X		X		X	X		X	X		X	X		X	X		X	Heuristic for the departure time and speed
26	Huang et al. (2012)		X			X		X		X						X		X		X		X	X								X			X	Exact solution using commercial solver
29	Jabir et al. (2015)		X			X		X	X							X		X		X		X	X			X					X			X	Ant Colony+ Variable Neighborhood Search Algorithm
30	Jaramillo J.R (2010)		X			X		X	X									X		X		X	X							X					Local search procedure
32	Kara et al. (2007)		X			X		X	X				X					X		X		X	X							X				X	Exact solution using commercial solver
33	Koç et al. (2014)		X			X		X	X			X							X	X		X	X		X	X			X		X	X	X	X	Metaheuristic: Hybrid Evolutionary algorithm
34	Koç and Karaoglan (2016)		X			X		X	X								X	X			X	X		X	X				X	X			X	X	Simulated annealing heuristic
36	Küçükoğlu et al. (2013)		X			X		X	X				X					X		X		X	X								X			X	Exact solution using commercial solver
38	Kramer et al. (2015a)		X			X		X	X			X						X		X		X	X		X	X			X		X			X	Speed and departure optimization algorithm
39	Kramer et al. (2015b)		X			X		X	X			X						X		X		X	X		X	X			X		X	X		X	Local search procedure
40	Kwon et al. (2016)		X			X		X	X							X			X	X		X	X											X	Tabu search algorithm
42	Li et al. (2015)			X		X		X	X								X	X		X		X	X						X		X			X	Clark and Wright savings heuristic+Local search
44	Mancini S (2015)		X			X		X	X									X		X		X		X						X				X	Large Neighborhood Search
47	Qian & Eglese (2016)		X			X		X	X								X	X			X	X	X		X	X			X			X		X	A column generation based tabu search algorithm
52	Schneider et al. (2012)			X		X		X	X									X		X		X	X			X				X				X	Hybrid Variable neighborhood Search+Tabu search
55	Soysal et al. (2015)		X			X		X	X			X						X		X		X	X				X	X		X	X	X		X	Exact solution using commercial solver
56	Suzuki Y (2011)			X		X		X	X							X		X		X		X	X			X	X							X	Simulations
57	Taha et al. (2014)			X		X		X	X								X	X		X		X	X							X				X	Exact solution using commercial solver
58	Tajik et al. (2014)			X		X		X		X			X					X		X		X	X		X	X	X		X		X	X		X	Robust counterpart mathematical model
61	Tiwari and Chang (2015)		X			X		X	X				X						X	X		X	X							X				X	Block recombination algorithm
64	Xiao and Konak (2015a)		X			X		X	X			X						X		X			X		X	X	X	X	X		X			X	Simulated annealing algorithm
65	Xiao and Konak (2015b)		X			X		X	X				X						X	X		X	X		X		X	X	X		X			X	Dynamic Programming + Genetic Algorithm
66	Xiao et al. (2012)		X			X		X	X				X					X		X		X	X								X			X	Simulated Annealing Algorithm
67	Yang et al. (2015)			X		X		X	X										X	X		X	X			X			X		X			X	Genetic Algorithm
68	Zare-Reisabadi et al. (2015)			X		X		X	X										X	X		X	X		X	X	X				X			X	Ant Colony+ Tabu Search Algorithm
	CVRP								X													X													
	VRPTW								X													X			X	X									
	VRPPD									X												X													

Table 3: Selected papers devoted to Approached Methods

Number	Paper	1	1.1.1	1.1.2	1.2.1	1.2.2	1.3.1	1.3.2	1.4.1	1.4.2	2	2.1.1.1	2.1.1.2	2.1.2.1	2.1.2.2	2.1.2.3	2.1.3	2.2.1.1	2.2.1.2	2.2.2.1	2.2.2.2	2.2.3	2.2.4.1	2.2.4.2	2.3.1	2.3.2	2.3.3	2.3.4	2.4	2.6	Real case study	Rich	Methods
23	Gaur et al. (2013)		X		X		X	X									X	X		X		X	X								X	Heuristic of Partitionning
27	Jabali et al. (2012)		X		X		X	X							X			X		X		X	X					X	X		X	Tabu search procedure
28	Jabbarpour et al. (2015)		X		X		X	X					X					X		X		X	X		X			X	X		X	X	Ant based algorithm
31	Jemai et al. (2012)		X		X		X	X								X		X		X		X	X					X	X			X	NSGA-II Evolutionary Algorithm
37	Kuo Y (2010)		X		X		X	X									X	X		X		X	X					X	X			X	Simulated annealing algorithm
45	Montoya et al. (2015)		X		X		X	X									X	X		X		X		X					X			X	Modified Multi Space Sampling Heuristic
54	Scott et al. (2010)		X		X		X	X						X					X	X		X	X						X		X		Exact solution using commercial solver
59	Tavares et al. (2008)				X		X	X						X								X							X		X		Shortest distance heuristic
69	Zhang et al. (2015a)		X		X		X	X								X		X		X		X											Evolutionary Local Search

5 Conclusion

In this paper, a new taxonomy for the GVRP has been proposed based on some common attributes mentioned through the literature. An attempt to distinguish Rich GVRP from GVRP has been done according to relevant attributes relative to some variants of the GVRP. Some percentages were provided trying to give a detailed analysis of the taxonomy and to introduce new research opportunities. The use of approximate methods, especially metaheuristics in the field of GVRP is in a rapidly growing and is still presenting opportunities of research. A clustering algorithm is under implementation to prove that our taxonomy is valid when a classification of the variants in distinct clusters will be obtained. From the proposed taxonomy, any researcher in the GVRP context may position its research compared to the existing literature and to the treated attributes.

References

[1] ArchettiC, Feillet D and Sperenza M.G, *"Complexity of routing problems with release dates",* European Journal of Operational Research 247 (2015), pp. 797-803.

[2] Ardekani S, Hauer E and Jemai B, *"Traffic impact models",* In Traffic flow theory (1996), Washington, DC, USA, pp.1-7.

[3] Barth M and Boriboonsomsin K, *"Real-world CO2 impacts of traffic congestion",* Transportation Research Board, 2058 (1) (2008), pp.163-171.

[4] Barth M, Younglove T and Scora G, *"Development of a heavy-duty diesel modal emissions and fuel consumption model",* Technical report, UC Berkeley: California Partners for Advanced Transit and Highways (PATH), California, USA.

[5] Bauer J, Bektaș T and Crainic T.G, *"Minimizing Greenhouse Gas Emissions in intermodal freight transport: An application to Rail service design",* Journal of Operational Research Society 61 (3) (2010), pp.530-542.

[6] Bektaș T and Laporte G, "*The pollution routing problem",* Transportation Research Part B 223 (2011), pp.1232-1250.

[7] Bigazzi A.Y and Bertini R.L*, "Adding green performance metrics to a transportation data archive",* Transportation Research Record: Journal of the Transportation Reseacrh Board, 2121 (1) (2009), pp.30-40.

[8] Cooke K.L and Halsey E, *"The shortest route through a network with time-dependent intermodal transit times",* Journal of mathematical Analysis and Applications, 14 (3) (1966), pp.493-498.

[9] Dantzig G.B and Ramser J.H, "*The truck dispatching problem",* Management Science 6 (1959), pp.80-91.

[10] Demir E, Bektaș T and Laporte G, *"A comparative analysis of several vehicle emission models for road freight transportation", Transportation* Research Part D: Transport and Environment 6 (5) (2011), pp.347-357.

[11] Demir E, Bektaș T and Laporte G, *"An adaptive large neighborhood search heuristic for the pollution routing problem",* European Journal of Operational Research 223 (2012), pp.346-359.

[12] Demir E, Bektaș T and Laporte G, *"A review of recent research on green road freight transportation", European Journal of Operational* Research 237 (2014a), pp.775-793.

[13] Demir E, Bektaș T and Laporte G, *"The bi-objective pollution-Routing Problem", European Journal of Operational* Research 232 (2014b), pp.464-478.

[14] Duque J, Barbosa-Povoa A.P.F.D and Novais A.Q, *"Synthesis and optimization of the recovery route for residual products under uncertain product demand",* Computers and Operations Research, 34 (2007), pp.1463-1490.

[15] Eglese R.W and Black D, *"Optimizing the routing of vehicles",* In A. McKinnon, S. Cullinane, M. Browne and W. Whiteing (Eds), Green Logistics: Improving the environmental sustainability of logistics (2010) (pp.215-228). Great Britain/ Kogan Page.

[16] Erdogan S, Miller-Hooks E, *"A green Vehicle Routing Problem",* Transportation Research Part E 48 (2012), pp.100-114.

[17] Ericsson E, Larsson H and Brundell-Freij K, *"Optimizing route choice for lowest fuel consumption_ Potential effects of a new driver support tool",* Transportation Research Part C 14 (2006), pp.369-383.
[18] Eskandarpour M, Zegordi S.H and Nikbakhsh E, *"A parallel variable neighborhood search for the multi-objective sustainable post-sales network design problem",* International Journal of Production Economics 148 (1) (2013), pp.117-131.
[19] Fagerholt K, Laporte G and Norstad I, *"Reducing fuel emissions by optimizing speed on shipping routes*", Journal of Operational Research Society 61 (2010), pp.523-529.
[20] Figliozzi M, *"Vehicle Routing Problem for emissions minimization",* Transportation Research Record: Journal of the Transportation Research Board, No. 2197, Transportation Research Board of the National Academies, Washington, D.C., (2010), pp. 1–7.
[21] Fleischmann M, Bloemhof-Ruwaard JM, Dekker R, Van Der Laan EA, Van Nunen JAEE and Van Wassenhove LN, *"Quantitative models for reverse logistics: a review", European* Journal of Operational Research Society 103 (1997), pp.1-17.
[22] Franceschetti A, Honhon D, Woensel T.V, Bektaș T and Laporte G, *"The time-dependent pollution routing problem",* Transportation Research Part B, 56 (2013), pp.265-293.
[23] Gaur D.R, Mudgal A and Singh R.R, *"Routing vehicles to minimize fuel consumption",* Operations Research Letters 41 (2013), pp.576-580.
[24] Golden B.L, Assad A and Dahl R, *"Analysis of a large scale vehicle routing problem with an inventory component"*, Large Scale Systems, 7 (1984), pp.181-190.
[25] Hickman J, Hassel D, Joumard R, Samaras Z and Sorenson S, *"MEET-Methodology for calculating transport Emissions and Energy consumption ",* European Commission, DG VII, Technical report (1999).<http://www.transport-research.info/Upload/Documents/200310/meet.pdf> (04-05-16)
[26] Huang Y, Shi C.H and Zhao L, *"A study on carbon reduction in the vehicle routing problem with simultaneous pickups and deliveries",* Service Operations and Logistics, and Informatics (SOLI) (2012), IEEE International Conference on Suzhou, pp.302-307.
[27] Jabali O, Van Woensel T and De Kok A.G, *"Analysis of travel times and CO2 emissions in time-dependent vehicle routing",* Production and Operations Management 21(6) (2012), pp.1060-1074.
[28] Jabbarpour M.R, Noor R.Md and Khokhar R.H, *"Green Vehicle Traffic Routing system using ant-based algorithm",* Journal of Network and Computer Applications 58 (2015), pp.294-308.
[29] Jabir E, V.Panicker V and Sridharan R, *"Modeling ants analysis of a green vehicle routing problem"*, the Twelfth AIMS International Conference on Management, Indian Institute of Management Kozhikode, India, January (2015).
[30] Jaramillo J.R, *"The green single vehicle routing problem",* SE INFORMS Annual Meeting, Myrtle Beach, SC (2010).
[31] Jemai J, Zekri M and Mellouli Kh, *"An NSGA-II algorithm for the green vehicle routing problem",* Springer- Verlag Berlin Heidelberg (2012).
[32] Kara I, Kara B.Y and Yetis M.K, *"Energy minimizing vehicle routing problem",* Combinatorial Optimization and applications, Springer (2007), pp. 62-71.
[33] Koç C, Bektas T, Jabalib O and Laporte G, *"The fleet size and mix pollution routing problem",* Transportation Research Part B 70 (2014), pp. 239-254.
[34] Koç C and Karaoglan I, *"The green vehicle routing problem: A heuristic based exact solution approach".* Applied Soft Computing 39 (2016), pp.154-164.
[35] Kouridis C, Gkatzoflias D, Kioutsioukis I, Ntziachristos L, Pastorello C and Dilara P, *"Uncertainty estimates and guidance for road transport emission calculations",* Technical report, European Commission Joint Research Center Institute for Environment and Sustainability.
[36] Küçükoğlu I , Ene S, Aksoy A and Öztürk N, *"A green capacitated vehicle routing problem with fuel consumption fuel optimization model",* International Journal of Computational Engineering Research 3 (7) (2013), pp.16-23.
[37] Kuo Y, *"Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem",* Computers and Industrial Engineering 59 (2010), pp.157-165.

[38] Kramer R, Maculan N, Subramanian A and Vidal T, *"A speed and departure time optimization algorithm for the pollution-routing problem", European* Journal of Operational Research Society 247 (2015a), pp.782-787.
[39] Kramer R, Subramanian A, Vidal T and F. Cabral L.A, *"A metaheuristic approach for the Pollution-Routing Problem*", European Journal of Operational Research Society 243 (2015b), pp.523-539.
[40] Kwon Y.J, Choi Y.J and Lee D.H, *"Heterogeneous fixed fleet vehicle routing considering carbon emission",* Transportation Research Part D, 2 (2016), pp.81-89.
[41] Lahyani R, Khemakhem M and Semet F, *"Rich Vehicle Routing Problem: From a taxonomy to a definition"*, European Journal of Operational Research, 241 (2015), pp.1-14.
[42] Li H, Lv T and Li Y, *"The tractor and semitrailer routing problem with many-to-many demand considering carbon dioxide emissions",* Transportation Research Part D, 34 (2015), pp.68-82.
[43] Lin C, G.T.S.Ho Ch, Chung S.H and Lam H.Y, *"Survey of Green Vehicle Routing Problem: Past and future trends",* Expert Systems with Applications, 41 (4) (2013), pp.1118-1138.
[44] Mancini S, *"The hybrid vehicle routing problem",* Optimization Online, OR and Management Sciences, July2015.
[45] Montoya A, Gueret CH, Mendoza J.E and Villegas J.G, "A multi-space sampling heuristic for the green vehicle routing problem", Transportation Research Part C: Emerging Technologies, (2015), In press.
[46] Palmer A, *"The development of an integrated routing and carbon dioxide emissions model for goods vehicles",* Ph.D. thesis, Canfield University, School of Management.
[47] Qian J and Eglese R, "Fuel emissions optimization in vehicle routing problems with time-varying speeds", European Journal of Operational Research 248 (2016), pp.840-848.
[48] Reinsman A, *"Management science knowledge: It's creation generalization and consolidation"*, Westport, CT: Quorum Books Publishing Company (1992).
[49] Russell R, *"An effective heuristic for the M-Tour traveling salesman problem with some side conditions",* Operations Research, 25 (1977), pp.517-524.
[50] Sbihi A and Eglese R.W, *"Combinatorial optimization and Green Logistics", 4OR: A Quarterly Journal of Operations Research, 5 (2007a), pp.99-116.*
[51] Sbihi A and Eglese R.W, *"The relationship between Vehicle Routing & Scheduling and Green Logistics- A literature survey",* Department of Management Science, Lancaster University Management School, LA14YX, UK (2007b).
[52] Schneider M, Stenger A and Goeke D, *"The electric vehicle routing problem with time windows and recharging stations",* Technical report, University of Kaiserslautern, Germany (2012).
[53] Scora G and Barth M*, "Comprehensive Modal Emission Model (CMEM), Version 3.01. User's guide"*,Technical Report (2006), University of California, USA. http://www.cert.ucr.edu/cmem/docs/CMEM_User_Guide_v3.01d.pdf > (04-05-2016)
[54] Scott C, Urquhart N and Hart E, "Influence of topology and payload on CO2 optimized vehicle routing", Springer- Verlag Berlin Heidelberg (2010).
[55] Soysal M, Bloemhof-Ruwaard J.M and Bektas T, *"The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations",* International Journal of Production Economics 164 (2015), pp.366-378.
[56[Suzuki Y, *"A new truck-routing approach for reducing fuel consumption and pollutants emission",* Transportation Research Part D, 16 (2011), pp. 73-77.
[57] Taha M, Fors M.N and Shoukry A.A, *"An exact solution for a class of green vehicle routing problem",* Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management, Bali, Indonesia, January 7-9,(2014).
[58] Tajik N, Tavakkoli-Moghaddam R, Vahdani B and Mousavi M, *"A robust optimization approach for pollution routing problem with pickup and delivery under uncertainty",* Journal of Manufacturing System 33 (2014), pp.277-286.

[59] Tavares G, Zsigraiova Z, Semiao V and Carvalho M.G, *"A case study of fuel savings through optimization of MSW transportation routes",* Management of Environmental Quality: An international Journal 19 (4) (2008), pp.444-454.
[60] Tillman F.A, *"The multiple terminal delivery problem with probabilistic demands",* Transportation Science 3 (1969), pp.192-204.
[61] Tiwari A and Chang P.CH, *"A block recombination approach to solve green vehicle routing problem",* International Journal of Production Economics 164 (2015), pp.379-387.
[62] Wang F, Lai X and Shi N*, "A multi-objective optimization for green supply chain network design"*, Decision Support Systems, 51 (2011), pp.262-269.
[63] Wilsons H and Weissberg H, *"Advanced dial-a-ride algorithms research project: final report"*, Technical report. R76-20,(1967), Departement of Civil Engineering, MIT, Cambridge, MA.
[64] Xiao Y and Konak A, *"A simulating annealing algorithm to solve the green vehicle routing and scheduling problem with hierarchical objectives and weighted tardiness",* Applied Soft Computing , 34 (2015a), pp 372-388.
[65] Xiao Y and Konak A, *"Green Vehicle Routing Problem with time-varying traffic congestion",* 14th INFORMS Computing Society Conference Richmond, Virginia, January 11–13, (2015b), pp. 134–148.
[66] Xiao Y, Zhao Q, Kaku I and Xu Y, *"Development of a fuel consumption optimization model for the capacitated vehicle routing problem",* Computers and Operational Research, 39 (2012), pp 1419-1431.
[67] Yang B, Hu Z.H, Wei CH, Li S.Q and Jia SH, *"Routing with time-windows for multiple environmental vehicle types",* Computers and Industrial Engineering 89 (2015), pp.150-161.
[68] Zare-Reisabadi E and Mirmohammadi S.H, *"Site dependent Vehicle Routing Problem with soft time window: Modeling and solution approach",* Computers and Industrial Engineering 90 (2015), pp.177-185.
[69] Zhang Z.H, Wei L and Lim A, *"An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints",* Transportation Research Part B, 82 (2015a), pp.20-35.
[70] Zhang Z.H, Zhao Y, Xue W and Li J, *"Vehicle routing problem with fuel consumption and carbon emission",* International Journal of Production Economics 170 (2015b), pp.234-242.

Hub Location and Routing Problem: a Variable Neighborhood Decomposition Search based solving approach

F.F.C. Fontes[1,2] and G. Goncalves[1]

[1] Univ. Artois, EA 3926, Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), Béthune, F-62400, France
gilles.goncalves@univ-artois.fr
[2] Universidade Federal Rural do Semi-Árido Departamento de Ciências Exatas e Naturais CEP: 59625-900, Mossoró RN, Brazil
fabio_fontes@ufersa.edu.br

Abstract. Global transport of goods currently is prioritized fundamental points, for example, customer satisfaction, CO_2 emission and delivery short time. Efficient network design is fundamental to be met the important factors of supply chain network and to keep a competitive advantage. In this work is showed a Variable Neighborhood Decomposition Metaheuristic using exact methods to solve a new structure of hub location and routing problem. Network is composed by a hierarchical structure with hubs, sub-hubs and spoke nodes representing liner shipping operations. Results with small instances are available and compared with the ones provided by the Cplex solver, trying evaluate the efficiency of the matheuristic on the studied problem.

Mots-Clefs. VNDS matheuristic, exact methods, hub location and routing problem, sub-hubs, liner shipping operations.

1 Introduction

Development of an efficient and sustainable supply network, nowadays, is accounted an important factor when it is desired to obtain/maintain a competitive advantage in the logistic scenario globalized.

When the network design problem is solved in an optimized way, can be deployed, more easily, a supply service; decreased the expenses with fuel, quantity of vehicles and crews; met customers at a shorter time, allowing a greater customer satisfaction; and decreased the CO_2 emissions.

Liner shipping operations, responsible by transport of goods at containers, have the network formed by hub route (deep sea operations), where hub ports are met by big vessels in a cyclic service form, and regional routes (short sea operations) where the cyclic service is realized by short sea vessels linking regional ports (spokes) with its respective hub port.

A hub location and routing problem is presented in [1], where each hub can have more than one cyclic regional route. Moreover in [2] a model with cyclic regional routes composed of a limited number of spokes, is proposed.

New concept of hub and spoke network is presented at [3] and [4], where were proposed mathematical models of a hierarchical structure of hub and spoke with sub-hubs. In the first one, hub location and routing problem was represented by one network with cyclic routes at hub level and at regional level, representing deep sea and short sea service, respectively. The second one, hub route is represented by a pendular form [5] and only regional routes are in cyclic form.

According to [6], such hub network design formulations are very difficult to solve.

In this work the concept of sub-hubs is also used, however the mathematical model developed allows to form clusters (hub and its set of ports allocated) with an isolated hub or with only the hub and one spoke.

Trying to prove the efficiency of sub-hubs at the structure of hub and spoke, the results of developed model are compared with the ones provided by a mathematical model with the same characteristics but without sub-hubs.

Location, allocation and service design problems are explored by the models, together with routing cargo problem, where demands are delivery from each port to all others.

Due the complexity of studied problem, optimum results for instances with more than 12 nodes were not obtained at the realized tests with CPLEX. Therefore, it was implemented a variant of Variable Neighborhood Search (VNS), knows as Variable Neighborhood Decomposition Search (VNDS) [7] to analyze the two models with large instances.

In this work is available the results of Cplex with the results of VNDS, aiming to evaluate the efficiency of matheuristic on the studied problem.

2 Hub Location and Routing Problem

Network design is considered an important strategic decision in liner shipping transport because the routes are published and may not be changed by several years [5].

Hub and spoke structure is formed by a group of nodes representing hubs, where remaining nodes (spokes) are allocated to these hubs. Cargos of different points are collected, classified and distributed in these hub ports.

Liner shipping operations, transhipment is realized at a hub port because it connects a hub route with a regional route [8], allowing regional ports to send goods to others regions and to receive goods from others regions.

Sub-hub nodes are spoke nodes because they are allocated to hub nodes, but they are also considered as transhipment points because they connect two near regional routes, allowing, for some situations, the direct transport of goods between two clusters without the hub route using.

Though economies of scale can be offered by the use of hub routes, the reduced path and reduced delivery time could be achieved by the transport using sub-hubs, trying minimize the global network cost.

As in [3] and [4], a second mathematical model representing a hub and spoke network for deep sea and short sea service without sub-hubs has been developed in this work. Results between two models are available trying to prove that model with sub-hubs is more efficient.

New structure of hub and spoke with sub-hubs is proposed in this work, where isolated ports with big demand can be represented by a hub without no spokes. Different, also, of cited problems, where each cluster was composed of one hub and two spokes at least. in your new proposed model one cluster can be constituted with one hub and just one spoke node.

Examples of analyzed networks are presented Fig.1 and Fig.2, where hub ports are represented by squares, spokes are the circle nodes, triangles represent sub-hubs, dashed lines are the deep sea services and continuous lines represent short sea services.

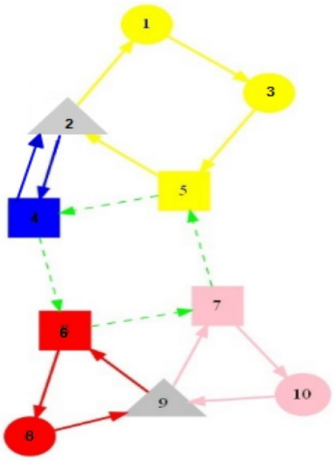


Fig. 1. Network with sub-hubs

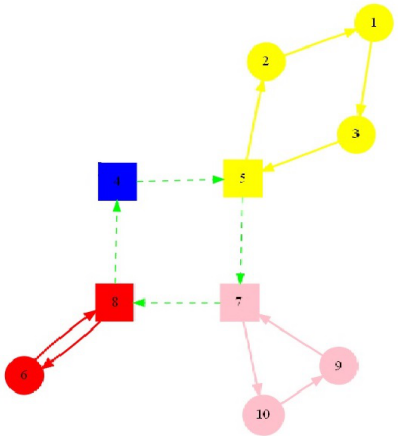


Fig. 2. Network without sub-hubs

Complex problems are presented by two models because they are also constituted by flows of goods at the network, where products are sent from each node to all others nodes. Number

of alternative paths to transport goods is increased by model with sub-hubs, increasing also the complexity of problem.

These problems can be solved by metaheuristic method offering good results at a viable execution time.

2.1 Mathematical Model

Mathematical model developed in this work is a binary integer linear programming for the hub location and spoke allocation problem together with the service design problem and the routing problem.

Let V be the set of nodes with n nodes, c_{ij} is the cost per unit of flow from node i to node j where $i,\ j \in V$, q is the quantity of hubs, fl_{ij} is the flow from node i to node j and α is the discount factor on the unit cost of flow between hubs.

Hub location, sub-hub location and spoke allocation are defined by variables of model: $u_i = 1$ if the vertex i is a hub without spoke allocated for it, or 0, otherwise, $x_{ij} = 1$, if the vertex i is allocated to hub j, or 0, otherwise, and $t_i = 1$, if i is sub-hub, or 0, otherwise.

The variable $y_{ij} = 1$, if there exists the arc-non-hub $(i;j)$, or 0, otherwise, and $b_{ij} = 1$, if there exists the arc-hub $(i;j)$, or 0, otherwise, are responsible by the service design.

While the variables $w_{ijkl} = 1$, if there exists a flow from i to j routed via arc-hub $(k;l)$, or 0, otherwise and $s_{ijkl} = 1$, if there exists a flow from i to j routed via arc-non-hub $(k;l)$, or 0, otherwise, correspond to the cargo routing.

The model can be defined as follows:

$$min \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} fl_{ij}\ c_{ij}\ x_{ij} + \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} \sum_{k=1,k\neq j}^{n} \sum_{l=1,l\neq k,l\neq i}^{n} fl_{ij}\ c_{kl}\ s_{ijkl} + \\ + \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} \sum_{k=1,k\neq j}^{n} \sum_{l=1,l\neq k,l\neq i}^{n} fl_{ij}\ \alpha\ c_{kl}\ w_{ijkl}$$

$s.t.:$

$$\sum_{j=1}^{n} x_{jj} + \sum_{j=1}^{n} u_j = q \tag{1}$$

$$x_{i,j} \leq x_{jj}, \qquad \forall i,j \in V \tag{2}$$

$$t_i \leq \sum_{j=1,j\neq i}^{n} x_{ij}/2, \qquad \forall i \in V \tag{3}$$

$$u_i + t_i + x_{ii} \leq 1, \qquad \forall i \in V \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} = 1 + t_i - u_i, \qquad \forall i \in V \tag{5}$$

$$x_{ij} + x_{ik} + x_{lj} + x_{lk} \leq 3, \qquad \forall i,j,k,l \in V, \\ i \neq j, k \neq j, k \neq i, l \neq j, l \neq k, l \neq i \tag{6}$$

$$y_{ij} + t_i + t_j \leq 2, \qquad \forall i,j \in V,\ i \neq j \tag{7}$$

$$y_{ij} + x_{ik} + x_{jl} \leq 2 + x_{jk} + x_{il}, \quad \forall i,j,k,l \in V,\ i \neq j, k \neq j, i \neq l, l \neq k \tag{8}$$

$$x_{kj} + x_{lj} + y_{ik} + y_{il} \leq 3, \quad \forall i,j,k,l \in V, \ j \neq i, k \neq i, l \neq i, l \neq k \tag{9}$$

$$y_{ij} + y_{ji} \leq 1 + x_{ii} + x_{jj}, \quad \forall i,j \in V,\ i \neq j \tag{10}$$

$$\sum_{j=1, j\neq i}^{n} y_{ij} = 1 + t_i - u_i, \quad \forall i \in V \tag{11}$$

$$\sum_{j=1, j\neq i}^{n} y_{ji} = 1 + t_i - u_i, \quad \forall i \in V \tag{12}$$

$$\sum_{j=1, j\neq i}^{n} b_{ij} = x_{ii} + u_i, \quad \forall i \in V, \tag{13}$$

$$\sum_{j=1, j\neq i}^{n} b_{ji} = x_{ii} + u_i, \quad \forall i \in V, \tag{14}$$

$$b_{ij} + b_{ji} \leq 1 + 2/q, \quad \forall i,j \in V,\ i \neq j \tag{15}$$

$$\sum_{l=1, l\neq i}^{n} w_{ijil} + \sum_{l=1, l\neq i}^{n} s_{ijil} = 1, \quad \forall i,j \in V,\ j \neq i \tag{16}$$

$$\sum_{l=1, l\neq j}^{n} w_{ijlj} + \sum_{l=1, l\neq j}^{n} s_{ijlj} = 1, \quad \forall i,j \in V,\ j \neq i \tag{17}$$

$$\sum_{l=1, l\neq i, l\neq k}^{n} w_{ijkl} + \sum_{l=1, l\neq i, l\neq k}^{n} s_{ijkl} = \sum_{l=1, l\neq j, l\neq k}^{n} w_{ijlk} + \sum_{l=1, l\neq j, l\neq k}^{n} s_{ijlk} \quad \forall i,j,k \epsilon V,\ k \neq i, k \neq j, j \neq i \tag{18}$$

$$s_{ijkl} \leq y_{kl} \quad \forall i,j,k,l \in V, \ l \neq k, l \neq i, k \neq j, j \neq i \tag{19}$$

$$w_{ijkl} \leq b_{kl} \quad \forall i,j,k,l \in V, \ l \neq k, l \neq i, k \neq j, j \neq i \tag{20}$$

$$x_{ij},\ u_i,\ t_i,\ y_{i,j},\ b_{ij},\ w_{ijkl},\ s_{ijkl} \in \{0,1\} \quad \forall i,j,k,l \in V \tag{21}$$

The objective function minimizes the allocation cost and the overall transportation cost.

About allocation constraints there are: constraints (1), where the quantity of hubs is equal to q. Two different types of hubs are allowed by the model, hubs with allocations x_{jj} and standalone hubs (hub with no spokes) u_j; Constraints (2), where a vertex i just will be allocated to a vertex j if j is a hub with allocation; Constraints (3), where a vertex i just will be a sub-hub if it is allocated for 2 hubs; Constraints (4), where every vertex may be a hub with allocation, or a hub without allocation, or a spoke. And a vertex just will be able to be a sub-hub if it is not a hub;

Constraints (5), where every vertex i will be allocated to 1 hub j, or to 2 hubs j in the case of i to be a sub-hub. But it can be also a hub without allocation; Constraints (6), where 2 sub-hubs can not be allocated for the same hubs.

For constraints representing the service design, where is created the arcs linking the nodes, there exists: Constraints (7), where these constraints do not permit an arc y_{ij} linking 2 sub-hubs; Constraints (8), where these constraints will not permit the existence of an arc linking the vertex i and j if the vertex i and j are allocated to different hubs. Just will exist arc linking the vertex i and j if they are allocated to the same hub. Then, arc y_{ij} also will not link hubs. Or will link spokes or 1 hub with 1 spoke; Constraints (9), where these constraints will not allow two arcs going out of a vertex i to two other vertex that belong to a same hub j; Constraints (10), where the network uses the arc-non-hub (i, j) or the arc-non-hub (j, i). But if the cluster has just one spoke, it is possible to use the arcs-non-hub (i, j) and (j, i); Constraints (11), where of every vertex i goes out 1 arc-non-hub, but if i is a sub-hub, it goes out 2 arcs. However, if the vertex i is a hub without allocation, it is not possible to go out 1 arc-non-hub; Constraints (12), where every vertex i will receive 1 arc-non-hub, but if i is a sub-hub, it will receive 2 arcs. However, if the vertex i is a hub without allocation, it is not possible to receive 1 arc-non-hub; Constraints (13), where of every vertex hub i goes out 1 arc-hub; Constraints (14), where every vertex hub i will receive 1 arc-hub; Constraints (15), where the network uses the arc-hub (i, j) or the arc-hub (j, i). But if the problem has just 2 hubs, it is possible to use both arcs (i, j) and arc (j, i).

Cargo routing constraints, in the built network, is done by: Constraints (16), where the flow of every vertex i toward to every vertex j will leave the source using the arc (i, l), where the flow will can be a flow w (flow in a arc-hub) or a flow s (flow in a arc-non-hub). Arc-hub links 2 hubs and arc-non-hub links a hub with a spoke or links two spokes; Constraints (17), where the flow of every vertex i toward to every vertex j will arrive the destination using the arc (l, j), where the flow will can be a flow w (flow in a arc-hub) or a flow s (flow in a arc-non-hub); Constraints (18), where all flows that leave a intermediate vertex k (it is not the source or the destination), will be equal to the flows that have arrived to this vertex; Constraints (19), where the flow s just will exist if there exists the arc y; Constraints (20), where the flow w just will exist if there exists the arc-hub b.

Constraints (21) are integrality constraint.

For problems with 6 or more hubs, it is possible to form subtour at the hub-route. Constraints to eliminate subtours are added in the previous model.

For the same reasons, at regional route, constraints to eliminate subtours are added in the previous model when a cluster has 5 nodes or more.

Then, for each supposedly viable solution found, H will be the set of selected hub nodes with q nodes, C will be the set of selected clusters with $C = \{C_1, C_2, ..., C_m\}$, C_p will be the set of selected nodes which belong to the same cluster p and G will be any subset of nodes from a set H or from a set C_p. The constraints are:

$$\sum_{i \in G, j \in H-G} b_{ij} \geqslant 1, \qquad 2 \leq \mid G \mid \leq \mid H \mid -2, \quad with \mid H \mid = q \tag{22}$$

$$\sum_{i \in G, j \in C_p - G} y_{ij} \geqslant 1, \qquad 2 \leq \mid G \mid \leq \mid C_p \mid -2,$$

$$\forall\, C_p, \quad where\ p \in \{1, 2, ..., \sum_{j=1}^{n} x_{jj}\} \tag{23}$$

Where constraints (22) are included whenever necessary, to eliminate subtours at hub route and constraints (23) are included whenever necessary, to eliminate subtours at regional routes.

3 Variable Neighborhood Search (VNS)

Variable neighborhood search (VNS) is a metaheuristic whose basic idea is systematic change of neighborhood within a local search [9].

Systematic change of neighborhood is based on the fact that a local optimum for one neighborhood structure can not be the local optimum for another neighborhood; global optimum found,

will be a local optimum to all neighborhoods; and for many problems local optimum with respect to one or several neighborhoods are relatively close to each other structures [9], [10].

[7]is presented the ingredients used to build different variants of VNS. Between the variants, there is the Variable Neighborhood Decomposition Search(VNDS), in which is used the improvement phase ingredient just in a reduced part of problem.

VNDS is not yet very used by researchers. But there are some works at literature, where the VNDS is used to solve, for example, the supply chain management planning problems [11], a large-scale continuous location allocation problems [12], 0-1 multidimensional knapsack problem [13], the p-median problem [14] and the minimum weighted k-cardinality tree problem [15].

3.1 Proposed Variable Neighborhood Decomposition Search (VNDS)

In this study, the problem is composed of 4 sub-problems: hub and sub-hub location, spoke allocation, service design and network routing.

Operators of a Variable Neighborhood Decomposition Search (VNDS), algorithm (1), were implemented considering the characteristics of those sub-problems. Moving between operators is guided by neighborhood change sequential procedure, algorithm (2).

Algorithm 1 Variable Neighborhood Decomposition Search

Initialization
  Find a Initial solution $x \in S$ by a constructive heuristic
  Define the order of operators in the set $N = N_1, ..., N_{k_{max}}$
  $x_{actual} \leftarrow x$
repeat
  $k \leftarrow 1$
  while $k \leq k_{max}$ **do**
    step(1) - Getting subproblem(N_k):
      select exactly p solution attributes from previous solution x_{actual};
      denote this subset by y;
    step(2) - Solving subproblem:
      $y' \leftarrow$ Improvement procedure(y); //y' improved solution
    step(3) - Substituting the solution (it creates original problem solution):
      $x_{best} = (x_{actual} \setminus y) \bigcup y'$;
    step(4) - Making the problem cost evaluation:
      objective function $f(x_{best}) = f_A(x_{best}) + f_T(x_{best})$;
    step(5) - Moving between operators problem:
      Neighborhood change sequential $(x_{actual}, x_{best}, k)$;
  end while
  $x_{actual} \leftarrow Shake(x_{best}, N_{special})$
until the stopping condition is met

For location sub-problem was developed 1 operator responsible for search of a local optimum for hubs and sub-hubs location. Concerning the sub-problem of spoke nodes allocation, two operators were defined. First operator is responsible for getting the local optimum at the allocation of each spoke. The second operator ($N_{special}$) algorithm (1), was attributed the function of shake at the matheuristic, allowing the allocation on a different cluster of an aleatory spoke node. For service design sub-problem were developed 2 operators and about the routing subproblem was used Dijkstra algorithm.

At the proposed problem is allowed to have between 1 route (composed of only standalone hub nodes) and $n + 1$ routes (n is the number of hubs). Therefore, 2 operators were developed for the service design sub-problem.

Cost of 2 routes, randomly chosen, can be improved by a first operator of that sub-problem. This operator was considered by the fact that the routing cost of demands is minimized by the synchronization of services between each cluster and at hub level.

While cost of only 1 route, randomly chosen, is improved by a second operator. Using this operator is allowed to escape of possible local optimum achieved by the previous one.

Algorithm 2 Procedure Moving Between Operators for VNDS Metaheuristic

Procedure Neighborhood_change_sequential(x, x', k)
if $f_T(x') < f_T(x)$ **then** //$f_T(x')$ is the function of transport cost
    $x \leftarrow x'$;
    $k \leftarrow 1$; //return to first neighborhood
else
    $k \leftarrow k + 1$;
end if

Dijkstra algorithm was used at the routing subproblem because offers the smallest path at the network for the demands from each node to all others nodes.

About the *improve hub and improve sub-hub* operator of the step 2 of algorithm 1, several neighborhood structures are explored through of Cyclic Variable Neighborhood Descendent (C-VND) procedure.

Algorithm 3 Cyclic VND

Initialization
    Select the set of neighborhood structures N_L, for $L = 1, ..., \mathrm{L}_{max}$;
    Consider the previous solutions y;
    Choose the function f;
repeat
    $L \leftarrow 1$
    $y' \leftarrow y$
    while $L \leq L_{max}$ **do**
        $y'' \leftarrow argmin_{z \in N_L(y)} f(z)$ //z is a solution of neighborhood of y, where f(z) has minimum value.
        Change Neighborhood (y, y'', L);
    end while
until $f(y') \leq f(y)$ (no improvement is obtained)
return y';

The set of neighborhood structures of Cyclic VND is composed by 2 neighborhood structures (*improve-hub*, *improve-sub-hub*).

At the *improve-hub* structure is tested, for each cluster, the switch between each spoke node with its previous (original) hub (new hub is allocated). Hub route is built by the Nearest Neighbor Algorithm for each switch hub-spoke, allowing evaluate the sub-problem cost.

Improve-sub-hub structure is evaluated, for each neighbor cluster, the distance cost between each candidate sub-hub (spoke node from cluster neighborhood) and the hubs from neighborhood. Each spoke node is tested, trying to find a new sub-hub with distance cost less than actual sub-hub.

Procedure Change Neighborhood used at the Cyclic VND is the Cyclic neighborhood change step.

Algorithm 4 Procedure Change Neighborhood for Cyclic-VND Algorithm

Procedure Neighborhood_change_cyclic(y, y'', L)
$L \leftarrow L + 1$
if $f(y'') < f(y)$ **then**
    $y \leftarrow y''$;
end if

Two route improve operator, at the step 2 (Improvement Procedure), on VNDS (algorithm 1), 2 random and different routes are selected and, for each one, the 2-opt Algorithm described by [7] is applied. In this operator, the cost evaluated by the sub-problem is total cost of selected route.

Algorithm 2-opt was implemented to select the neighbor solution of y with the smallest cost, independently if the global cost is better than the actual solution.

One route improve operator one route was created with the same steps of last one. Just that, one random route is selected. In this operator is allowed to escape of possible local optimum achieved by the previous one.

For problems where all nodes are hubs, can not be improved by *two routes improve* operator, but can be improved by *one route improve* operator.

Location-allocation sub-problem has been solved using p-median approach during the constructive heuristic. Because in your problem the allocation will be dependent not only of distance between hub and spoke, but also of the demand flows at network nodes, one *allocation spoke nodes improve* operator has been developed to improve the allocation.

Although allocation of spoke nodes has been realized by an exact method (solving the p-median problem), initial solution is provided by a constructive heuristic. Note that in our model, the allocation will be dependent not only of distance between hub and spoke, but also of the demands at network nodes.

Allocation spoke nodes operator was developed to improve the allocation when thought for all network and not only inside each cluster.

In this operator, the subproblem evaluated is the transport cost after selecting one random spoke node and trying to insert it into a different cluster of its original cluster. After to test the random spoke for each cluster, best improvement of search will be accepted only if verified that it is better than the actual solution.

Allocation spoke node operator is closed after the first best improvement to be accepted or after a specific number of tests without improvement.

Objective function in this problem is composed by minimize two parts, allocation cost and transport cost. Step (4) at VNDS (algorithm 1) is necessary to be solved the smallest path problem (routing subproblem) using Dijkstra algorithm to evaluate transport costs.

Transport and allocations costs are calculated at the step (4), but at the step (5), Neighborhood change sequential procedure, only the transport cost is considered. This choice was taken because one observed that transport cost is more representative than allocation cost in the objective function.

Dijkstra algorithm is used at the step (4), except when executed the *allocation spoke node* operator, because in this case, at the subproblem, is evaluated the transport cost.

This heuristic implemented can be classified as a matheuristic because it is used exact methods inside some procedures of the algorithm.

Shake procedure is used at VNS with goal to diversify the search, getting away from a local optimum, and maybe, allowing to find the global optimum.

Operator used at shake procedure has the same structure of the *allocation spoke node* operator, except that the first best improvement solution is accepted.

4 Computational Results

The computational tests were executed with a processor Intel Core i7 with 3.0 GHz and 8 GB of memory. Matheuristic and the mathematical model were implemented using Cplex Concert Technology C++.

Mathematical model without sub-hubs, (Fig. 2), was created in this work to compare this basic model with the one with sub-hubs.

For this model version without sub-hubs, it was removed the constraints (3), (6), (7) and (9) and changed the constraints (4), (5), (8), (11) and (12) of mathematical model.

Sequence of operators used at VNDS was: *improve hub and improve sub-hub*, *two routes improve*, *one route improve* and *allocation spoke nodes improve*. Whereas, the sequence of neighborhood structures used at VND, inside of *improve hub and improve sub-hub* operator was: *improve-hub* and *improve-sub-hub*.

All instances were generated. The results are presented in the table 1:

Where OF is the objective function value and ET is the execution time. The execution time is in seconds. The execution time limited for Cplex was 86400 as for VNDS was 9000.

VNDS was executed 30 times for each instance. Then, the objective function and the execution time presented at table 1 are the average of all executions.

Table 1. Results with Cplex and VNDS for the two models

Test	Model	n	q	α	O.F. (Cplex)	E.T. (Cplex)	O.F. (VNDS)	E.T. (VNDS)	Percentage Deviation	Standard Deviation
1	with sub-hubs	7	3	0.3	332726	44.31	437272	8.74	31.42	845.35
2	without sub-hubs	7	3	0.3	355191	24.61	437383	8.74	23.14	1016.68
3	with sub-hubs	7	3	0.6	443089	127.69	546845	8.26	23.41	0
4	without sub-hubs	7	3	0.6	488832	61.28	546845	8.74	11.87	0
5	with sub-hubs	7	3	0.9	535382	438.7	656640	8.03	22.65	0
6	without sub-hubs	7	3	0.9	583903	106.05	656640	7.92	12.45	0
7	with sub-hubs	7	4	0.3	273331	10.89	273331	8.02	0	0
8	without sub-hubs	7	4	0.3	300956	36.91	317349	8.73	5.45	1010.5
9	with sub-hubs	7	4	0.6	409166	161.67	409166	8.04	0	0
10	without sub-hubs	7	4	0.6	477442	332.2	490448	8.55	2.72	0
11	with sub-hubs	7	4	0.9	501977	1180.28	532169	8	6.01	0
12	without sub-hubs	7	4	0.9	586352	372.66	664358	8.73	13.3	0
13	with sub-hubs	9	3	0.3	484099 gap:4.55	86400	463198	8.05	-4.31	0
14	without sub-hubs	9	3	0.3	478477	4661.67	478477	8.75	0	0
15	with sub-hubs	9	3	0.4	—	—	497994	7.94	—	0
16	without sub-hubs	9	3	0.4	522050	939.88	522050	8,75	0	0
17	with sub-hubs	9	3	0.6	—	—	567586	8.63	—	0
18	without sub-hubs	9	3	0.6	609196	923.41	609196	7.94	0	0
19	with sub-hubs	9	4	0.3	388768	46569.58	418723	8.74	7.71	0
20	without sub-hubs	9	4	0.3	395103	908.74	395103	8.75	0	0
21	with sub-hubs	9	4	0.6	—	—	560854	8.71	—	2974.11
22	without sub-hubs	9	4	0.6	560864	3438.98	560864	8.75	0	0
23	with sub-hubs	10	4	0.3	—	—	587757	8.68	—	0
24	without sub-hubs	10	4	0.3	597835	1934.36	597835	8.1	0	0
25	with sub-hubs	10	4	0.6	—	—	808090	8.01	—	0
26	without sub-hubs	10	4	0.6	923419	27756.89	923419	7.9	0	0
27	with sub-hubs	11	4	0.3	731891 gap:11.33	86400	724193	8.02	-1.05	0
28	without sub-hubs	11	4	0.3	756180	15030.92	756180	8,01	0	0
29	with sub-hubs	12	4	0.3	—	—	785332	8.02	—	0
30	without sub-hubs	12	4	0.3	876251 gap:15.54	86400	837008	8.03	-4.48	0

5 Conclusion

Instances with 9 nodes, at the model without sub-hubs, the optimal solution was found for each instance. Model with sub-hubs just the test 19 was not presented good results. For all others, the VNDS results (objective function) of model with sub-hubs was better than the results (objective function) of the model without sub-hubs.

Instances with 10 and 11 nodes, the optimal was also found at all tests with the model without sub-hubs. VNDS showed, also, better results of the objective function for the model with sub-hubs.

Test 27, the VNDS found a result of objective function better than Cplex, because the Cplex stopped with a gap of 11.33%. But the Cplex result is better than the optimum result of model without sub-hubs (test 28).

Test 30, Cplex stopped with a gap of 15.5% and the VNDS showed a better result.

Test 29, VNDS showed result better than the test 30.

Only tests with 7 nodes were not presented good results because the VNDS found the optimum just at tests 7 an 9.

Instances of model with sub-hubs where the Cplex did not find results with small gap were not presented (tests 15, 17, 21, 23, 25 and 29). But the objective functions values found with VNDS are better than the optimum of model without sub-hubs. This information confirms the quality of network with sub-hubs for small instances.

Implemented VNDS, where differents parts of analyzed problem are explored by sub-problems, showed good results for the analyzed instances and could be used with larger instances in order to confirm if the network with sub-hubs is more efficient than a network without sub-hubs.

Acknowledgement

The authors would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES), a foundation of the Ministry of Education of Brazil that finances a PhD scholarship for the development this study.

References

1. Ricardo Saraiva de Camargo, Gilberto de Miranda, and Arne Løkketangen. A new formulation and an exact approach for the many-to-many hub location-routing problem. *Applied Mathematical Modelling*, 37(12):7465–7480, 2013.
2. Inmaculada Rodríguez-Martín, Juan-José Salazar-González, and Hande Yaman. A branch-and-cut algorithm for the hub location and routing problem. *Computers & Operations Research*, 50:161–174, 2014.
3. Fabio Francisco da Costa Fontes and Gilles Goncalves. Hub location and routing problem with alternative paths. In *Advanced Logistics and Transport (ICALT), 2015 4th International Conference on*, pages 317–322. IEEE, 2015.
4. Fabio Francisco da Costa Fontes and Gilles Goncalves. Routing problem with pendular and cyclic service in a hierarchical structure of hub and spoke with multiple allocation of sub-hubs. In *Industrial Engineering and Systems Management (IESM), 2015 International Conference on*, pages 561–567. IEEE, 2015.
5. Marielle Christiansen, Kjetil Fagerholt, Bjørn Nygreen, and David Ronen. Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483, 2013.
6. Teodor Gabriel Crainic, Kap Hwan Kim, et al. Intermodal transportation. *Transportation*, 14:467–537, 2006.
7. Pierre Hansen, Nenad Meladenovic, Raca Todosijevic, and Said Hanafi. Variable neighborhood search, basics and variantes. *Cahiers du GERAD*, 2016.
8. Martin W Andersen and DTU Transport. *Service Network Design and Management in Liner Container Shipping Applications*. DTU Transport, 2010.
9. P Hansen, N Mladenovic, J Brimberg, and JA Moreno-Pérez. Variable neighbourhood search, Handbook of Metaheuristics, Gendreau, M., Potvin, J.-Y. *International Series in Operations Research & Management Sciences*, 146:61–86, 2010.
10. Pierre Hansen and Nenad Mladenovic. *A tutorial on variable neighborhood search*. Groupe d'études et de recherche en analyse des décisions, HEC Montréal, 2003.
11. Miguel A Lejeune. A variable neighborhood decomposition search method for supply chain management planning problems. *European Journal of Operational Research*, 175(2):959–976, 2006.
12. Jack Brimberg, Pierre Hansen, and Nenad Mladenović. Decomposition strategies for large-scale continuous location–allocation problems. *IMA Journal of Management Mathematics*, 17(4):307–316, 2006.
13. Saïd Hanafi, Jasmina Lazić, Nenad Mladenović, Christophe Wilbaut, and Igor Crévits. Hybrid variable neighbourhood decomposition search for 0-1 mixed integer programming problem. *Electronic Notes in Discrete Mathematics*, 36:883–890, 2010.
14. Pierre Hansen, Nenad Mladenović, and Dionisio Perez-Britos. Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335–350, 2001.
15. Dragan Urošević, Jack Brimberg, and Nenad Mladenović. Variable neighborhood decomposition search for the edge weighted k-cardinality tree problem. *Computers & Operations Research*, 31(8):1205–1213, 2004.

Author Index

Sponsors


informatiques mathématiques
Inria

cnrs

Université
de Lille

EMI

gdr
macs
CRIStAL
Centre de Recherche en Informatique,
Signal et Automatique de Lille

SYNERGY
Horizon 2020
GA No 692286

GdR PO

LIRIMA