Impact of Explicit Memory on Dynamic Conflict Resolution

Sarah Degaugue, Nicolas Durand, Jean-Baptiste Gotteland

Impact of Explicit Memory on Dynamic Conflict Resolution

Sarah Degaugue, Nicolas Durand, Jean-Baptiste Gotteland

ENACLab/OPTIM
7 av. Édouard Belin
31055 Toulouse Cedex, France
sarah.degaugue, nicolas.durand, jean-baptiste.gotteland @enac.fr

Air Traffic Control, Conflict Detection And Resolution, Dynamic Evolutionary Algorithm

***Abstract*—**
Due to uncertainties in weather conditions, trajectory prediction and constant flight evolution, controlling traffic in a sector is a dynamic problem. Furthermore, when the traffic increases, Air traffic control can become a complex dynamic optimization problem difficult to handle by human operators. In the context of offering air traffic controllers intelligent decision support tools adapted to the dynamic nature of traffic, we compare two options to address this issue. Previous work has already used an evolutionary algorithm to solve conflicts at given time steps. In this paper, we compare two different approaches using this evolutionary algorithm. The first one periodically calls an automatic solver, and the second one uses a memory method to guide successive resolutions. In order to choose the more adapted, we test them on different scenarios of continuous traffic. The memory approach can handle higher densities by maneuvering fewer aircraft and inducing lower delays. It is also more stable over time as early planned maneuvers are more likely to comply to effective maneuvers.

I. Introduction

Dynamic evolutionary optimization is an active research topic and has many applications in real world. Dynamic problems are problems that change over time. Dynamic Optimization Problems (DOP) are dynamic problems solved by an optimization algorithm. In real life, most DOP involve uncertainties where changes occur over time in the objective function, environmental parameters or constraints [1]. These different changes can have consequences in the objective function, on the optimium solution and its evolution.

Many dynamic optimization algorithms are based on the principles of Artificial Evolution. Evolutionnary Algorithms (EAs) belong to this class of algorithms because they are inspired by the theory of species evolution. However, an adapted Dynamic Evolutionnary Algorithm (DEA) have to be able to follow a changing optimal solution and needs to take advantage of information gathered in previous generations to speedup the optimization search, even if the environment changes quickly.

Air traffic conflict resolution in a control sector can be defined as a DOP. The problem evolves because aircraft enter or exit the sector, their positions evolve and their trajectory prediction constantly change with time, because of uncertainties on speed and maneuver executions. Air traffic controllers manage the traffic by taking decisions on trajectories at specific time while the situation is permanently evolving.

In the literature, DOP are either defined as a sequence of static problems linked by dynamic rules ([2], [3], [4]), or as a problem that has time-dependent parameters in its mathematical expression ([5],[6]). Generally in stationary optimization, the only goal of optimization algorithms is to find the global optimum as quickly as possible. However, in Evolutionary Dynamic Problem (EDP) research where the considered problems vary over time, the goal shifts from finding the global optimum to detecting changes and then tracking the evolution of optima (local optima or ideally the global optimum) over time. Additionally, in the case where the problem after the change somehow correlates with the problem before the change, an optimization algorithm should also learn as much as possible from its previous search experience to hopefully move forward search more efficiently.

A. Naive approches

A basic approach consists in restarting the EA when a change happens in the system environment. In the Rerandomization PSO (RPSO) described by Hu and Eberhard [7], the entire or a part of the swarm is randomly relocated in the search space when a change occurs. The principal risk of this approach is to lose information from previous research and, consequently, to slow down the convergence.

B. Implicit memory

Implicit memory approach calls on redundant representations in order to memorize some good solutions which may be reuse later (redundant representations are representations containing more information than necessary to define a phenotype). Whereas Goldberg and Smith introduced a triallelic scheme in 1987 [8], where an allele can take "0," "recessive 1," or "dominant 1" values, Ng and Wong proposed in 1995 a new diploid [9] scheme using a diploid scheme with four possible alleles ("0 recessive," "1 recessive," "0 dominant," and "1 dominant"). In their experiments, diploid scheme seem better than haploid or triallelic schemes.

C. Explicit memory

Explicit memory approach uses external memory to store past elements population that may be useful in future steps of the evolutionary process. Contrary to implicit memory, with explicit memory we know how and when the specific information stored will be reused in later generations. An explicit memory was introduced by Louis and Xu in 1996 [10] on an open shop scheduling or re-scheduling problem as an example. When a change occurs during the simulation, a part of elements from previous generations of the EA are reused, and the other chromosomes are randomly initialized. The main limit encountered is when the environment changes are too much significant. As an example, in their case, a deleted task had an important repercussion on new good schedules and it was much more efficient to build a new solution from scratch. A similar method employed a short-term memory in order to recall, when a change occurs, individual's ancestors that have been evaluated good in last generations [11]. In [12], authors add a local memory on each chromosome about their ancestors solutions. Their conclusion is linked to Louis and Xu's approach because it highlights the fact that the method efficiency is correlated with the nature of environment changes.

D. Multiple Population Approaches

Multiple population approaches use multiple sub-populations to track multiple peaks in the landscape. A popular design proposed in ([13]), called a shifting equilibrium genetic algorithm, deployed a base population to exploit the best optimum found so far and multiple colonies to explore the search space. A measure of diversity, distance from the main population, was included in assessments of colony fitness. Interesting results have been obtained by the self-organizing scouts proposed by Branker in 2000 [14]. This algorithm use a mother population to explore research space, whereas daughter populations follow local optima, over the objective function changes.

II. Original Evolutionary Algorithm

In this article we deal with traffic in the horizontal plane and only accept one maneuver per aircraft. In order to comply with air traffic controllers behavior, our model takes into account uncertainties in trajectory prediction, which increase over time. Many realistic uncertainty models have been presented in previous work [15], [16]. In this article we use a simplified version of uncertainty described in [17] that increases the size of the separation standard linearly with time.

A. Maneuvers

Time is discretized into steps of duration τ to describe maneuvers. τ is small enough to detect every conflict in the application (in the simulations, $\tau = 3\,\mathrm{s}$, [18] discusses the topic).

In the trajectory model chosen, maneuvers are heading changes of α degrees, starting at time t_0 and ending at time t_1. Heading changes α take values that are discretized by steps of 5 degrees in order to comply with air traffic controllers practice. α is relative to the current heading. Figure 1 summarizes a current Air Traffic Control maneuver which can easily be implemented by pilots and current FMS technologies (cf. [19]).

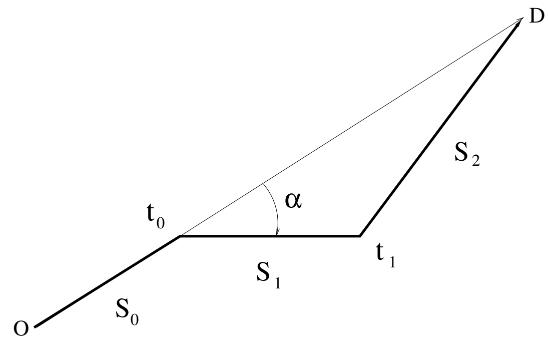

Fig. 1: Maneuver model.

Uncertainty is added to the model by increasing the separation standard linearly with time. Using such a growing norm is very convenient for quickly calculating the conflicting zone, but it can only model an isotropic growth of uncertainty. For example a 5% of the mean speed increase on the separation standard models a 5% speed uncertainty and a 3°heading uncertainty.

B. Data

The EA used is similar to those described in [20] and [21]. It finds optimal maneuvers t_0, t_1, α for each aircraft of a scenario with the uncertainty model.

An example of a population element (chromosome) is given in figure 2 for a situation including n aircraft. Each value is coded by a positive float.

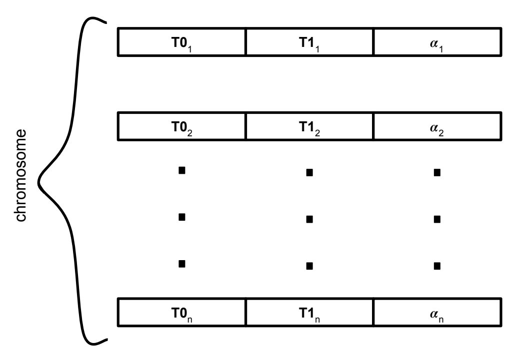

Fig. 2: Structure of a chromosome

The algorithm receives as input the current time t_c, the current flight plans of each aircraft, and their current maneuver (null if the aircraft is not maneuvered). If a maneuver has already started for an aircraft, its heading α and t_0 cannot be modified. However, if the end of maneuver t_1 is more than 60 seconds after t_c, t_1 can still vary and will be optimized.

A lag ($lag = 30$ seconds in the experiments) was introduced to prevent any maneuver modification while the solver is optimizing the solution. For entering aircraft, another lag ($lag_b = 120$ seconds in the experiments) was introduced. If t_b is the entering time of an aircraft, a new maneuver cannot begin before $t_c + lag$ or before $t_b + lag_b$. This lag simulates the necessary time to take a new aircraft into account. We also make sure that the position at time t_1 allows an acceptable maneuver, i.e. a maneuver which respects the maximum turning rate toward the exit point. required (the $3°$ per second standard rate was chosen).

The Evolutionary Algorithm chosen used is described in [20]. An intelligent crossover operator is used to recombine the most promising maneuvers of each parent solutions. An intelligent mutation operator is also used to correct problematic maneuvers in the chromosome chosen for mutation. 100 generations are executed, the population is set to 150 elements. At each generation, 40% of the population is crossed and 40% of the population is muted. A $\sigma = 2$ truncation scaling is used and a simple sharing process is adopted to prevent convergence to local minimum. It divides the fitness of solutions that share the same characteristics by their occurrence numbers: two solutions share the same characteristics if they turn right or left or keep straight the same aircraft.

In the initial population and for both approaches, when a new maneuver is randomly defined, we chose to give one time out of 10 a direct route to the aircraft in order to favor solutions with the least number of maneuvers and shortest delay in the algorithm.

C. Fitness

The fitness function takes into account the four following criteria ranked by their criticality:

1) eliminate remaining conflicts (n_{conf} the number of conflicts). Whenever a conflict remains, we try to delay it as late as possible because it could get a chance to be solved later. A parameter $v_{tmin} \in [0 : 1]$ measures the latest remaining conflict ;
2) minimize number of aircraft maneuvered (n_{man} the number of maneuvered aircraft);
3) minimize delays caused by maneuvers ($delay$ the sum of delays);
4) start maneuvers as late as possible (n_b the sum of the differences between the maximum possible value of maneuver start $t_{0_{max}}$ and the actual maneuvering start t_0) : because of uncertainties, it is worth waiting as late as possible before maneuvering an aircraft.

The EA first tries to eliminate every conflict, and then optimize trajectories resulting from maneuvers.

Finally, the resulting fitness of a chromosome is :

$$F = \begin{cases} \frac{1}{1+n_{conf}} * v_{tmin} & \text{if } n_{conf} > 0 \\ 1 + (n - n_{man}) + \frac{1}{1+2*delay+n_b} & \text{else.} \end{cases}$$

III. Approaches Description

In this section, we detail the two different approaches in order to continuously solve conflicts. The main difference between these approaches concerns the notion of memory defined in I-C.

A. Naive Approach (NA)

This approach is a naive version which launches the EA every $\Delta_s = 30$ seconds. Maneuvers calculated by the solver are applied if they begin in less than 60 seconds, otherwise they are not sent to the aircraft because they may be applied at the next step.

B. Explicit Memory Approach (EMA)

With the Explicit Memory Approach, the idea is to keep information found from previous optimization to build new solutions. Unlike the Naive Approach, the EMA tries to build new solutions that are closer to the old ones. Indeed, after every resolution performed by the solver, we keep in an external file the last generation population in order to initialize the new population of the solver ($\Delta_s = 30$ seconds later). Only solutions that still comply with the current environment constraints are kept.

There are three main cases :

- A new aircraft arrived in the sector : a new maneuver allele will be randomly initialized and added for each population element ;
- An aircraft left the sector : the corresponding maneuver allele will be removed from each population element ;
- An aircraft has been maneuvered since the last EA optimisation : population elements which suggested another maneuver for this aircraft (a different t_0 or a different α for example), will have this maneuver allele randomly initialized with a maneuver which respects the new constraints (same t_0 and same α).

IV. Experiments

In this section, the air traffic generation scenario is described and the criteria used to compare the two approaches are presented.

A. Exercise generation

In order to generate continuous random traffic scenarios with different conflicts configurations, we consider a circular sector of 50 nautical mile radius (about 15 minutes of flying time for an aircraft), with 14 possible entry points regularly positioned on its circumference.

For our tests, the number of aircraft in the traffic situations is regulated using a Poisson law with variable λ depending on the scenario. Each aircraft is randomly assigned:

- a nominal speed, randomly chosen between 370 and 550 knots;
- its entry point, in a rectangular area of 10 nautical miles around one of the sector's entry points;
- an exit point on the opposite side of the sector, in a slice extending by plus or minus 30 degrees around the opposite point on the circle.

Initially, each aircraft flies directly from its entry point to its exit point.

However, in order to build conflict scenarios and avoid unmanageable traffic situations, a minimal duration of 4 minutes

TABLE I: Scenarios

λ	Number of aircraft
1.1	97
1.2	102
1.3	103
1.4	105
1.6	113
1.9	123
2.0	128
2.5	136
3.0	151
3.5	161

is required before the first conflict of an aircraft entering the sector.

Finally, different density scenarios using increasing values of λ (see table I) are compared with both approaches. Each scenario last 3 hours and 20 minutes (12000 seconds).

B. Comparison parameters

1) The first objective, included in the EA fitness function, is to solve all conflicts. Because we only included one flight level and strong restrictions on maneuvers (only heading change per aircraft), some situations become difficult to solve when the density increases. We first compare the number of unsolved conflicts in both approaches.
2) Second, we compare the percentage of aircraft maneuvered for both approaches.
3) Third, we compare the percentage of delay due to maneuvers for both approaches.
4) Fourth, we compare the number of conflict pairs detected by both approaches. Because both approaches lead to different decisions and thus different trajectories, the number of conflict pairs detected at the end of each scenario is different.
5) Fifth, we compare the number of maneuver heading changes : this happens when the solution calculated 30 seconds before contains a maneuver in the opposite direction compared to the new one. These elements will impact controllers reactions and appreciation of the tool if such a tool is used to help them make decisions : the more stable the resolution over time, the more acceptable and the less stressful it is for air traffic controllers.

The last item (maneuver heading changes) deserves to be more explained. For an aircraft in particular, there is a heading change if it is in one of the following situations at time t, with $t + lag > t_0$ and $t - \Delta_s + lag < t_0$:

- at time $t - \Delta$, the aircraft had a heading change α to the right and at time t the aircraft has heading change to the left;
- at time $t - \Delta$, the aircraft had a heading change α to the left and at time t the aircraft has heading change to the right;
- at time $t - \Delta$, the aircraft had no heading change ($\alpha = 0$) and at time t the aircraft has a heading change ($\alpha \neq 0$);

V. Results

Results from simulations on each scenario are transcribed in table II.

A. Remaining conflicts

There is a big difference between the two approaches on the number of unsolved conflicts. For low densities, both approaches solve all the conflicts. Remaining conflicts for low densities are probably caused by the lack of coordination when aircraft enter the sector. Some significant number of unsolved conflicts appear when $\lambda \geq 1.9$ for the naive approach, whereas the memory version keeps a low number of unsolved conflicts for $\lambda \geq 3$.

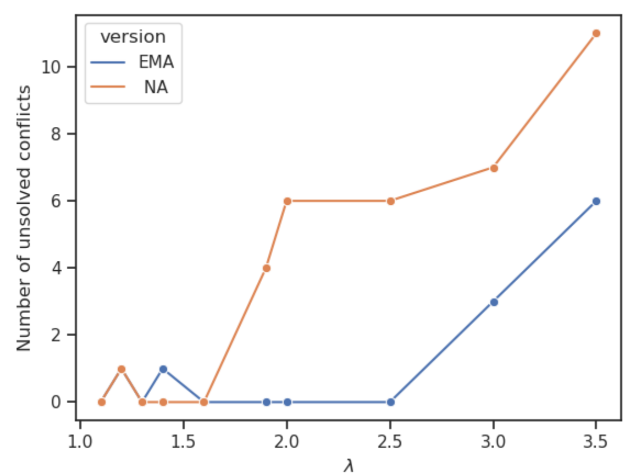

Fig. 3: Number of unsolved conflicts

B. Percentage of maneuvered aircraft

The number of maneuvered aircraft is for all experiments smaller with the Explicit Memory Approach than with the Naive Approach. For low traffic densities, there are more than 30% more maneuvers with the NA than with the EMA. This ratio tends to decrease when the density increases, which is due to the fact that in high densities, every aircraft tends to be maneuvered.

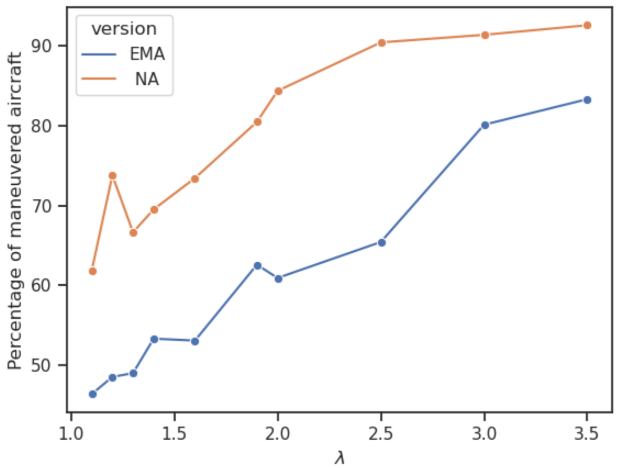

Fig. 4: Percentage of maneuvered aircraft

C. Delays

The percentage of delay per aircraft increases with the traffic density for both approaches. However there is a big difference between the two approaches : the EMA approach induces a smaller delay to aircraft. For $\lambda = 3.5$, the delay observed with the NA is 70% higher than the delay observed with the EMA.

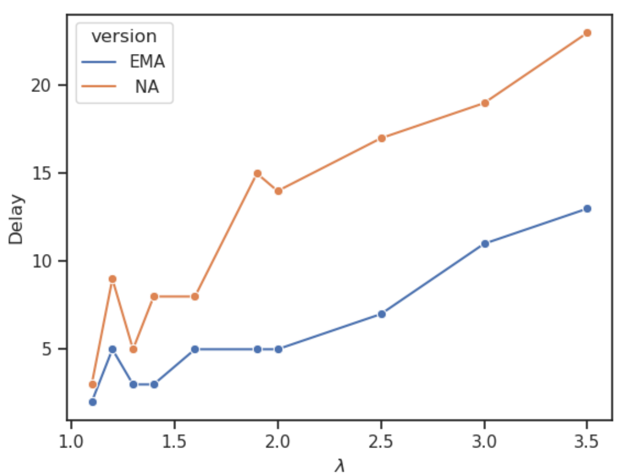

Fig. 5: Delay (in $\%$)

D. Conflict pairs

As expected, the number of conflict pairs detected increases with the density. However it seems that the EMA solutions reduce the number of conflict pairs.

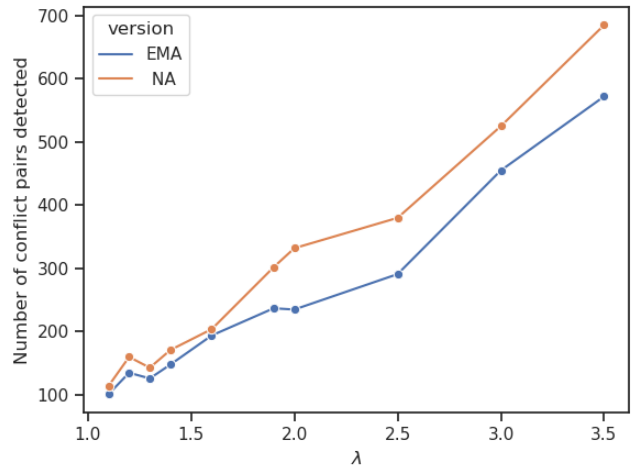

Fig. 6: Number of conflict pairs detected

E. Percentage of Heading changes

The percentage of heading changes varies a lot with the scenarios. However the EMA tends to keep it lower than the NA, which was the goal of the Memory Approach. When the density is very high, both percentages of heading changes converge, probably because situations change more often.

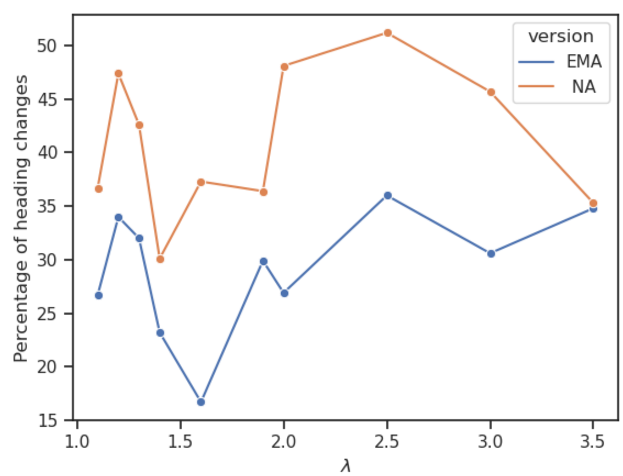

Fig. 7: Percentage of heading changes

F. Amount of useful memory

The main particularity of the Explicit Memory Approach is to keep as much information as possible from the last generation of the previous optimization and to reuse it in the current generation, we measured the proportion of alleles reused during each experiments, which we called the amount of useful memory.

The following figure traces the percentage of reused alleles for each λ scenarios.

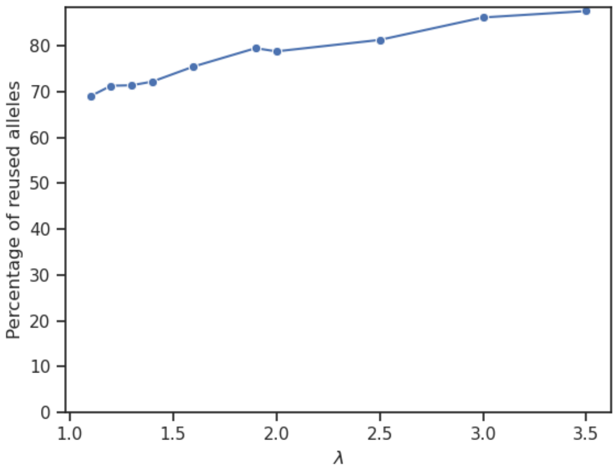

Fig. 8: Percentage of reused alleles

The percentage of reused alleles increases with traffic density. It can be explained by the fact that the higher the traffic density, the more maneuvering constraints there are. When aircraft started a maneuver they cannot change the maneuver start time or heading angle, this information is kept in the next generation. It would be interesting in further work to separately count these cases.

VI. Conclusion and Further Work

In this article, we compared two approaches for conflict resolution in the evolving environment of a sector. Results globally show that an explicit memory approach is better than a naive approach in terms of number of maneuvers, delays, unsolved conflicts, conflict pairs and heading changes.

Indeed, the memory-based approach allows to keep the number of unsolved conflicts smaller for higher densities. It reduces significantly the number of maneuvered aircraft, which is a very important factor for building manageable solutions. The delays per aircraft during the simulations are also significantly reduced which probably impacts the number of conflict pairs. The percentage of heading changes measures the number of times a solution is drastically changed over time. It measures the stability of the solutions planned in advance. This percentage is also much lower with a Memory approach. A maneuver predictor should be as stable as possible if it were to be used as an aid for air traffic control. We can notice that on the average, 70% to 90% percent (according to the traffic density) of previous alleles are used in the memory process.

In future work, we should add a vertical dimension to the problem, which would make it more realistic and allow to explore a bigger range of maneuvers and higher densities of traffic.

Another memory approach could be imagined by keeping alive a population of solutions and picking regularly in this evolving population the best solution to decide the maneuvers to assign to aircraft.

References

[1] Y. Ong and Y. Jin, *Evolutionary Computation in Dynamic and Uncertain Environments*, vol. 51. 03 2007.

[2] K. Weicker, "An analysis of dynamic severity and population size," *Parallel Problem Solving from Nature*, vol. VI, 2002.

[3] V. S. Aragon and S. C. Esquivel, "A evolutionary algorithm to track changes of optimum value locations in dynamic environments," *Journal of Computer Science and Technology*, vol. 4, no. 3, pp. 127–134, 2004.

[4] P. K. L. P. Rohlfshagen and X. Yao, "Dynamic evolutionary optimisation: An analysis offrequency and magnitude of change," in *Proceedings of the 2009 Genetic and Evolutionary Computation Conference GECCO'09*, pp. 1713–1720, 2009.

[5] P. A. N. Bosman, "Learning and anticipation in online dynamic optimization," in *Studies in Computational Intelligence* (E. C. in Dynamic and U. Environments, eds.), vol. 51, pp. 129–152, 2007.

[6] Y. G. Woldesenbet and G. G. Yen, "Dynamic evolutionary algorithm with variable relocation," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 500–513, 2009.

[7] X. Hu and R. Eberhart, "Adaptive particle swarm optimization: detection and response to dynamic systems," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 2, pp. 1666–1670 vol.2, 2002.

[8] D. E. Goldberg and R. E. Smith, "Nonstationary function optimization using genetic algorithms with dominance and diploidy," in *ICGA*, 1987.

[9] K. P. Ng and K. C. Wong, "A new diploid scheme and dominance change mechanism for non-stationary function optimization.," 1995.

[10] S. Louis and Z. Xu, "Genetic algorithms for open shop scheduling and re-scheduling," in *ISCA 11th Intl. Conf. on Computers and their Applications*, pp. 99–102, 1996.

[11] K. Trojanowski and Z. Michalewicz, "Searching for optima in non-stationary environments," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, pp. 1843–1850 Vol. 3, 1999.

[12] K. Trojanowski, Z. Michalewicz, and J. Xiao, "Adding memory to the evolutionary planner/navigator," pp. 483 – 487, 05 1997.

[13] F. Oppacher and M. Wineberg, "The shifting balance genetic algo-rithm: Improving the ga in a dynamic environment," in *Proc. Genetic Evol. Comput. Conf.*, pp. 504 – 510, 1999.

[14] J. Branke, T. Kaußler, C. Schmidt, and H. Schmeck, "A multi-population approach to dynamic optimization problems," 01 2000.

[15] C. Allignol, N. Barnier, N. Durand, and J.-M. Alliot, "A new framework for solving en-routes conflicts," in *10th USA/Europe Air Traffic Management Research and Developpment Seminar*, 2013.

[16] C. Allignol, N. Barnier, N. Durand, A. Gondran, and R. Wang, "Large Scale 3D En-Route Conflict Resolution," in *ATM Seminar, 12th USA/Europe Air Traffic Management R&D Seminar*, (Seattle, United States), June 2017.

[17] N. Durand, J.-B. Gotteland, and N. Matton, "Visualizing complexities: the human limits of air traffic control," *Cognition, Technology and Work*, Feb. 2018.

[18] N. Barnier and C. Allignol, "Trajectory deconfliction with constraint programming," *The Knowledge Engineering Review*, vol. 27, no. 03, pp. 291–307, 2012.

[19] G. Granger, N. Durand, and J. Alliot, "Optimal resolution of en-route conflicts," in *4th ATM R&D Seminar*, 2001.

[20] N. Durand, J.-M. Alliot, and J. Noailles, "Automatic aircraft conflict resolution using genetic algorithms," in *Proceedings of the Symposium on Applied Computing, Philadelphia*, ACM, 1996.

[21] N. Durand and J.-M. Alliot, "Optimal resolution of en route conflicts," in *First ATM Seminar Europe/USA, Saclay*, 1997.

TABLE II: Results

λ	Approach	Remaining conflicts	Percentage of maneuvered aircraft	Delay (%)	Conflict pairs detected	Percentage of Heading changes
1.1	EMA	0	46	2	101	26
	NA	0	62	3	114	37
1.2	EMA	1	48	5	135	34
	NA	1	73	9	160	47
1.3	EMA	0	49	3	126	32
	NA	0	66	5	143	43
1.4	EMA	1	53	3	148	23
	NA	0	69	8	171	30
1.6	EMA	0	53	5	194	16
	NA	0	73	8	204	37
1.9	EMA	0	63	5	237	30
	NA	4	80	15	302	36
2	EMA	0	60	5	235	27
	NA	6	84	14	332	48
2.5	EMA	0	65	7	291	36
	NA	6	90	17	380	51
3.0	EMA	3	80	11	455	31
	NA	7	91	19	525	46
3.5	EMA	6	83	13	572	35
	NA	11	92	23	685	35

Sarah Degaugue is a PhD student at the École Nationale de l'Aviation Civile (ENAC). She graduated from ENAC as an engineer with a double master's degree in Operational Research (OR) in 2021.

Nicolas Durand is a professor at the École Nationale de l'Aviation Civile (ENAC). He graduated from the École Polytechnique de Paris in 1990 and from ENAC in 1992. He has been a design engineer at the Centre d'Études de la Navigation Aérienne (then DSNA/DTI R&D) from 1992 to 2008, holds a PhD in Computer Science (1996) and got his HDR (French tenure) in 2004.

Jean-Baptiste Gotteland is an assistant professor at the École Nationale de l'Aviation Civile (ENAC). He graduated from ENAC as an engineer in 1995, and received a PhD (2004) in computer science from the University of Toulouse.