A New Neighborhood for the QAP

Elizabeth F. G. Goldbarg[1]

Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil

Nelson Maculan[2]

COPPE
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil

Marco C. Goldbarg[3]

Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil

Abstract

Local search procedures are popular methods to solve combinatorial problems and neighborhood structures are the main part of those algorithms. This paper presents a new neighborhood for the Quadratic Assignment Problem. The proposed neighborhood is compared with the classical 2-exchange neighborhood.

Keywords: Local search, quadratic assignment, 3-assignment neighborhood.

[1] Email: `beth@dimap.ufrn.br`
[2] Email: `nmaculan@educacao.rj.gov.br`
[3] Email: `gold@dimap.ufrn.br`

1 Introduction

The idea behind local search is very simple. Each iteration starts with an initial solution ρ, then a number of neighbor solutions are obtained with the application of a specific operator to ρ. If a neighbor solution ρ' is better than ρ,the process is re-started with ρ' replacing ρ. Versions of the basic algorithm are implemented with, for example, ρ' being the best solution in the neighborhood of ρ or with ρ' being the first neighbor better than ρ.

In general, the neighborhood structure depends on the problem under consideration and is a determinant factor for the performance of the local search algorithm regarding both quality of solution and processing time. This paper investigates a new neighborhood structure for the Quadratic Assignment Problem, QAP [5]. Given two $n \times n$ matrices $F = (f_{kl})$ and $D = (d_{ij})$, the QAP can be stated as follows:

$$min_{\rho \epsilon \Pi_n} \sum_{i,j} f_{\rho(i)\rho(j)} d_{ij} \tag{1}$$

where Π_n is the set of all permutations ρ of $\{1, 2, ..., n\}$. One of the major applications of the QAP is in location theory where matrix F is a flow matrix, i.e. f_{kl} is the flow of materials between facilities k and l, and D is a distance matrix, i.e., d_{ij} represents the distance from location i to location j. The objective is to find an assignment of all facilities to all locations (a permutation $\rho \; \epsilon \; \Pi_n$), such that the total cost of the assignment is minimized. In terms of Graph Theory, the QAP can be thought of as an assignment of vertices between two complete graphs of order n, G'_D and G'_F corresponding to matrices D and F, respectively.

Once the QAP is an NP-hard problem [7] and models a number of real world applications, several heuristics have been proposed for handling near optimum solutions [6]. Pairwise exchanging operations are basic operations for most successfull approaches presented to solve the QAP. The neighborhood built by pairwise exchanging operations is known as the 2-exchange neighborhood. Given a solution ρ, the 2-exchange neighborhood $\aleph(\rho)$ is defined by the set of permutations which can be obtained by exchanging two elements of ρ, that is $\aleph(\rho) = \{\rho' \mid \rho'[r] = \rho[s],\ \rho'[s] = \rho[r], \text{and } \rho'[i] = \rho[i],\ \forall i,\ i \neq r, s\}$. There are $n(n-1)/2$ possible combinations of two locations on a permutation of size n and there is only one way of exchanging the facilities of those locations. The objective function difference of exchanging the facilities of two locations can be computed in $O(n)$ [10].

A natural extension of the 2-exchange neighborhood is the k-exchange neighborhood, where the facilities of k locations are exchanged. A version of

k-exchange neighborhoods is proposed by Ahuja *et al.* [1]. In their paper, they develop a very large-scale neighborhood structure for the QAP and, using the concept of improvement graph, they enumerate multi-exchange neighbors of a given solution. They compare their proposal with the 2-exchange neighborhood on benchmark QAP instances.

The 3-assignment neighborhood is introduced in the next section. A computational experiment compares the proposed neighborhood with the 2-exchange neighborhood by means of two versions of an iterated local search algorithm.The computational experiment and some conclusions are presented in section 3.

2 The 3-assignment Neighborhood

This work proposes a neighborhood that utilizes the fact that an edge assignment between two complete graphs of order 3 corresponds to a vertex assignment on those graphs. The proposed neighborhood, called 3-*assignment*, is based upon a naive lower bound for the QAP. Ranking the elements of F and D in non-increasing and non-decreasing orders, respectively, one obtains vectors F^- and D^+, respectively. The scalar product of F^- and D^+ is a weak lower bound for a QAP instance. It corresponds to an assignment of edges. If that assignment corresponds to a vertex assignment, then it is an optimal solution for the correspondent QAP instance. Though the lowest cost edge assignment for a QAP instance is very easy to solve, usually that assignment does not correspond to an assignment of vertices for instances with $n > 3$. If it was the case, the QAP would be solved in polynomial time. The 3-assignment neighborhood of a solution ρ is, then, composed with solutions ρ' such that, given three facilities q, r, s, locations ρ'$[q]$, ρ'$[r]$ and ρ'$[s]$ correspond to the lowest cost assignment of the edges of the 3-clique formed by $\rho[q]$, $\rho[r]$ and $\rho[s]$ to the edges of the 3-clique formed by q, r, s, and $\forall i, i \neq q, r, s,\ \rho'[i] = \rho[i]$.

The idea of combining optimization models and heuristic methods was introduced in the context of Tabu Search [4] with the name of *Referent-Domain Optimization*. The goal of referent-domain optimization is to introduce *"one or more optimization techniques to strategically restructure the problem or neighborhood, accompanied by auxiliary heuristic or algorithmic process to map the solutions back to the original problem space."* In this context, the idea behind the 3-assignment neighborhood is fixing a restricted number of variables and solving a relaxation of the correspondent subproblem. Clearly, the proposed neighborhood can be extended to k-assignment neighborhoods.

3 Computational Experiments and Conclusion

The QAP instances utilized in this work belong to two classes. The first class is composed with the Tai*xx*a instances [9]which are randomly generated from a uniform distribution. The second class was proposed by Drezner *et al.* [3] (http://business.fullerton.edu/zdrezner). Those instances are symmetric, easy to solve by exact methods, and difficult for local search heuristics.

In order to evaluate the potential of the proposed neighborhood regarding quality of solution, iterated local search algorithms were implemented with basis on the 2-exchange, *Ils-2e*, and on the 3-assign, *Ils-3a*, neighborhoods. Iterated local search is a stochastic method in which a sequence of local search procedures is applied to solutions iteratively. Rather than sampling the search space as multi-start heuristics do, it repeatedly applies local search on the local optimum resultant from the previous iteration after making a perturbation on that solution. Versions of an iterated local search algorithm for the QAP utilizing the 2-exchange neighborhood were presented by Stützle [8].

The general framework of the iterated local search algorithm utilized in this work is presented in Algorithm 1. The algorithm makes use of a memory implemented as a list initialized with m local optima resultant from random starting points, named *history*. The best solution of *history* is chosen as the starting point of the algorithm. While a given number of iterations is not reached, the algorithm repeats the following steps. The solution is disturbed by changing the facilities of k locations. The correspondent local search is applied to the disturbed solution. Finally, an acceptance criterion defines which solution will be the starting point of the next iteration. The procedure *acceptance_criterion*() has three input parameters: the original solution, ρ, the disturbed solution, ρ', and *history*. If ρ' has a better objective function value than ρ, then the former solution is accepted as the starting point of the next generation and *history* is updated with the new solution. Otherwise, a new solution of *history* is randomly chosen to be the next starting solution.

A preliminary experiment tested the following values for k: $10\%n$, $20\%n$, $30\%n$ and $40\%n$. The best results were obtained with $k = 10\%n$ for instances Tai*xx*a and with $k = 30\%n$ for instances Dre*xx*. One hundred independent runs of each algorithm were executed for each instance on a Pentium IV, 1.8 Ghz, under Ubuntu Linux. Table 1 shows the results of the computational experiment. The columns show the name of the instance, the best known solution, the minimum, the average, the standard deviation and the processing time in seconds of each version of the iterated local search algorithm.

Concerning Tai*xx*a instances, except for Tai40a, *Ils-3a* found better mini-

Algorithm 1 *Iterated Local Search*

```
begin
 Initialize(history)
 $\rho$ = best_solution(history)
 repeat
   $\rho$ = disturb($\rho$)
   $\rho$' = local_search($\rho$)
   $\rho$ = acceptance_criterion($\rho$,$\rho$',history)
 until stop criterion is satisfied
end
```

Instance	BKS	*Ils-3a*				*Ils-2e*			
		Min	Av	SD	T(s)	Min	Av	SD	T(s)
Tai20a	703482	0	1.35	5.00	0.20	0.30	1.42	6.20	0.04
Tai30a	1818146	0.02	1.62	4.78	1.03	0.91	1.91	4.19	0.14
Tai40a	3139370	1.05	1.90	3.37	3.66	0.88	2.27	4.23	0.46
Tai50a	4941410	1.53	2.18	3.17	9.64	1.80	2.70	2.59	1.01
Tai60a	7205962	1.53	2.29	2.51	21.92	2.14	2.77	2.35	2.04
Tai80a	13546960	1.18	1.86	1.77	74.49	1.79	2.34	1.74	5.63
Tai100a	21123042	1.14	1.68	1.49	217.82	1.85	2.17	1.23	12.83
Dre42	764	48.17	63.95	54.10	3.99	50.26	68.93	56.00	0.47
Dre56	1086	63.17	80.71	46.75	12.98	76.61	87.31	44.96	1.24
Dre72	1452	73.00	86.73	44.06	41.41	83.06	95.09	38.51	3.20
Dre90	1838	86.62	96.63	36.09	101.17	95.43	105.75	37.27	7.15

Table 1

Results of the computational experiment

mum results than the *Ils-2e*. The minimum results obtained with *Ils-3a* are, in average, 49.92% better than the minimum results obtained with *Ils-2e*. All the average solutions found by *Ils-3a* are better than the ones found by the *Ils-2e* with improvements ranging from 5.2 to 29.17%. In average, the *Ils-3a* presents improvements of 20% over the average values found by the *Ils-2e*. Similar results are observed for Dre*xx* instances. The results presented for *Ils-3a* are, in average, 15.51% and 8.76% regarding the best and the average solutions, respectively.

The statistical test of Mann-Whitney (U-test) is utilized to verify the significance of the experimental results. That test, also called Mann-Whitney-Wilcoxon test or Wilcoxon rank-sum test is a non-parametric test used to verify the null hypothesis that two samples come from the same population [2]. The results of the U-test with a level of significance of 0.01, rejected the null hypothesis for all instances.

Processing times of the algorithmic version that implemented the proposed neighborhood are higher than the other algorithm. Nevertheless, no special data structures were utilized on those implementations. Specific data structures designed for the proposed neighborhood may lead to lower runtimes.

References

[1] Ahuja, R., Jha, K., Orlin, J., and D. Sharma, "Very Large-scale Neighborhood Search for the Quadratic Assignment Problem“, Working Paper, Operations Research Center, MIT, Cambridge, MA, 2002.

[2] Conover,W.J., "Practical Nonparametric Statistics", John Wiley & Sons, 2001.

[3] Drezner Z., Hahn, P.M., and E.D. Taillard, *Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for metaheuristic methods*, Annals of Operations Research **139**(2005), 65–94.

[4] Glover F., *Tabu search and adaptive memory programming - advances, applications and challenges*, in Interfaces in Computer Science and Operations Research, Barr, Helgason and Kennington (eds.), Kluwer Academic Publishers, 1996, 1–75.

[5] Koopmans, T.C., and M.J. Beckmann, *Assignment problems and the location of economic activities*, Econometrica **25** (1957), 53–76.

[6] Loiola, E.M., Abreu, N.M.M., Boaventura-Netto, P.O., Hahn, P., and T. Querido, *A survey for the quadratic assignment problem*, European Journal of Operational Research **176**(2) (2007), 657–690.

[7] Sahni, S., and T. Gonzalez, *P-complete approximation problems*, Journal of the Association of Computing Machinery **23** (1976), 555–565.

[8] Stützle, T., *Iterated local search for the quadratic assignment problem*, European Journal of Operational Research **174** (2006), 1519-1539.

[9] Taillard, E.D., *Robust tabu search for the quadratic assignment problem*, Parallel Computing **17** (1991),443–455.

[10] Taillard, E.D., *Comparison of iterative searches for the quadratic assignment problem*, Location Science **3** (1995), 87–105.

Some Inverse Traveling Salesman Problems

Yerim Chung [1]

Paris School of Economics, Paris 1 University, France

Marc Demange [2]

Département SID, ESSEC, Cergy Pontoise, France

Abstract

Usual inverse combinatorial optimization problems consist in modifying as little as possible the instance parameters to make a given solution optimal. In this paper we consider several extensions taking into account constraints on the weight system and inverse problems against a specific algorithm. We consider TSP under this point of view and devise both complexity and approximation results.

Keywords: Inverse combinatorial optimization, TSP, $TSP_{\{1,2\}}$, $2opt$

1 Introduction

Inverse combinatorial optimization problems have been extensively studied for weighted problems during the last decade [1,5]. Given an instance of a weighted combinatorial optimization problem P and a fixed feasible solution, the corresponding inverse problem, denoted by IP, consists in modifying as little as possible (with respect to a fixed norm) the weight system in such a way

[1] Email: `Yerim.Chung@malix.univ-paris1.fr`

[2] Email: `demange@essec.fr`

the given solution becomes an optimal solution of P in the new instance. We only consider the L_1-norm. In [1,5], some results on the relative complexity of a combinatorial optimization problem and its inverse version are devised. In particular, by using linear programming it is shown that for a general class of polynomial problems, the related inverse problem is also polynomial. It is given in [9] an example of a polynomial problem admitting a NP-hard inverse version. On the other hand, it is straightforward to verify that if the optimality test which is to decide if a given solution is optimal, is *co*-NP-complete, then the corresponding inverse problem is NP-hard. Indeed, the optimality test is equivalent to decide whether the related inverse instance is of optimal value 0. In [3], we consider some modifications of IP, denoted by $IP_{W,A}$, where W denotes some properties that the weight have to satisfy and A denotes a given algorithm (either an optimal or an approximation algorithm). Given an instance of P and a fixed feasible solution x_0, $IP_{W,A}$ is to find a minimal modification of the weight system such that (i) the new system satisfies W and (ii) the fixed solution x_0 will be returned by A in the new instance. In [3], we have proposed an application model and also complexity and approximation results for such issues in the case of the maximum stable set problem. Here we consider the case of the minimum Traveling Salesman Problem, denoted by TSP. Our aim is essentially theoretical although some models may involve it.

We investigate the problems $ITSP_{W,A}$ where $W \in \{\mathbb{R}, \Delta, \{1,2\}\}$ and $A \in \{\emptyset, CN, 2opt\}$. $W = \mathbb{R}, \Delta, \{1,2\}$ respectively corresponds to the unconstrained case, the metric case and the case where weights are either 1 or 2. In this last case, changes on the instance correspond either to replace 1 by 2 or 2 by 1; so, any solution can be seen as the set of edges to be changed, the objective value being the number of these edges. $A = \emptyset$ corresponds to inverse TSP against every optimal algorithm. $2opt$ is the usual local search algorithm finding a hamiltonian cycle without 2-improvement (obtained by replacing two edges), and CN is a greedy algorithm repeatedly selecting the closest non visited neighbor. We denote by w the original weight system ($w(e)$ is the weight of an edge $e \in E$) and by w' the new weight system.

2 Complexity and approximation results

Against any optimal algorithm ($A = \emptyset$), one can show that $ITSP_{\mathbb{R}}$, $ITSP_{\Delta}$ and $ITSP_{\{1,2\}}$ are NP-hard by using a polynomial time reduction to the problem of deciding if, given a graph and a hamiltonian cycle, there exists a second hamiltonian cycle [8]. On the other hand, for $W \in \{\mathbb{R}, \Delta\}$ and $A \in \{2opt, CN\}$, we have:

Proposition 2.1 *$ITSP_{\mathbb{R},CN}$, $ITSP_{\mathbb{R},2opt}$, $ITSP_{\Delta,CN}$ and $ITSP_{\Delta,2opt}$ can be solved in polynomial time.*

Proof (sketch)
The problem is to find w' minimizing the quantity $\|w - w'\|_1$ under some constraints. For the case of algorithm CN, constraints are of the form: $w'(e_{i,i+1}) \leqslant w'(e_{i,j})$, $\forall i \in \{1, \cdots, n-2\}, \forall j > i+1$. For algorithm $2opt$, the 2-optimality in the new instance can be expressed by linear constraints: $w'(e_{i,i+1}) + w'(e_{j,j+1}) \leqslant w'(e_{i,j}) + w'(e_{i+1,j+1})$, $\forall i < i+1 < j < j+1$. Finally, triangle inequalities can also easily be represented by linear constraints. To linearize the objective, we replace it by $\sum_{e \in E} z_e$, where $z_e, e \in E$ are new variables and we add for every edge e the constraints $w(e) - w'(e) \leqslant z_e$ and $w'(e) - w(e) \leqslant z_e$. We get a linear program with a polynomial number of constraints.

Let us now focus on $ITSP_{\{1,2\},2opt}$. $TSP_{\{1,2\}}$ is a particular case of metric TSP that has been widely studied and $2opt$ is a popular algorithm for this version of TSP, known to guarantee the approximation ratio of 3/2.

Theorem 2.1

(i) *$ITSP_{\{1,2\},2opt}$ is APX-hard, even if the graph induced by edges of weight 1 has the maximum vertex degree 4 in the original instance. Moreover, if $P \neq NP$ then it cannot be approximated within 1,36.*

(ii) *$ITSP_{\{1,2\},2opt}$ can be approximated within the ratio $(2 - \frac{1}{\delta})\rho_{VC}$, where δ is the maximum number of 2-opt swaps to which an edge may belong, and ρ_{VC} is any known approximation ratio for vertex covering. By using [6], we have $\rho_{VC} = 2 - \Theta(\frac{1}{\sqrt{logn-1}})$. So $ITSP_{\{1,2\},2opt}$ is APX-complete.*

Proof (sketch)
1. We devise a reduction preserving approximation from the vertex cover problem, VC. Let $G = (V, E)$ be an instance of VC where $V = \{x_1, \cdots, x_n\}$. We construct from G an instance $(H = (V', E'), w, HC^*)$ of $ITSP_{\{1,2\},2opt}$ as follows: $V' = \{x_1, \cdots, x_n\} \cup \{y_1, \cdots, y_n\}$, $E' = \{x_i x_j | \ \forall i \neq j\} \cup \{x_i y_j | \ \forall i, j\} \cup \{y_i y_j | \ \forall j \neq i\}$ and $w(e)$ is either 1 if $e = x_i x_j \in E$ *or* $e = y_i x_{i+1} \forall i$ or 2 otherwise. Let $HC_0 = \{x_1 y_1 x_2 y_2 x_3 y_3 \cdots x_n y_n x_1\}$ be the fixed solution.

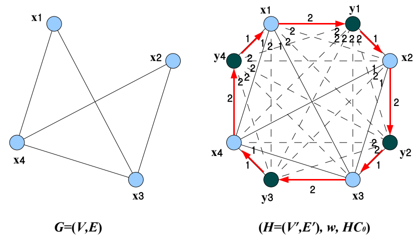


Figure 1. Construction of $(H = (V', E'), w, HC_0)$ from $G = (V, E)$

We show that we can restrict ourself to the case where the modification of the weights affects only the edges $x_i y_i, i = 1, \ldots n$ and that changing the weight of the edges $\{x_i y_i, i \in I_0 \subset \{1 \cdots, n\}\}$ makes HC_0 2-optimal if and only if the set $\{x_i, i \in I_0\}$ is a vertex covering of G. Note finally that this reduction is an L-reduction and recall that VC restricted to the graphs of maximum vertex degree 3 is APX-complete [8]. In H, the degree of the graph induced by the edges of weight 1 is equal to 4 if and only if the degree of G is 3.

2. Conversely, let $(G = (V, E), w, HC_0)$ be an instance of $ITSP_{\{1,2\},2opt}$, where G is a complete graph of order n, $\forall e \in E, w(e) \in \{1, 2\}$ and HC_0 is the fixed Hamiltonian cycle. HC_0 admits three types of 2-opt swaps as follows:

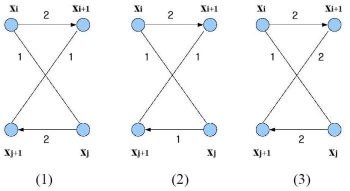


Figure 2. 2-opt swaps reducing the length of HC_0

Let sw_{ij} be the 2-opt swap containing $x_i x_{i+1}$ and $x_j x_{j+1}$ and SW_k be the set of swaps of type $k \in \{1, 2, 3\}$ in Figure 2. Let β_c ($\beta_{\overline{c}}$) be the best value obtained by changing only the weights on (outside) the cycle, respectively.

If one only changes the edges of the fixed cycle, then every edge of length 2 in swaps of $SW_1 \cup SW_2$ belongs to any feasible solution and without loss of generality, we can assume that the instance contains only the third type of swaps. Let $HC_0(2)$ be the set of edges of weight 2 in HC_0. We construct an instance $H = (V', E')$ of the minimum vertex cover problem, VC as follows: $V' = HC_0(2)$ and $\forall u, v \in V'$, $uv \in E'$ iff $\exists sw \in SW_3$ s.t. $u \in sw$ and $v \in sw$. To make HC_0 2-optimal, we need to select for each 2-opt swap sw_{ij} one edge between x_ix_{i+1} and x_jx_{j+1}, which is equivalent to find a vertex cover in H. So, we can use any approximation result on the problem VC for $ITSP_{\{1,2\},2opt}$ to construct a solution of value $\rho_{VC}\beta_c$. Let us now assume that we change only the edge weights outside the cycle; then it is sufficient to consider only swaps in SW_2 since for the other swaps the edges to change are obtained by necessary conditions. Note that in this case the transformation to VC still works. We construct an instance $H = (V'E')$ of VC as follows: a vertex of H corresponds to an edge outside the cycle of weight 1 and two vertices of H are connected if and only if their corresponding edges in G belong to a same 2-opt swap. Since the maximum vertex degree of H is equal to 2, a minimum vertex cover of H of size $\beta_{\overline{c}}$ can be computed in polynomial time. So we get a solution of value $\lambda \leqslant min\{\rho_{VC}\beta_c, \beta_{\overline{c}}\}$.

Let $\beta = \beta_1 + \beta_2$ be an optimal solution of $ITSP_{\{1,2\},2opt}$, where β_1 and β_2 are the numbers of changes occurred outside the cycle (changes from 1 to 2) and on the cycle (from 2 to 1), respectively. Since an edge outside the cycle intervenes in at most two 2-opt swaps, there is a solution of value $2\beta_1 + \beta_2$ containing only changes on the cycle. Consequently, $\beta_c \leqslant 2\beta_1 + \beta_2$. On the other hand, let δ be the maximum number of 2-opt swaps in which an edge of the cycle can intervene. By replacing a modification on the cycle by δ modifications outside the cycle we can obtain a solution of value at most $\beta_1 + \delta\beta_2$ containing only edges outside the cycle, so $\beta_{\overline{c}} \leqslant \beta_1 + \delta\beta_2$. We deduce $\lambda \leqslant min\{\rho_{VC}(2\beta_1 + \beta_2), \beta_1 + \delta\beta_2\}$. By using $\beta = \beta_1 + \beta_2$ and $\rho_{VC} \geqslant 1$, we obtain the ratio of $(2 - \frac{1}{\delta})\rho_{VC}$.

3 Discussion on some related problems

Whenever one considers TSP in a given metric space $\mathcal{E}$, we can consider a very natural inverse problem: given n requests (cities) in $\mathcal{E}$, a fixed order for these requests and a specific algorithm A solving metric TSP, it is to reposition the n requests (vertices) in $\mathcal{E}$, so that A chooses them respecting the given order. The aim is to minimize the total length of movings.

Consider for instance the line with the metric defined by absolute values. A simple optimal TSP algorithm is to select requests going from left to right and come back. The related inverse problem is equivalent to the an inverse sorting problem, also called *isotone optimization*, shown to be polynomially solved in [7] even with integral constraints (W=$\mathbb{N}$). It can be shown to be equivalent to the following problem: given a permutation $\pi = [\pi_1, \cdots, \pi_n]$ on n consecutive numbers, modify as little as possible the values in π (with respect to L_1 norm) such that the new sequence is not decreasing (i.e. the related permutation graph is a stable set (1-colorable graph)). Any optimal algorithm for TSP on the line selects some vertices going from left to right and the others during coming back. Considering now any such algorithm, the related inverse problem on the line is closely related to modifying as few as possible the numbers of the given sequence such that it can be divided into an increasing subsequence I and a decreasing subsequence D such that $max_{x \in I}\ x \leqslant min_{x \in D}\ x$ (i.e. the transformed permutation induces a threshold graph [4] that is a particular case of $(1,1)$-colorable graph where a graph is said to be (p,k)-colorable if it can be divided into p cliques and k stable sets).

In a further work [2], we consider the so called inverse (p,k)-colorability problem in permutation and interval graphs with nice applications. It aims to modify as few as possible the instance such that the resulting graph is (p,k)-colorable. We show that for any pair of constants (p,k), the inverse (p,k)-colorability problem in permutation graphs can be solved with complexity $O(n^{2(p+k)})$ by dynamic programming. It holds in particular for the above mentioned inverse TSP problems on the line.

References

[1] Ahuja, R. and J. Orlin, *Inverse optimization*, Operations Research **49** (2001), pp. 771–783.

[2] Chung, Y., J. Culus and M. Demange, *On some inverse chromatic number problems*, manuscript.

[3] Chung, Y. and M. Demange, *The 0-1 inverse maximum stable set problem*, to be published (2007).

[4] Golumbic, M., “Algorithmic Graph Theory and Perfect Graphs,” Academic press, New-york, 1980.

[5] Heuberger, C., *Inverse combinatorial optimization: A survey on problems, methods, and results*, J. Comb. Optim. **8** (2004), pp. 329–361.

[6] Karakostas, G., *A better approximation ratio for the vertex cover problem*, Electronic Colloquium on Computational Complexity (ECCC) (2004).

[7] Liu, M.-H. and V. A. Ubhaya, *Integer isotone optimization*, SIAM J. OPTIM **7** (1997), pp. 1152–1159.

[8] Papadimitriou, C., “Computational complexity,” Addison-whesley, 1994.

[9] Zhang, J., X. Yang and M.-C. Cai, *The complexity analysis of the inverse center location problem*, J. of Global optimization **15** (1999), pp. 213–218.

Limited Packings in Graphs

Robert Gallant [a,1,2] Georg Gunther [a,3] Bert Hartnell [b,4]
Douglas Rall [c,5]

[a] *Department of Mathematics, Sir Wilfred Grenfell College*
Memorial University of Newfoundland, Corner Brook, Canada

[b] *Department of Mathematics and Computing Science*
Saint Mary's University, Halifax, Canada

[c] *Department of Mathematics*
Furman University, Greenville, U.S.A.

Abstract

We define a k-limited packing in a graph, which generalizes a packing in a graph, and give several bounds on the size of a k-limited packing. One such bound involves the domination number of the graph, and here we show, when k=2, that all trees attaining the bound can be built via a simple sequence of operations. We also consider graphs where every maximal 2-limited packing is a maximum 2-limited packing, and characterize those of girth 14 or more.

Keywords: graph, domination, packing

[1] The first and third authors acknowledge the support of NSERC

[2] Email: `rpgallant@swgc.mun.ca`

[3] Email: `ggunther@swgc.mun.ca`

[4] Email: `Bert.Hartnell@smu.ca`

[5] Email: `Doug.Rall@furman.edu`

1 Introduction

Consider the following scenarios:

Network Security: A set of sensors are to be deployed to covertly monitor a facility. Too many sensors close to any given location in the facility can be detected. Where should the sensors be placed so that the total number of sensors deployed is maximized?

NIMBY: A city requires a large number of obnoxious facilities (such as garbage dumps), but no neighborhood should be close to too many such facilities, nor should the facilities themselves be too close together. Where should the facilities be located?

Market Saturation: A fast food franchise is moving into a new city. Market analysis shows that each outlet draws customers from both its immediate city block and from nearby city blocks. However it is also known that a given city block cannot support too many outlets nearby. Where should be outlets be placed?

A graph model of these scenarios might maximize the size of a vertex subset subject to the constraint that no vertex in the graph is near too many of the selected vertices. The well-known packing number of a graph is the maximum size of a set of vertices B such that for any vertex v the closed neighborhood of v, $N[v]$, satisfies $|N[v] \cap B| \leq 1$. In this paper we consider relaxing the constraint to $|N[v] \cap B| \leq k$, for some fixed integer k.

Our notation is standard. Specifically, given a graph G then $V(G)$ is the set of vertices of G, $\gamma(G)$ is the domination number of G, $\rho(G)$ the packing number, $\delta(G)$ is the minimum degree of a vertex in G, $\Delta(G)$ is the maximum degree of a vertex in G, and for a vertex $v \in V(G)$, $N[v]$ is the closed neighborhood of v, which is the set of vertices adjacent to v along with v itself. The girth of a graph is the length of the shortest cycle in the graph, which is said to be infinite if the graph is a forest. The symbol P_t denotes the path with t vertices, and if a vertex v in a tree is adjacent to a stem of degree 2, we will say v has a P_2 attached.

Definition 1.1 Let G be a graph, and let $k \in \mathbb{N}$. A set of vertices $B \subseteq V(G)$ is called a k-limited packing in G provided that for all $v \in V(G)$, we have $|N[v] \cap B| \leq k$.

In [1], the author introduces a notation unifying the description of many graph theoretic parameters. Specifically, in the context of a given graph G, a set $B \subseteq V(G)$ is called a $[\rho_{\leq k}, \sigma_{\leq k-1}]$-set provided any vertex v in G has $|N[v] \cap B| \leq k$, which is what we are calling a k-limited packing. Similarly a

2-limited packing in a graph would be called a $[\rho_{\leq 2}, \sigma_{\leq 1}]$-set.

A k-limited packing B in a graph G is called *maximal* if there does not exist a k-limited packing B' in G such that $B \subsetneq B'$. A k-limited packing B in a graph G is called *maximum* if there does not exist a k-limited packing B' in G such that $|B| < |B'|$.

We are interested in the maximum size of a k-limited packing in an arbitrary graph.

Definition 1.2 Let G be a graph, and let $k \in \mathbb{N}$. The k-limited packing number of G, denoted $\mathrm{L}_k(G)$, is defined by

$$\mathrm{L}_k(G) = \max\{|B|| \ B \text{ is a } k\text{-limited packing in } G \ \}.$$

If a subset of vertices B is a packing then the distance between any pair of distinct vertices in B is at least 3, in which case $|N[v] \cap B| \leq 1$ for any vertex v in the graph, so B is also a 1-limited packing. But a 1-limited packing B has $|N[v] \cap B| \leq 1$ for any vertex v, and so the distance between any pair of distinct vertices in B is at least 3. Thus 1-limited packings and packings are the same, and so $\mathrm{L}_1(G) = \rho(G)$.

Since a k-limited packing is also a $(k+1)$-limited packing we immediately obtain the following inequalities:

$$\rho(G) = \mathrm{L}_1(G) \leq \mathrm{L}_2(G) \leq \ldots \leq \mathrm{L}_{\Delta(G)+1}(G) = |V(G)|.$$

We collect some easily verified facts about the k-limited packing numbers of some familiar graphs in the following lemma.

Lemma 1.3 *Let $m, k, n \in \mathbb{N}$ with $m \geq 3$. Let P_m be the path on m vertices, let C_m be the cycle on m vertices, and let K_m be the clique on m vertices. Then:*

- $\mathrm{L}_1(P_m) = \lceil \frac{m}{3} \rceil$,
- $\mathrm{L}_2(P_m) = \begin{cases} \frac{2m}{3} & \text{if } m \equiv 0 \bmod 3, \\ \lfloor \frac{2m}{3} \rfloor + 1 & \text{otherwise} \end{cases}$
- $\mathrm{L}_1(C_m) = \lfloor \frac{m}{3} \rfloor$
- $\mathrm{L}_2(C_m) = \lfloor \frac{2m}{3} \rfloor$
- $\mathrm{L}_k(K_m) = \min\{k, m\}$
- $\mathrm{L}_k(K_{m,n}) = \begin{cases} 1 & \text{if } k = 1, \\ \min\{k-1, m\} + \min\{k-1, n\} & \text{if } k > 1. \end{cases}$

2 Bounds on k-limited packings

In this section we bound the k-limited packing number of a graph G. First we observe some connections to domination numbers of G.

For a positive integer $k \leq \delta(G) + 1$, a subset D of $V(G)$ is called a k-tuple dominating set in G if $|N[v] \cap D| \geq k$ for every vertex $v \in V(G)$. The minimum cardinality of a k-tuple dominating set in G is denoted by $\gamma_{\times k}(G)$. The familiar domination number is thus $\gamma(G) = \gamma_{\times 1}(G)$.

Lemma 2.1 *Let G be a graph with maximum degree Δ and minimum degree δ, and let $\{B, R\}$ be a partition of $V(G)$. Then:*

(i) *If $k \leq \delta - 1$ and B is a $(\delta - k)$-limited packing in G, then R is a $(k+1)$-tuple dominating set in G.*

(ii) *If $k \leq \Delta - 1$ and R is a $(k+1)$-tuple dominating set in G, then B is a $(\Delta - k)$-limited packing in G.*

When the graph is regular even more can be said.

Lemma 2.2 *If G is an r-regular graph, and $k \leq r - 1$, then*

$$\mathrm{L}_{r-k}(G) + \gamma_{\times(k+1)}(G) = |V(G)|.$$

The following bound also involves the domination number, and arises naturally when considering linear programs associated with k-limited packings.

Lemma 2.3 *If G is a graph, then $\mathrm{L}_k(G) \leq k\gamma(G)$. Furthermore, equality holds if and only if for any maximum k-limited packing B in G and any minimum dominating set D in G both the following hold:*

(i) *For any $b \in B$ we have $|N[b] \cap D| = 1$.*

(ii) *For any $d \in D$ we have $|N[d] \cap B| = k$.*

One can bound the size of a k-limited packing solely in terms of the number of vertices in G.

Lemma 2.4 *If G is a connected graph with $|V(G)| \geq 3$, then $\mathrm{L}_2(G) \leq \frac{4}{5}|V(G)|$.*

The upper bound $|B| = \frac{4}{5}|V(G)|$ is achieved only if both inequalities in the proof hold with equality.

If we impose constraints on the minimum degree $\delta(G)$ of G, then similar reasoning gives the following.

Lemma 2.5 *If G is a connected graph, and $\delta(G) \geq k$, then* $\mathrm{L}_k(G) \leq \frac{k}{k+1}|V(G)|$.

This bound can always be achieved; let H be any connected graph, and to each vertex v in H attach a new K_k by making v adjacent to each vertex in the K_k.

When the graph is regular stronger bounds are possible. The following is representative.

Lemma 2.6 *Let G be a cubic graph. Then* $\frac{1}{4}|V(G)| \leq \mathrm{L}_2(G) \leq \frac{1}{2}|V(G)|$.

3 Uniformly 2-limited graphs

A greedy algorithm will quickly find a maximal k-limited packing in a graph, but that set will not usually be a maximum k-limited packing. In this section we consider graphs G where every maximal 2-limited packing in G is a maximum 2-limited packing. We only state the main results (without proof).

Definition 3.1 A graph G is said to be uniformly 2-limited if every maximal 2-limited packing in G has the same cardinality.

For example P_3 is uniformly 2-limited, but P_4 and P_5 are not. The following gives a sufficient condition for a graph G to be uniformly 2-limited.

Lemma 3.2 *Let G be a graph, and let $\{s_1, s_2, \ldots, s_m\}$ be the set of stems in G. Suppose $\{N[s_i] | 1 \leq i \leq m\}$ is a partition of $V(G)$, and if a stem s_i is adjacent to exactly one leaf, then all non-leaf neighbors of s_i have degree 2. Then G is uniformly 2-limited.*

The main result of this section (after a sequence of lemmas) is that the conditions of Lemma 3.2 are also necessary when a uniformly 2-limited graph G contains leaves and has girth at least 11.

Theorem 3.3 *Let G be a connected, uniformly 2-limited graph of girth at least 11. Suppose $\{s_1, s_2, \ldots, s_m\}$ is the set of stems in G, and $m \geq 1$. Then the set $\{N[s_i] | 1 \leq i \leq m\}$ is a partition of $V(G)$, and if a stem s_i is adjacent to exactly one leaf, then all non-leaf neighbors of s_i have degree 2.*

The conditions of Theorem 3.3 are in fact necessary and sufficient conditions for a graph of girth at least 11 (and hence any tree) that contains a stem to be uniformly 2-limited.

In addition it is possible to show the following.

Lemma 3.4 *If G has girth at least 14 and has minimum degree at least 2, then G is not uniformly 2-limited.*

4 Trees T with $\mathrm{L}_2(T) = 2\gamma(T)$

By Lemma 2.3, all graphs G satisfy $\mathrm{L}_2(G) \leq 2\gamma(G)$. In this section (details omitted) a constructive characterization of those trees that attain this bound is given. Note that the graphs considered in the last section are relevant here.

Lemma 4.1 *If T is a tree and T is uniformly 2-limited, then* $\mathrm{L}_2(T) = 2\gamma(T)$.

There are trees, other than the uniformly 2-limited ones, that belong to the collection. The characterization is given by the set $\mathcal{C}$ defined next.

Definition 4.2 Let $\mathcal{C}$ be the set of graphs consisting of P_2 together with any tree that can be obtained from P_2 by any finite sequence of the following operations.

(i) Add a new leaf to any stem s already in the graph.

(ii) Add a new P_3 to the graph, making a leaf of the new P_3 adjacent to any vertex x already in the graph..

(iii) Add a new P_3 to the graph, making the central vertex of the P_3 adjacent to any vertex x already in the graph that is not in some maximum 2-limited packing in the graph.

(iv) Add a new P_5 to the graph, making the central vertex of the P_5 adjacent to any vertex x already in the graph that is not in some maximum 2-limited packing in the graph.

References

[1] J. Arne Telle, Complexity of domination-type problems in graphs, Nordic Journal of Computing 1(1994), 157-171.

[2] John F. Fink and Michael S. Jacobson, On n-domination, n-dependence and forbidden subgraphs, Graph Theory and Its Applications to Algorithms and Computer Science, John Wiley & Sons, Inc. (1985) 301-312.

[3] A. Meir and J.W. Moon, Relations between packing and covering numbers of a tree, Pacific Journal of Mathematics 61 (1975) 225-233.

Edge Coloring of Split Graphs [⋆]

S. M. Almeida [1], C. P. de Mello [2]

Institute of Computing
UNICAMP
Campinas, Brazil

A. Morgana [3]

Department of Mathematics
University of Rome "La Sapienza"
Rome, Italy

Abstract

A *split graph* is a graph whose vertex set admits a partition into a stable set and a clique. The chromatic indexes for some subsets of split graphs, such as split graphs with odd maximum degree and split-indifference graphs, are known. However, for the general class, the problem remains unsolved. This paper presents new results about the classification problem for split graphs as a contribution in the direction of solving the entire problem for this class.

Keywords: edge coloring, split graphs, classification problem.

[⋆] This research was partially supported by CAPES and CNPq (300934/2006-8 and 470420/2004-9).

[1] Email: `sheila@ic.unicamp.br`

[2] Email: `celia@ic.unicamp.br`

[3] Email: `morgana@mat.uniroma1.it`

1 Introduction

A *k-edge coloring* of a graph G is an assignment of one of k colors to each edge of G such that there are no two edges with the same color incident to a common vertex. In the discussion below, a "coloring" of a graph always means an edge coloring, while a "k-coloring" is a coloring that uses only k colors. A k-coloring partitions the set of edges of G into k color classes. The *chromatic index*, $\chi'(G)$, of G is the minimum k such that G has a k-coloring. By definition, $\chi'(G) \geq \Delta(G)$, where $\Delta(G)$ is the maximum degree of G.

In 1964, Vizing [7] showed that for any simple graph G, $\chi'(G) \leq \Delta(G) + 1$. It was the origin of the *Classification Problem*, that consists of deciding whether a given graph G has $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$. In the first case, we say that G is *Class 1*, otherwise, we say that G is *Class 2*. Despite the powerful restriction imposed by Vizing, it is very hard to compute the chromatic index in general. In fact, it is NP-complete to decide if a graph is Class 1 whereas Class 2 recognition is co-NP-complete [4]. In 1991, Cai and Ellis [1] proved that this holds also when the problem is restricted for some classes of graphs.

A *split graph* is a graph whose vertex set admits a partition into a stable set and a clique. The chromatic indexes for some subsets of split graphs, such as split graphs with odd maximum degree [2] and split-indifference graphs [5], are known. However, for the general class, the problem remains unsolved. This paper presents new results about the classification problem for split graphs as a contribution in the direction of solving the entire problem for this class.

2 Definitions and Necessary Background

In this paper, G denotes a simple, finite, undirected and connected graph; $V(G)$ and $E(G)$ are the vertex and edge sets of G. Write $n = |V(G)|$ and $m = |E(G)|$. The maximum degree of G is $\Delta(G)$ and, when there is no ambiguity, we use the simplified notation Δ. A $\Delta(G)$-vertex is a vertex of a graph G with degree $\Delta(G)$. A universal vertex is a vertex with degree $n - 1$. Let v be a vertex of G. The set of vertices adjacent to v in G is denoted by $N(v)$, and $N[v] = \{v\} \cup N(v)$. A *clique* is a set of pairwise adjacent vertices of a graph. A *maximal clique* is a clique that is not properly contained in any other clique. A *stable set* is a set of pairwise no adjacent vertices. A subgraph of G is a graph H with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For $X \subseteq V(G)$, denote by $G[X]$ the subgraph induced by X, that is, $V(G[X]) = X$ and $E(G[X])$ consists of those edges of $E(G)$ having both ends in X. Let $D \subseteq E(G)$. The subgraph induced by D is the subgraph H with $E(H) = D$ and $V(H)$ is the set of vertices v having at least one edge of D incident to v. We denoted by K_n a complete graph with n vertices.

The following lemmas are used in our discussion about the coloring of split graphs.

Lemma 2.1 *[3] The complete graph K_n is Class 1 if, and only if, n is even.*

Lemma 2.2 *[3] Every bipartite graph is Class 1.*

A graph G is *overfull* when $m > \Delta(G) \left\lfloor \frac{n}{2} \right\rfloor$ and n is odd. In [6], Planthold shows that a graph G with n odd and containing a universal vertex is Class 1 if, and only if, G is not overfull.

An *equitable k-coloring* of a graph G is a k-coloring of G such that the sizes of any two color classes differ by at most one. We say that a vertex v misses a color c (or that a color c misses a vertex v) when there is no edge with color c incident to v. Otherwise, we say that the color c appears in v.

Lemma 2.3 *Let n be an odd integer. If K_n is colored with n colors, then each one of these n colors misses exactly one vertex and each vertex misses exactly one color.*

Lemma 2.4 *Let n be an even integer. Then K_n has an equitable n-coloring such that each vertex misses one color, each one of $\frac{n}{2}$ colors misses two vertices, and the other $\frac{n}{2}$ colors appears in every vertex of K_n.*

Lemma 2.5 *Let n be an even integer and $G = K_n \setminus F$, where F is a subset of $E(K_n)$ with $|F| = k$. Then G has an equitable $(n-1)$-coloring, such that there are $k' = \min\{k, n-1\}$ colors missing at least two vertices of G.*

Lemma 2.6 *Let n be an odd integer and $G = K_n \setminus F$, where F is a subset of $E(K_n)$ with $|F| = k$, $k \geq \frac{n-1}{2}$. Then G has an equitable $(n-1)$-coloring, each one of the $\min\{k - \frac{n-1}{2}, \frac{n-1}{2}\}$ colors misses at least three vertices of G, and each one of the remaining $\max\{0, \frac{3(n-1)}{2} - k\}$ colors misses at least one vertex of G.*

3 Coloring Split Graphs

A *split graph* is a graph that admits a partition $\{Q, S\}$ of its set of vertices such that Q is a clique and S is a stable set. The classification problem is solved for some subclasses of split graphs [5,2]. In this work, we solve the classification problem for a new subset of split graphs, presented in Theorem 3.1. This result is strongly based on the work of Planthold [6]. In the following, we assume that Q is a maximal clique.

Theorem 3.1 *Let G be a split graph with even maximum degree. If exists a $\Delta(G)$-vertex v such that $N[v]$ admits a partition $\{L, R\}$ where $|R| = \frac{\Delta(G)}{2}$, $R \subset Q$, the vertices in L are not adjacent to vertices in $V(G) \backslash N[v]$, $\overline{G[L]}$ has k edges, $k \geq \frac{\Delta}{4}$, and R has at most k' $\Delta(G)$-vertices, $k' = \min\{k, \frac{\Delta}{2}\}$, then G is Class 1.*

Proof. Let G be a split graph with partition $\{Q, S\}$ and $\Delta = \Delta(G)$ even. Suppose that there exist a Δ-vertex v of G as described above. Let $P = V(G) \backslash \{L \cup R\}$. (See Fig. 1.) Since, by hypothesis, $|R| = \frac{\Delta}{2}$, then $|L| = \frac{\Delta}{2} + 1$. Hence, the maximum degree of $G[L]$ is at most $\frac{\Delta}{2}$. We consider two cases: $\frac{\Delta}{2}$ is odd and $\frac{\Delta}{2}$ is even.

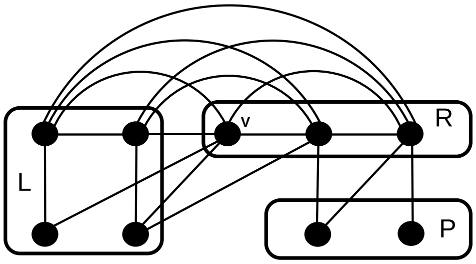


Fig. 1. A split graph G and the subsets L, R and P of $V(G)$.

Case 1: $\frac{\Delta}{2}$ *is odd*

The graph $G[L]$ is isomorphic to a subgraph of $K_{\frac{\Delta}{2}+1}$ and, by hypothesis, $\overline{G[L]}$ has $k \geq \frac{\Delta}{4}$ edges. By Lemma 2.5, $G[L]$ has an equitable $\frac{\Delta}{2}$-coloring where each color c_i misses at least two vertices in L, $1 \leq i \leq k' = \min\{k, \frac{\Delta}{2}\}$.

Let $R = \{v_1, v_2, \dots, v_{\frac{\Delta}{2}}\}$, and let $J = \{v_1, v_2, \dots v_{|J|}\}$ be the subset of vertices of R that are adjacent to every vertex of L. The vertices in J are adjacent to $\frac{\Delta}{2} - 1$ vertices in R and $\frac{\Delta}{2} + 1$ vertices in L, therefore these vertices have degree Δ. By hypothesis, there are at most k' Δ-vertices in R, so $|J| \leq k'$. The graph $G[R]$ is isomorphic to $K_{\frac{\Delta}{2}}$ and $\frac{\Delta}{2}$ is odd, hence, by lemmas 2.1 and 2.3, R can be colored with $\frac{\Delta}{2}$ colors such that each color misses exactly one vertex and each vertex misses one color. By the symmetry of $G[R]$, we can perform the coloring of $G[R]$ such that the color missed by vertex v_i is c_i, $1 \leq i \leq \frac{\Delta}{2}$. Since $|J| \leq k'$ and the vertices in J are adjacent to every vertex in L, each vertex v_i in J is adjacent to a vertex u in L that misses the color c_i. Assign the color c_i to the edge $\{v_i, u\}$. For each vertex v

in $R\backslash J$ with degree $\frac{\Delta}{2}+1$, there is a vertex w in P such that w is adjacent to v. So, assign the color c, missed by v in the coloring of $G[R]$, to the edge $\{v, w\}$. This process can be repeated for every vertex in $R\backslash J$ that is adjacent to $\frac{\Delta}{2}+1$ vertices in $L \cup P$ because the color missed by each vertex in R is distinct of the other ones.

By hypothesis, the vertices in L are not adjacent to vertices in $V(G)\backslash V(H)$. Hence the graph induced by the uncolored edges of G is a bipartite graph and its maximum degree is at most $\frac{\Delta}{2}$. Therefore, by Lemma 2.2, we can color this subgraph with $\frac{\Delta}{2}$ new colors.

Case 2: $\frac{\Delta}{2}$ *is even*

In this case, $G[L]$ is isomorphic to a subgraph of $K_{\frac{\Delta}{2}+1}$ with $\frac{\Delta}{2}$ even, $|E(\overline{G[L]})| = k$ and $k \geq \frac{\Delta}{4}$. So, by Lemma 2.6, $G[L]$ has an equitable $\frac{\Delta}{2}$-coloring such that each one of $p = \min\{k - \frac{\Delta}{4}, \frac{\Delta}{2}\}$ colors misses at least 3 vertices in L and each one of the other $\frac{\Delta}{2} - p$ colors misses at least 1 vertex in L. Let $c_1, c_2, \ldots, c_p$ be the colors missed by at least 3 vertices in L.

The graph $G[R]$ is isomorphic to $K_{\frac{\Delta}{2}}$ and $\frac{\Delta}{2}$ is even. So, by Lemma 2.4, $G[R]$ has an equitable $\frac{\Delta}{2}$-coloring such that each one of $\frac{\Delta}{4}$ colors misses two vertices, each one of the other $\frac{\Delta}{4}$ colors does not miss any vertex in R, and each vertex in R misses exactly one color. The set R is divided in three subsets namely J, J', and N, where: $J = \{v_1, \ldots, v_{|J|}\}$ is the set of the Δ-vertices in R that are adjacent to every vertex in L; $J' = \{v_{|J|+1}, \ldots, v_{|J|+|J'|}\}$ is the set of the Δ-vertices in R that are adjacent to at least one vertex in $P = V(G)\backslash\{L \cup R\}$; and $N = \{v_{|J|+|J'|+1}, \ldots, v_{\frac{\Delta}{2}}\}$ is the set of the vertices in R that are not Δ-vertices. Note that each one of this sets can be an empty set, and $|J| + |J'| + |N| = \frac{\Delta}{2}$.

The symmetry of R allows us to choose which vertex in R misses a specific color. Let $p' = \min\{p, \frac{\Delta}{4}\}$, and let $X = \{v_1, \ldots, v_{2p'}\}$ be the set of vertices in R such that vertices v_{2i-1} and v_{2i} miss the color c_i, $1 \leq i \leq p'$. Note that X can be empty. By hypothesis, there are at most $k' = \min\{k, \frac{\Delta}{2}\}$ Δ-vertices in R, so at least $\frac{\Delta}{2} - k'$ vertices in R are not Δ-vertices. Let $Z = \{v_{k'+1}, \ldots, v_{\frac{\Delta}{2}}\}$ be the set of these vertices and $Y = \{v_{2p'+1}, \ldots, v_{k'}\} = R \setminus (X \cup Z)$. The sets X, Y, and Z are pairwise disjoint and $|X| + |Y| + |Z| = \frac{\Delta}{2}$.

If $k \geq \frac{\Delta}{2}$, then $X = R$ and the sets Y and Z are empty. If $\frac{\Delta}{4} \leq k < \frac{\Delta}{2}$, then Z and Y have size $\frac{\Delta}{2} - k' > 0$. In this case, let $\alpha = \{c_{p'+1}, \ldots, c_{\frac{\Delta}{4}}\}$. Each color of α has to miss two vertices in R and each vertex in $Y \cup Z$ has to miss one color. Since $|\alpha| = |Y| = |Z| = \frac{\Delta}{2} - k'$, for each color c in α, we choose one vertex in Y and one vertex in Z to miss the color c. Now, no two vertices of Y miss the same color.

If X is nonempty, for each vertex v in X that misses a color c and has $\frac{\Delta}{2}+1$ uncolored edges incident to it, we color one of these edges as follows. If v belongs to J, we choose a vertex u that belongs to L and misses the color c, and we assign the color c to the edge $\{v,u\}$. If v belongs to J', there are two cases. If v is adjacent to a vertex w in P such that the color c is not incident to w, then we assign the color c to the edge $\{v,w\}$. Otherwise, v is adjacent to a vertex u that belongs to L and misses the color c, so we assign the color c to the edge $\{v,u\}$. If Y is nonempty, we can color one edge incident to each Δ-vertex that is in Y. For each Δ-vertex v_i in Y, choose a neighbor w in P, and assign the color missed by v_i to $\{v_i,w\}$, $2p'+1 \leq i \leq k'$. Remind that there are no Δ-vertices in Z, so each vertex of Z has at most $\frac{\Delta}{2}$ uncolored incident edges. Now, each Δ-vertex in R has at most $\frac{\Delta}{2}$ uncolored incident edges. The graph induced by the uncolored edges of G is a bipartite graph with a partition $\{L\cup P,R\}$ and maximum degree $\frac{\Delta}{2}$. So, by Lemma 2.2, we can color this subgraph with $\frac{\Delta}{2}$ new colors.

Therefore, by cases 1 and 2, we conclude that G is Class 1. □

We believe that the result of Theorem 3.1 can be extended to other subsets of split graphs. It is easy to see that a split graph with odd $\frac{\Delta}{2}$ containing more than k' Δ-vertices in R but with at most k' vertices adjacent to every vertex in L, while the other conditions of Theorem 3.1 are satisfied, is Class 1.

References

[1] Cai, L., and J. A. Ellis, NP-completeness of edge-colouring some restricted graphs. *Discrete Applied Mathematics*, 30:15–27, 1991.

[2] Chen, B-L., H-L. Fu, and M. T. Ko, Total chromatic number and chromatic index of split graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 17:137–146, 1995.

[3] Fiorini, S., and R. J. Wilson, *Edge-colourings of graphs.* Research Notes in Mathematics, Pitman, 1977.

[4] Holyer, I., The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10:718–720, 1981.

[5] Ortiz, C., N. Maculan, and J. L. Szwarcfiter, Characterizing and edge-colouring split-indifference graphs. *Discrete Applied Mathematics*, 82:209–217, 1998.

[6] Plantholt, M., The chromatic index of graphs with a spanning star. *Journal of Graph Theory*, 5:45–53, 1981.

[7] Vizing, V. G., On an estimate of the chromatic class of a p-graph. *Diskret. Analiz.*, 3:25–30, 1964.

Strong oriented chromatic number of planar graphs without cycles of specific lengths

Mickaël Montassier[1] Pascal Ochem[1] Alexandre Pinlou[1]

LaBRI, Université Bordeaux 1, 351 cours de la libération, 33405 Talence, France

Abstract

A *s*trong oriented k-coloring of an oriented graph G is a homomorphism φ from G to H having k vertices labelled by the k elements of an abelian additive group M, such that for any pairs of arcs $\overrightarrow{uv}$ and $\overrightarrow{zt}$ of G, we have $\varphi(v)-\varphi(u) \neq -(\varphi(t)-\varphi(z))$. The *s*trong oriented chromatic number $\chi_s(G)$ is the smallest k such that G admits a strong oriented k-coloring. In this paper, we consider the following problem: Let $i \geq 4$ be an integer. Let G be an oriented planar graph without cycles of lengths 4 to i. What is the strong oriented chromatic number of G?

1 Introduction

Oriented graphs are directed graphs without loops nor opposite arcs. Let G be an oriented graph. We denote by $V(G)$ its set of vertices and by $A(G)$ its set of arcs. An *oriented k-coloring* of an oriented graph G is a mapping φ from $V(G)$ to a set of k colors such that (1) $\varphi(u) \neq \varphi(v)$ whenever $\overrightarrow{uv}$ is an arc in G, and (2) $\varphi(u) \neq \varphi(x)$ whenever $\overrightarrow{uv}$ and $\overrightarrow{wx}$ are two arcs in G with $\varphi(v) = \varphi(w)$. The *oriented chromatic number* of an oriented graph, denoted by $\chi_o(G)$, is defined as the smallest k such that G admits an oriented k-coloring.

[1] Email: {montassi,ochem,pinlou}@labri.fr

Let G and H be two oriented graphs. A *homomorphism* from G to H is a mapping $\varphi : V(G) \to V(H)$ such that: $\overrightarrow{xy} \in A(G) \Rightarrow \overrightarrow{\varphi(x)\varphi(y)} \in A(H)$

An oriented k-coloring of G can be equivalently defined as a homomorphism from G to H, where H is an oriented graph of order k. Then, the *oriented chromatic number* $\chi_o(G)$ of G can be defined as the smallest order of an oriented graph H such that G admits a homomorphism to H.

The problem of bounding the oriented chromatic number has already been investigated for various graph classes: graphs with bounded maximum average degree [1], graphs with bounded degree [2], graphs with bounded treewidth [7,8], graphs subdivisions [9].

Raspaud and Nešetřil [5] introduced the *strong oriented chromatic number* $\chi_s(G)$. A *strong oriented* k*-coloring* of an oriented graph G is a homomorphism φ from G to H with k vertices labelled by the k elements of an abelian additive group M of order k, such that for any pair of arcs $\overrightarrow{uv}$ and $\overrightarrow{zt}$ of $A(G)$, $\varphi(v) - \varphi(u) \neq -(\varphi(t) - \varphi(z))$. The $s$trong oriented chromatic number $\chi_s(G)$ is the smallest k such that G admits a strong oriented k-coloring.

Therefore, any strong oriented coloring of G is an oriented coloring of G; hence, $\chi_o(G) \leq \chi_s(G)$.

Let M be an additive group and let $S \subset M$ be a set of group elements. The *Cayley digraph* associated with (M, S), denoted by $C_{(M,S)}$, is then defined as follows: $V(C_{(M,S)}) = M$ and $A(C_{(M,S)}) = \{(g, g+s) \ ; g \in M, s \in S\}$. If the set S is a group generator of M, then $C_{(M,S)}$ is connected. Assuming that M is abelian and $S \cap -S = \emptyset$, then $C_{(M,S)}$ is oriented (neither loops nor opposite arcs), and for any pair $(g_1, g_1 + s_1)$ and $(g_2, g_2 + s_2)$ of arcs of $C_{(M,S)}$, $g_1 + s_1 - g_1 \neq -(g_2 + s_2 - g_2)$. Thus, finding a strong oriented k-coloring of an oriented graph G may be viewed as finding a homomorphism from G to an oriented Cayley graph $C_{(M,S)}$ of order k, for some abelian group M with $S \subset M$ and $S \cap -S = \emptyset$.

In the following we will consider the Paley tournament QR_p (where $p \equiv 3 \pmod 4$) is a prime power) that is the Cayley graph $C_{(M,S)}$ with $M = \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ and $S = \{x^2 \ ; \ x \in \mathbb{F}_p \setminus \{0\}\}$.

Strong oriented coloring of planar graphs was recently studied. Sámal [6] proved that every oriented planar graph admits a strong oriented coloring with at most 672 colors. Marshall [3] improved this result and proved the following:

Theorem 1.1 *[3] Let G be an oriented planar graph. Then $\chi_s(G) \leq 271$.*

Borodin *et al.* [1] studied the relationship between the oriented chromatic number and the maximum average degree of a graph, where the maximum average degree, denoted by $Mad(G)$ is: $Mad(G) = \max\{2|E(H)|/|V(H)|, H \subseteq$

$G\}$. Since they considered homomorphisms to oriented Cayley graphs, they proved that if $Mad(G) < 7/3$ (resp. $8/3$, 3, $10/3$) then $\chi_s(G) \leq 5$ (resp. 7, 11, 19). The *girth* of a graph G is the length of a shortest cycle of G. Since every planar graph G with girth g satisfies $Mad(G) < \frac{2g}{g-2}$, it follows that if G is planar with girth at least 14 (resp. 8, 6, 5), then $\chi_s(G) \leq 5$ (resp. 7, 11, 19).

In this paper, we consider the following problem:

Problem 1.2 *Let $i \geq 4$ a integer. Let G be a planar graph without cycles of lengths 4 to i. What is the smallest value k such that $\chi_s(G) \leq k$ for each such G?*

We proved [4] that if G is a planar graph without cycles of lengths 4 to i with $i \geq 5$, then $Mad(G) < 3 + \frac{3}{i-2}$ and that, for any $\epsilon > 0$, there exists a planar graph G without cycles of lengths 4 to i with $3 + \frac{3}{i-2} - \epsilon < Mad(G)$. Consequently, we obtain the following corollary by the above result of Borodin *et al.* [1]:

Corollary 1.3 *Let G be a planar graph without cycles of lengths 4 to 14, $\chi_s(G) \leq 19$.*

A first improvement over Corollary 1.3 is given by the authors [4].

Theorem 1.4 *[4] Every oriented planar graph without cycles of lengths 4 to 11 has a homomorphism to the Cayley graph QR_7.*

In this paper, we continue this study and prove that:

Theorem 1.5 (i) *Every oriented planar graph without cycles of length 4 has a homomorphism to the Cayley graph QR_{43}.*

(ii) *Every oriented planar graph without cycles of lengths 4 and 5 has a homomorphism to the Cayley graph QR_{19}.*

(iii) *Every oriented planar graph without cycles of lengths 4 to 9 has a homomorphism to the Cayley graph QR_{11}.*

In the following, we present a sketch of the proof of Theorem 1.5.(i) based on the *method of reducible configurations and discharging procedure.* Theorems 1.5.(ii) and 1.5.(iii) are based on the same method of proof.

A k-vertex (resp. $^{\geq}k$-vertex, $^{\leq}k$-vertex) is a vertex of degree k (resp. $\geq k$, $\leq k$). The size of a face f, denoted by $d(f)$, is the number of edges on its boundary walk, where each cut-edge is counted twice. A k-face (resp. $^{\geq}k$-face, $^{\leq}k$-face) is a face of size k (resp. $\geq k$, $\leq k$). We say that an edge e is incident to a face f if e belongs to the boundary walk of f.

2 The strong oriented chromatic number of planar graphs without cycles of length 4 is at most 43

Let us define the partial order $\preceq$. Let $n_3(G)$ be the number of $^{\geq}3$-vertices in G. For any two graphs G_1 and G_2, we have $G_1 \prec G_2$ if and only if at least one of the following conditions holds:

- G_1 is a proper subgraph of G_2.
- $n_3(G_1) < n_3(G_2)$.

Note that this partial order is well-defined, since if G_1 is a proper subgraph of G_2, then $n_3(G_1) \leq n_3(G_2)$. So $\preceq$ is a partial linear extension of the subgraph poset.

Let H be a minimal counterexample to Theorem i according to $\prec$.

2.1 Structural properties of H

Let us begin with some definitions: A *light* 4-vertex is a 4-vertex incident to two 3-faces. A *light* 3-face is a 3-face incident to two light 4-vertices.

Claim 2.1 *The counterexample H does not contain:*

(C1) A 1-vertex.

(C2) A 2-vertex incident to a 3-face.

(C3) A 3-vertex.

(C4) A k-vertex adjacent to k 2-vertices with $k \leq 42$.

(C5) A k-vertex adjacent to $k-1$ 2-vertices with $2 \leq k \leq 21$.

(C6) A k-vertex adjacent to $k-2$ 2-vertices with $3 \leq k \leq 11$.

(C7) A k-vertex adjacent to $k-3$ 2-vertices with $4 \leq k \leq 5$.

(C8) A 3-face incident to three 4-vertices.

(C9) A 3-face incident to two 4-vertices and to a 5-vertex which is adjacent to a 2-vertex.

2.2 Discharging procedure

Lemma 2.2 *Let H be a connected plane graph with n vertices, m edges and r faces. Then we have the following:*

(1)
$$\sum_{v \in V(H)} (3d(v) - 10) + \sum_{f \in F(H)} (2d(f) - 10) = -20$$

We define the weight function ω by $\omega(x) = 3 \cdot d(x) - 10$ if $x \in V(H)$ and $\omega(x) = 2 \cdot d(x) - 10$ if $x \in F(H)$. It follows from identity (1) that the total sum of weights is equal to -20. In what follows, we define discharging rules (R1) to (R3) and redistribute weights accordingly. Once the discharging is finished, a new weight function ω^* is produced. However, the total sum of weights is kept fixed by the discharging rules. Nevertheless, we can show that $\omega^*(x) \geq 0$ for all $x \in V(H) \cup F(H)$. This leads to the following obvious contradiction:

$$0 \leq \sum_{x \in V(H) \cup F(H)} \omega^*(x) \leq \sum_{x \in V(H) \cup F(H)} \omega(x) = -20 < 0$$

Thus no such counterexample exists.

The discharging rules are defined as follows:

(R1) Each $^{\geq}6$-vertex gives 2 to each adjacent 2-vertex and to each incident 3-face.

(R2) Each 5-vertex gives 2 to each adjacent 2-vertex, $\frac{3}{2}$ to each incident non light 3-face and 2 to each incident light 3-face.

(R3) Let v be a 4-vertex.

(R3.1) If v is light, then it gives 1 to each incident 3-face

(R3.2) If v is not light, then it gives 2 to each incident 3-face.

Now, let us compute the new charges produced after the discharging procedure. Let v be a k-vertex, with $k \notin \{1, 3\}$ by (C1) and (C3).

If $k = 2$, then $\omega(v) = -4$. Since v is adjacent to $^{\geq}5$-vertices by (C1), (C3), (C5) and (C7), it receives 2 from each adjacent vertices by (R1) and (R2). So, $\omega^*(v) = 0$.

If $k = 4$, then $\omega(v) = 2$. If v is light, by (R3.1) it gives twice 1 and so, $\omega^*(v) = 0$. If v is not light, then v is incident to at most one 3-face. So, $\omega^*(v) \geq 0$ by (R3.2).

If $k = 5$, then $\omega(v) = 5$. By (C7), v is adjacent to at most one 2-vertex. Moreover, it can be incident to at most two 3-faces. If v is adjacent to a 2-vertex, then it is not incident to a light 3-face by (C9) and so, $\omega^*(v) \geq 5 - 2 \cdot \frac{3}{2} - 2 \geq 0$ by (R2). If v is not adjacent to a 2-vertex, then $\omega^*(v) \geq 5 - 2 \cdot 2 \geq 1$.

Observe that (R1) is equivalent for v to give 2 per edge incident to a 2-vertex and 1 per edge incident to a 3-face. It follows that the worst case of discharging for v appears when v is adjacent to the biggest number of 2-vertices according to (C4)-(C7). If $k = 6$, then $\omega(v) = 8$. By (C6), v is adjacent to at most three 2-vertices. So, $\omega^*(v) \geq 8 - 3 \cdot 2 - 2 \geq 0$. If $k = 7$, then $\omega(v) = 11$. By (C6), v is adjacent to at most four 2-vertices. So, $\omega^*(v) \geq 11 - 4 \cdot 2 - 2 \geq 1$.

If $k = 8$, then $\omega(v) = 14$. By (C6), v is adjacent to at most five 2-vertices. So, $\omega^*(v) \geq 14 - 5 \cdot 2 - 2 \geq 2$. If $k = 9$, then $\omega(v) = 17$. By (C6), v is adjacent to at most six 2-vertices. So, $\omega^*(v) \geq 17 - 6 \cdot 2 - 2 \geq 3$. If $k \geq 10$, then $\omega(v) = 3 \cdot k - 10$ and trivially $\omega^*(v) \geq 3 \cdot k - 10 - 2 \cdot k \geq k - 10 \geq 0$.

Let f be a 3-face; $\omega(f) = -4$. By (C2) and (C3), f is incident to $^{\geq}4$-vertices. By (C8), f is incident to at most two 4-vertices. Let x, y, z be the vertices incident to f. Without loss of generality, we consider that $4 \leq d(x) \leq d(y) \leq d(z)$. If $d(z) = 6$, then by (R1)-(R3), f receives at least $2 + 2 \cdot 1 = 4$ and so $\omega^*(f) \geq 0$. Consider $4 \leq d(x) \leq d(y) \leq d(z) \leq 5$. If $d(y) = 5$, then $\omega^*(f) \geq 2 \cdot \frac{3}{2} + 1 \geq 0$. Now, it remains one case: $d(x) = d(y) = 4, d(z) = 5$. If x (resp. y) is not light, then x (resp. y) gives 2 and $\omega^*(f) \geq 2 + 1 + \frac{3}{2} \geq \frac{1}{2}$. Consider that x and y are light; hence f is light and receives 1 from x, 1 from y by (R3) and 2 from z by (R2) and $\omega^*(f) = -4 + 2 \cdot 1 + 2 = 0$.

That shows that $\omega^*(x) \geq 0$ for all $x \in V(H) \cup F(H)$. The contradiction with (1) completes the proof.

References

[1] O.V. Borodin, A.V. Kostochka, J. Nešetřil, A. Raspaud, and É. Sopena. *On the maximum average degree and the oriented chromatic number of a graph.* Discrete Math., **206**, 77–89, 1999.

[2] A. V. Kostochka, É. Sopena, and X. Zhu. *Acyclic and oriented chromatic numbers of graphs.* J. Graph Theory, **24**, 331–340, 1997.

[3] T. H. Marshall. *Antisymmetric flows on planar graphs.* J. Graph Theory, **52**(3), 200–210, 2006.

[4] M. Montassier, P. Ochem, and A. Pinlou. *Strong oriented chromatic number of planar graphs without short cycles.* Technical Report RR-1380-06, LaBRI, 2006.

[5] J. Nešetřil and A. Raspaud. *Antisymmetric flows and strong colorings of oriented planar graphs.* Ann. Inst. Fourier, **49**(3), 1037–1056, 1999.

[6] R. Sámal. *Antisymmetric flows and strong oriented coloring of planar graphs.* Discrete Math., **273**(1-3), 203–209, 2003.

[7] É. Sopena. *The chromatic number of oriented graphs.* J. Graph Theory, **25**, 191–205, 1997.

[8] É. Sopena. *Oriented graph coloring.* Discrete Math., **229**(1-3), 359–369, 2001.

[9] D. R. Wood. *Acyclic, star and oriented colourings of graph subdivisions.* DMTCS, **7**(1), 37–50, 2005.

Homomorphisms of 2-edge-colored graphs

Amanda Montejano[a,1] Pascal Ochem[b,1] Alexandre Pinlou[c,1]
André Raspaud[d,1] Éric Sopena[d,1]

[a] *Dept. Matemàtica Aplicada IV, UPC, Barcelona, Spain*

[b] *LRI, Université de Paris-Sud, 91405 Orsay, France*

[c] *LIRMM, Univ. Montpellier 2, 161 rue Ada, 34392 Montpellier Cedex 5, France*

[d] *LaBRI, Univ. Bordeaux 1, 351 Cours de la Libération, 33405 Talence, France*

Abstract

In this paper, we study homomorphisms of 2-edge-colored graphs, that is graphs with edges colored with two colors. We consider various graph classes (outerplanar graphs, partial 2-trees, partial 3-trees, planar graphs) and the problem is to find, for each class, the smallest number of vertices of a 2-edge-colored graph H such that each graph of the considered class admits a homomorphism to H.

1 Introduction

Our general aim is to study homomorphisms of (n, m)-mixed graphs, that is graphs with both arcs and edges respectively colored with n and m colors. This notion was introduced by Nešetřil and Raspaud [5] as a generalization of the notion of homomorphisms of edge-colored graphs (see e.g. [1]) and the

[1] Emails: amanda@ma4.upc.edu, ochem@lri.fr, pinlou@lirmm.fr, {raspaud,sopena}@labri.fr

notion of oriented coloring (see e.g. [8]). In this paper, we focus on $(0,2)$-mixed graphs, that is 2-edge-colored graphs.

An (n,m)-*mixed graph* is a set of vertices $V(G)$ linked by arcs $A(G)$ and edges $E(G)$, such that the underlying graph is simple (no multiple edges or loops), the arcs are colored with n colors and the edges are colored with m colors. In other words, there is a partition $A(G) = A_1(G) \cup \ldots \cup A_n(G)$ of the set of arcs of G, were $A_i(G)$ contains all arcs with color i and a partition $E(G) = E_1(G) \cup \ldots \cup E_m(G)$ of the edges of G, where $E_j(G)$ contains all edges with color j. We denote the class of (n,m)-mixed graphs by $\mathcal{G}^{(n,m)}$. Observe that $\mathcal{G}^{(0,1)}$ is the class of simple graphs and $\mathcal{G}^{(1,0)}$ is the class of oriented graphs.

Let $G = \{V(G); \bigcup_{i=1}^{n} A_i(G), \bigcup_{j=1}^{m} E_j(G)\}$ and $H = \{V(H); \bigcup_{i=1}^{n} A_i(H), \bigcup_{j=1}^{m} E_j(H)\}$ be two (n,m)-mixed graphs. A *homomorphism* from G to H is a mapping $h : V(G) \rightarrow V(H)$ such that $(h(u), h(v)) \in A_i(H)$ whenever $(u,v) \in A_i(G)$ (for every $i \in \{1, \ldots n\}$), and $h(u)h(v) \in E_j(H)$ whenever $uv \in E_j(G)$ (for every $j \in \{1, \ldots m\}$). The existence of a homomorphism from G to H is denoted by $G \rightarrow H$, and $G \not\rightarrow H$ means there is no such homomorphism.

Given an (n,m)-mixed graph G, the problem is to find the smallest number of vertices of a graph H such that $G \rightarrow H$. This number is denoted by $\chi_{(n,m)}(G)$ and is called the *chromatic number* of the (n,m)-mixed graph G. For a simple graph G, the (n,m)-*mixed chromatic number* is the maximum of the chromatic numbers taken over all the possible (n,m)-mixed graphs having G as underlying graph. Note that $\chi_{(0,1)}(G)$ is the ordinary chromatic number $\chi(G)$, and $\chi_{(1,0)}(G)$ is the oriented chromatic number $\chi_o(G)$. Given a family $\mathcal{F}$ of simple graphs, we denote by $\chi_{(n,m)}(\mathcal{F})$ the maximum of $\chi_{(n,m)}(G)$ taken over all members in $\mathcal{F}$.

Note that a complexity result of Edwards and McDiarmid [3] on the harmonious chromatic number implies that to find the $(0,2)$-mixed chromatic number of a graph is in general an NP-complete problem.

Recall that an *acyclic coloring* of a simple graph G is a proper vertex-coloring satisfying that every cycle of G received at least three colors. The *acyclic chromatic number* of G, denoted by $\chi_a(G)$, is the smallest k such that G admits an acyclic k-vertex coloring. The class of graphs with acyclic chromatic number at most k is denoted by $\mathcal{A}_k$.

Nešetřil and Raspaud [5] proved that the families of bounded acyclic chromatic number have bounded (n,m)-mixed chromatic number. More precisely:

Theorem 1.1 *[5]* $\chi_{(n,m)}(\mathcal{A}_k) \leq k(2n+m)^{k-1}$.

Combining this result with the well-know result of Borodin [2] (every planar graph has an acyclic chromatic number at most 5), we get:

Corollary 1.2 *[5] Let $\mathcal{P}$ be the class of (n, m)-mixed planar graphs. Then $\chi_{(n,m)}(\mathcal{P}) \leq 5(2n + m)^4$.*

This last upper bound extends some previous known results on edge-colored planar graph [1] and on oriented planar graphs [6].

Nešetřil and Raspaud [5] also provided the exact (n, m)-mixed chromatic number of forests ($\mathcal{F}$ denotes the class of (n, m)-mixed forests):

Theorem 1.3 *[5] $\chi_{(n,0)}(\mathcal{F}) = 2n + 1$ and $\chi_{(n,m)}(\mathcal{F}) = 2(n + \lfloor \frac{m}{2} \rfloor + 1)$ for $m \neq 0$.*

Recently, Fabila et al. [4] studied the (n, m)-mixed chromatic number of paths. They proved that it is exactly the same as for the forests; this proves that the lower bound of Theorem 1.3 is reached with paths.

We can obtain new bounds on the (n, m)-mixed chromatic number of partial k-trees, planar graphs, and outerplanar graphs thanks to the above results.

A *k-tree* is a simple graph obtained from the complete graph K_k by repeatedly adding a new vertex adjacent to each vertex of an existing clique of size k. A *partial k-tree* is a subgraph of some k-tree. It is not difficult to see that every partial k-tree has acyclic chromatic number at most $k+1$. We then get the following from Theorem 1.1:

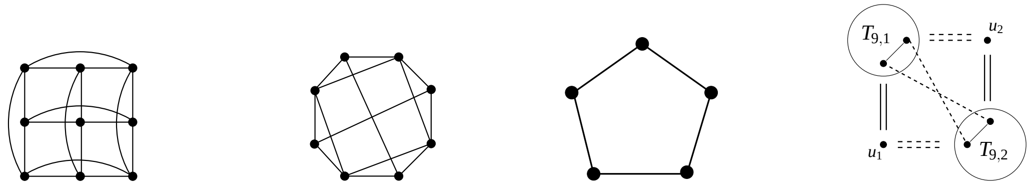
Corollary 1.4 *Let $\mathcal{T}^k$ be the class of (n, m)-mixed partial k-trees. Then $\chi_{(n,m)}(\mathcal{T}^k) \leq (k + 1)(2n + m)^k$.*

In addition, we can derive lower bounds for outerplanar graphs, planar graphs and partial 3-trees from Theorem 1.3 and the result of Fabila et al. [4]:

Corollary 1.5 *Let $\epsilon = 1$ for m odd or $m = 0$, and $\epsilon = 2$ for $m > 0$ even.*

1. *There exist outerplanar graphs G with $\chi_{(n,m)}(G) \geq (2n+m)^2+\epsilon(2n+m)+1$.*
2. *There exist planar partial 3-trees G with $\chi_{(n,m)}(G) \geq (2n + m)^3 + \epsilon(2n + m)^2 + (2n + m) + \epsilon$.*

In this extended abstract, we study the particular class of $(0, 2)$-mixed graphs. More precisely, we give the complete classification for the $(0, 2)$-mixed chromatic number of outerplanar graphs and partial 2-trees with given girth (this improves Corollary 1.4 for $k = 2$). We also provide the exact $(0, 2)$-mixed chromatic number of partial 3-trees. Finally, we obtain upper bounds for the $(0, 2)$-mixed chromatic number of the class of planar graphs with given girth.



(a) $T_9 = C_3 \times C_3$ (b) T_8 (c) $T_5 = C_5$ (d) T_{20}

Fig. 1. The four target graphs T_9, T_8, T_5, and T_{20}.

2 The target graphs

When studying homomorphisms to get bounds on the chromatic number of a graph class $\mathcal{C}$, one often tries to find an *universal* target graph for $\mathcal{C}$, that is a target graph H such that all the graphs of $\mathcal{C}$ admits a homomorphism to H. To prove that a target graph is universal for a graph class, we need "useful" properties. In this section, we construct four $(0, 2)$-mixed target graphs which will be used in the sequel to get upper bounds for $(0, 2)$-mixed chromatic number. Their useful properties are given below.

Consider the three graphs depicted in Figures 1(a), 1(b), and 1(c). These graphs are all self complementary (i.e. isomorphic to their complement). Thus, let T_9 (resp. T_8, T_5) be the complete $(0, 2)$-mixed graphs on 9 (resp. 8, 5) vertices where the edges of each color induce an isomorphic copy of the graphs depicted in Figure 1(a) (resp. 1(b), 1(c)).

Proposition 2.1 *For every pair of distinct vertices u and v of T_9 (resp. T_5) and every $(0, 2)$-mixed k-path $P_k = u_0, u_1, \ldots, u_k$, $k \geq 2$ (resp. $k \geq 3$), there exists a homomorphism h from P_k to T_9 (resp. T_5) such that $h(u_0) = u$ and $h(u_k) = v$.*

Proposition 2.2 *For each $v \in V(T_8)$ and each $(0, 2)$-mixed path of length k, the number of vertices in T_8 reachable from v by such a k-path is at least 3 (resp. 7, 8) if $k = 1$ (resp. $k = 2$, $k \geq 3$).*

For a $(0, 2)$-mixed graph, the edges can get two distinct colors: we will say that the edges with the first color are of *type 1* whereas the others are of *type 2*.

Let T_{20} be the complete $(0, 2)$-mixed graph defined as follows (the construction is illustrated by Fig. 1(d)). Take two disjoint copies of T_9, namely $T_{9,1}$, $T_{9,2}$, and two new vertices u_1 and u_2. We put edges of type 1 (resp. of type 2) linking u_i to all vertices of $T_{9,i}$ (resp. $T_{9,3-i}$) for $1 \leq i \leq 2$. We also add an edge of type 1 (resp. type 2) between $u \in V(T_{9,1})$ and $v \in V(T_{9,2})$ whenever

$uv \in E(T_9)$ is of type 2 (resp. type 1). This construction is known as the *Tromp construction* and was already used to bound the oriented chromatic number (i.e. the $(1,0)$-mixed chromatic number) [7].

Proposition 2.3 *For every triangle u, v, w of T_{20} and every triple $(a, b, c) \in \{1, 2\}^3$, there exists a vertex t adjacent to u, v and, w such that tu (resp. tv, tw) is of type a (resp. b, c).*

3 Results

Let $\mathcal{O}_g$ be the class of $(0, 2)$-mixed outerplanar graphs with girth at least g. Outerplanar graphs form a strict subclass of partial 2-trees (also known as series-parallel graphs); therefore, Corollaries 1.4 and 1.5 implies that $9 \leq \chi_{(0,2)}(\mathcal{O}_3) \leq 12$. We improve this result and characterize the $(0, 2)$-mixed chromatic number of outerplanar graphs for all girth:

Theorem 3.1 $\chi_{(0,2)}(\mathcal{O}_3) = 9$ *and* $\chi_{(0,2)}(\mathcal{O}_g) = 5$ *for* $g \geq 4$.

These bounds are obtained by showing that every $(0, 2)$-mixed outerplanar graph with girth 3 (resp. girth at least 4) admits a homomorphism to T_9 (resp. T_5). To get the second result, we construct, for every girth $g \geq 3$, an outerplanar graph G with girth g and $\chi_{(0,2)}(G) = 5$, which proves that $\chi_{(0,2)}(\mathcal{O}) \geq 5$.

In the same vein, we find the $(0, 2)$-mixed chromatic number of partial 2-trees for all girths ($\mathcal{T}^2_g$ denotes the class of partial 2-trees with girth at least g):

Theorem 3.2 $\chi_{(0,2)}(\mathcal{T}^2_3) = 9$, $\chi_{(0,2)}(\mathcal{T}^2_g) = 8$ *for* $4 \leq g \leq 5$, *and* $\chi_{(0,2)}(\mathcal{T}^2_g) = 5$ *for* $g \geq 6$.

We get the upper bounds by showing that $(0, 2)$-mixed partial 2-trees with girth 3 (resp. 4, 6) admits a homomorphism to T_9 (resp. T_8, T_5). Each lower bound is obtained by constructing a $(0, 2)$-mixed partial 2-tree with the required girth which needs the specified number of colors.

Theorem 1.5 shows that $\chi_{(0,2)}(\mathcal{T}^3) \geq 20$. We prove that this bound is tight:

Theorem 3.3 $\chi_{(0,2)}(\mathcal{T}^3) = 20$.

We get this result by showing that every $(0, 2)$-mixed partial 3-trees admits a homomorphism to T_{20}.

Finally, we bound the $(0, 2)$-mixed chromatic number of sparse graphs. The *maximum average degree* of a simple graph G, denoted by $\mathrm{mad}(G)$, is defined as $\mathrm{mad}(G) = \max\left\{\frac{2|E(H)|}{|V(H)|}, H \subseteq G\right\}$, where $H \subseteq G$ means H is a subgraph of G.

Theorem 3.4 *Let G be a simple graph. If $\mathrm{mad}(G) < \frac{8}{3}$ (resp. $\frac{7}{3}$), then $\chi_{(0,2)}(G) \leq 8$ (resp. $\chi_{(0,2)}(G) = 5$).*

Our proof technique is based on the well-know method of reducible configurations and discharging procedure. We consider a minimal counterexample H to Theorem 3.4. We prove that H does not contain a set S of configurations. Then, we prove, using a discharging procedure, that every graph containing none of the configurations of S has a maximum average degree greater than required by the theorem, that contradicts that H is a counterexample.

Let $\mathcal{P}_g$ be the class of $(0,2)$-mixed planar graphs with girth at least g.

Since every planar graph G with girth g verifies $\mathrm{mad}(G) < \frac{2g}{g-2}$, we get the following corollary for planar graphs with given girth:

Corollary 3.5 $\chi_{(0,2)}(\mathcal{P}_8) \leq 8$ *and* $\chi_{(0,2)}(\mathcal{P}_{14}) = 5$.

References

[1] Alon, N. and T. H. Marshall, *Homomorphisms of edge-colored graphs and coxeter groups*, J. Algebraic Combin. **8** (1998), pp. 5–13.

[2] Borodin, O. V., *On acyclic colorings of planar graphs*, Discrete Math. **25** (1979), pp. 211–236.

[3] Edwards, K. and C. J. H. McDiarmid, *The complexity of harmonious coloring for trees*, Discrete Appl. Math. **57** (1995), pp. 133–144.

[4] Fabila, R., D. Flores, C. Huemer and A. Montejano, *The chromatic number of mixed graphs* (2006), preprint submitted.

[5] Nešetřil, J. and A. Raspaud, *Colored homomorphisms of colored mixed graphs*, J. Comb. Theory Ser. B **80** (2000), pp. 147–155.

[6] Raspaud, A. and É. Sopena, *Good and semi-strong colorings of oriented planar graphs*, Inform. Process. Lett. **51** (1994), pp. 171–174.

[7] Sopena, É., *The chromatic number of oriented graphs*, J. Graph Theory **25** (1997), pp. 191–205.

[8] Sopena, É., *Oriented graph coloring*, Discrete Math. **229(1-3)** (2001), pp. 359–369.

Chromatic Edge Strength of Some Multigraphs

Jean Cardinal[1] Vlady Ravelomanana[2] Mario Valencia-Pabon[2]

Abstract

The edge strength $s'(G)$ of a multigraph G is the minimum number of colors in a minimum sum edge coloring of G. We give closed formulas for the edge strength of bipartite multigraphs and multicycles. These are shown to be classes of multigraphs for which the edge strength is always equal to the chromatic index.

Keywords: graph coloring, minimum sum coloring, chromatic strength

1 Introduction

During a banquet, n people are sitting around a circular table. The table is large, and each participant can only talk to her/his left and right neighbors. For each pair i, j of neighbors around the table, there is a given number m_{ij} of available discussion topics. Assuming that each participant can only discuss one topic at a time, and that each topic takes an unsplittable unit amount of time, what is the minimum duration of the banquet, after which all available topics have been discussed? What is the minimum average elapsed time before a topic is discussed?

In this paper, we show that there always exists a scheduling of the conversations such that these two minima are reached simultaneously. The underlying mathematical problem is that of coloring a *multicycle* with n vertices and m_{ij} parallel edges between consecutive vertices i and j.

[1] Université Libre de Bruxelles, Belgium. jcardin@ulb.ac.be.

[2] Université de Paris-Nord, France. {vlad,valencia}@lipn.univ-paris13.fr.

Let $G = (V, E)$ be a finite undirected (multi)graph without loops. A *vertex coloring* of G is an application from V to a finite set of colors such that adjacent vertices are assigned different colors. The *chromatic number* $\chi(G)$ of G is the minimum number of colors that can be used in a coloring of G. An *edge coloring* of G is an application from E to a finite set of colors such that adjacent edges are assigned different colors. The minimum number of colors in an edge coloring is called the *chromatic index* $\chi'(G)$.

In the sequel, we assume that colors are positive integers. The *vertex chromatic sum* of G is defined as $\Sigma(G) = \min \left\{ \sum_{v \in V} f(v) \right\}$ where the minimum is taken over all colorings f of G. Similarly, the *edge chromatic sum* of G, denoted by $\Sigma'(G)$, is defined as $\Sigma'(G) = \min \left\{ \sum_{e \in E} f(e) \right\}$, where the minimum is taken over all edge colorings. In both case, a coloring yielding the chromatic sum is called a *minimum sum coloring.* The chromatic sum is a useful notion in the context of parallel job scheduling (see [1,9] for example).

We also define the minimum number of colors needed in a minimum sum coloring of G. This number is called the *strength* $s(G)$ in the case of vertex colorings, and the *edge strength* $s'(G)$ in the case of edge colorings. Trivially, we have $s(G) \geq \chi(G)$ and $s'(G) \geq \chi'(G)$.

Previous results. Chromatic sums have been introduced by Kubicka in 1989 [11]. The computational complexity of determining the vertex chromatic sum of a simple graph has been studied extensively. It is NP-hard even when restricted to some classes of graphs for which finding the chromatic number is easy, such as bipartite or interval graphs [2,17]. Approximability results for various classes of graphs were obtained in the last ten years [1,6,9,5]. Similarly, computing the edge chromatic sum is NP-hard for bipartite graphs [7], even if the graph is also planar and has maximum degree 3 [12]. Strong hardness results have also been given for the vertex and edge strength of a simple graph by Salavatipour [16], and Marx [13].

It has been known for long that the vertex strength can be arbitrarily larger than the chromatic number [4]. Nicoloso et al. however showed that $s(G) = \chi(G)$ for proper interval graphs [15]. An analog of Brooks'theorem for the vertex strength of simple graphs has been proved by Hajiabolhassan, Mehrabadi, and Tusserkani [8].

Concerning the relation between the chromatic index and the edge strength, Mitchem, Morriss and Schmeichel [14] proved an inequality, similar to Vizing's theorem. Although it has been conjectured by Harary and Plantholt [18] that $s'(G) = \chi'(G)$ for any simple graph G, this has been disproved by Mitchem et al. [14] and Hajiabolhassan et al. [8].

2 Bipartite Multigraphs

Theorem 2.1 (König's theorem [10]) *Let $G = (V, E)$ be a bipartite multigraph and let Δ denotes its maximum degree. Then $\chi'(G) = \Delta$.*

Let C be the set of colors used in an edge coloring of a multigraph G. We denote by C_x the subset of colors in C assigned to edges incidents with vertex x of G. Moreover, let α and β be two different colors in C, thus a path in G in which the edges alternate between α and β will be called an (α, β)-path. We also denote by $d_G(x)$ the degree of vertex x in G. In [8] it is mentioned (without proof) that $s'(G) = \chi'(G)$ for any bipartite graph G. We now show that in a bipartite multigraph, the edge chromatic sum can always be obtained with $\chi'(G)$ colors.

Theorem 2.2 *Let $G = (V, E)$ be a bipartite multigraph and let Δ denotes its maximum degree. Then $s'(G) = \chi'(G) = \Delta$.*

Proof. We proceed by contradiction. It is sufficient to assume that $s'(G) = \Delta + 1$. So, there is an edge coloring f for G using $\Delta+1$ colors such that $\sum_{e\in E} f(e) = \Sigma'(G)$. Let $C = \{1, \ldots, \Delta + 1\}$ be the set of colors used by f. Choose an edge $[a, b]_0$ in G having color $\Delta + 1$. Clearly, $C_a \cup C_b = \{1, \ldots, \Delta + 1\}$, otherwise, there exists a color $\alpha \in \{1, \ldots, \Delta\}$ not used by any edge adjacent to both vertices a and b which can be used to color edge $[a, b]_0$. We would obtain a new edge coloring f' such that $\sum_{e\in E} f'(e) < \sum_{e\in E} f(e)$ which is a contradiction to the minimality of f. Therefore, there exist colors $\alpha \in C_a \setminus C_b$ and $\beta \in C_b \setminus C_a$ such that $\alpha, \beta \leq \Delta$.

Let $P_{\alpha\beta}$ denotes a maximal (α, β)-path beginning at vertex a. Notice that such a path cannot end at vertex b, otherwise G contains an odd cycle contradicting the fact that G is bipartite. So, we can recolor the edges on $P_{\alpha\beta}$ by swapping colors α and β. Moreover, after such a color swap, color α is such that $\alpha \notin C_a$ and $\alpha \notin C_b$ and thus we can color edge $[a, b]_0$ with color $\alpha \leq \Delta$ obtaining a new edge coloring f'.

We now prove that after such a recoloring, $\sum_{e\in E} f'(e) < \sum_{e\in E} f(e)$ (*). First, note that if the length of $P_{\alpha\beta}$ is even, the recoloring only affects the value of the edge $[a, b]_0$, so (*) holds. Therefore, it is sufficient to consider the effects of such a recoloring when the length of $P_{\alpha\beta}$ is odd. Let $2s+1$ be the length of $P_{\alpha\beta}$, with $s \geq 0$. Thus, initially for f we have the sub-sum $(\Delta + 1) + (s + 1)\alpha + s\beta$ corresponding to edge $[a, b]_0$ and to the $2s + 1$ edges on $P_{\alpha\beta}$. After the recoloring, we have for f' that such values have changed to $\alpha + (s + 1)\beta + s\alpha$. The change value of f' w.r.t. f is $\beta - \Delta - 1 < 0$ and so (*) always holds, contradicting in this way the minimality of f. Therefore, we have proved that if f is an edge coloring for G such that $\sum_{e\in E} f(e) = \Sigma'(G)$, then f uses at most Δ colors to color the edges of G. □

3 Multicycles

Let G be a multigraph without loops with m edges. It is easy to deduce that $\chi'(G) \geq \max\left\{\Delta, \lceil \frac{m}{\tau} \rceil\right\}$, where Δ denotes the maximum degree and τ denotes the cardinality of a maximum matching in G. This lower bound is indeed tight for multicycles, defined as cycles in which we can have parallel edges between two consecutive vertices.

Theorem 3.1 ([3]) *Let $G = (V, E)$ be a multicycle on n vertices with m edges and degree maximum Δ. Let τ denotes the maximal cardinality of a matching in G. Then*

$$\chi'(G) = \begin{cases} \Delta, & \text{if } n \text{ is even,} \\ \max\left\{\Delta, \lceil\frac{m}{\tau}\rceil\right\}, & \text{if } n \text{ is odd.} \end{cases}$$

In order to determine the edge strength of a multicycle, we need the following lemma proved by Berge in [3].

Lemma 3.2 (Uncolored edge Lemma [3]) *Let G be a multigraph without loops with $\chi'(G) = r + 1$. If a coloring of $G \setminus [a, b]_0$ using a set C of r colors cannot be extended to color the edge $[a, b]_0$, then the following identities are verified : (i) $|C_a \cup C_b| = r$; (ii) $|C_a \cap C_b| = d_G(a) + d_G(b) - r - 2$; (iii) $|C_a \setminus C_b| = r - d_G(b) + 1$; (iv) $|C_b \setminus C_a| = r - d_G(a) + 1$.*

Theorem 3.3 *Let $G = (V, E)$ be a multicycle on n vertices with m edges and maximum degree Δ. Let τ denotes the maximal cardinality of a matching in G. Then, $s'(G) = \chi'(G)$.*

Proof. If n is even, by Theorem 2.2, the result follows. So, we assume that $n = 2k+1$ for a positive integer k. We proceed by induction on m. Let $r = \max\left\{\Delta, \lceil\frac{m}{\tau}\rceil\right\}$. Assume that $m = 2k + 1$. In this case, G is a simple odd cycle. Color the edges in G in such a way that there exist k edges colored with color 1, k edges colored with color 2 and one edge colored with color 3. Clearly, it is always possible. As $\chi'(G) = 3$ and k is the size of a maximum matching in G, it is easy to deduce that the theorem holds for this case. Therefore, we assume that $m > 2k + 1$ and assume that the result holds for all multicycles on n vertices with fewer that m edges. Let $[a, b]_0$ be an edge in G and let $G' = G \setminus [a, b]_0$. By induction hypothesis, we have that there exists an edge coloring f' for G' using $r = \max\left\{\Delta, \lceil\frac{m}{\tau}\rceil\right\} \geq \max\left\{\Delta', \lceil\frac{m-1}{\tau}\rceil\right\} \geq \chi'(G')$ colors, such that $\sum_{e\in E'} f'(e) = \Sigma'(G')$, that is, under f' the multigraph G' verifies $s'(G') = \chi'(G') \leq r$. Assume, by contradiction, that $s'(G) = r+1$. Thus, there exists an edge coloring f for G which uses $r+1$ colors and verifies $\sum_{e\in E} f(e) = \Sigma'(G)$. Notice that the restriction of f to edges in G' verifies $\sum_{e\in E'} f(e) = \Sigma'(G')$, otherwise contradicting the optimality of f in G. So, the edge $[a, b]_0$ is the only edge in G colored by f with color $r + 1$. So, let $C = \{1, \ldots, r\}$ be the set of colors used by f on the edges in G' and for each $1 \leq i \leq r$, let E_i denotes the set of edges in G' colored with color i. By induction hypothesis, we have the following claim.

Claim 1 *There exists a color $\sigma \in C$ such that $|E_\sigma| < k$.*

The claim holds, otherwise we would have that $m - 1 = \sum_{i=1}^{r} |E_i| = kr$, and

$r = \frac{m-1}{k} < m/k$, a contradiction.
By Lemma 3.2, we know that $|C_a \cup C_b| = r$. Hence it is sufficient to analyze the cases $\sigma \in C_b \setminus C_a$ (or $\sigma \in C_a \setminus C_b$) and $\sigma \in C_a \cap C_b$.
- Case $\sigma \in C_b \setminus C_a$. By Lemma 3.2, there exists a color $\alpha \in Ca \setminus C_b$. Let $G(\alpha, \sigma)$ denote the induced subgraph of G' by the edges colored by f with colors α and σ. Let $G_b(\alpha, \sigma)$ denote the connected component of $G(\alpha, \sigma)$ containing the vertex b. Clearly, $G_b(\alpha, \sigma)$ is a simple (σ, α)-path having b as end-vertex and not containing vertex a, otherwise, there is a contradiction to Claim 1. So, we can recolor the edges on the path $G_b(\alpha, \sigma)$ by swapping colors α and σ in such a way that $\sigma \notin C_b$. As color $\sigma \notin C_a$, we can color the edge $[a, b]_0$ with color σ obtaining in this way an edge coloring f'' for G which uses r colors.

We want to show that $\sum_{e\in E} f''(e) < \sum_{e\in E} f(e)$ (**), contradicting $s'(G) > r$. If the length of the path $G_b(\alpha, \sigma)$ is even, then $\sum_{e\in E} f''(e) - \sum_{e\in E} f(e) = \sigma - r - 1 \leq r - r - 1 < 0$. If the length of the path $G_b(\alpha, \sigma)$ is odd (say $2s + 1$, with $s \geq 0$), then such a difference is equal to $(\sigma + (s + 1)\alpha + s\sigma) - (r + 1 + (s + 1)\sigma + s\alpha) = \alpha - r - 1 \leq r - r - 1 < 0$. Thus, inequality (**) always holds.
- Case $\sigma \in C_a \cap C_b$. By Lemma 3.2, there exist colors $\alpha \in C_a \setminus C_b$ and $\beta \in C_b \setminus C_a$ with $\alpha \neq \beta \neq \sigma$. By induction hypothesis, the result holds for $G' = G \setminus [a, b]_0$ and G' has a minimum sum edge coloring using at most r colors. Thus, the edge $[a, b]_0$ in G is the only edge colored by f with color $r + 1$.

Let us assume that vertices are ordered clockwise and let b be the right vertex of edge $[a, b]_0$. Recolor edge $[a, b]_0$ by color β and the edge of color β incident to b with color $r + 1$ respectively. Notice that such a procedure does change neither the value of the sum of colors nor the number of colors used. Let $[x, y]_0$ be the current edge colored by such a recoloring with color $r + 1$ such that x is its left vertex, and find a color $\beta_y \in C_y \setminus C_x$. By Lemma 3.2 such a color β_y exists, otherwise there is a color $\theta \leq r$ such that $\theta \notin C_x$ and $\theta \notin C_y$, and so we can recolor edge $[x, y]_0$ with color θ which gives a contradiction to the minimality of f. Repeat such a procedure until current edge $[x, y]_0$ in G colored with color $r + 1$ is such that $\sigma \in C_x \setminus C_y$ or $\sigma \in C_y \setminus C_x$. Clearly it is always possible, because the cycle is odd. Moreover, notice that $|E_\sigma| < k$ always hold. Assume w.l.o.g. that $\sigma \in C_y \setminus C_x$. By relabeling the vertex set of G in such a way that x becomes a and y becomes b, we are back to the first case. This concludes the proof. □

References

[1] A. Bar-Noy, M. Bellare, M. M. Halldórsson, H. Shachnai, and T. Tamir. On chromatic sums and distributed resource allocation. *Information and Computation*, 140(2):183–202, 1998.

[2] A. Bar-Noy and G. Kortsarz. Minimum color sum of bipartite graphs. *J. Algorithms*, 28(2):339–365, 1998.

[3] C. Berge. *Graphs and Hypergraphs.* North-Holland, 1976.

[4] P. Erdös, E. Kubicka, and A. Schwenk. Graphs that require many colors to achieve their chromatic sum. *Congressus Numerantium*, 71:17–28, 1990.

[5] U. Feige, L. Lovász, and P. Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.

[6] K. Giaro, R. Janczewski, M. Kubale, and M. Malafiejski. Approximation algorithm for the chromatic sum coloring of bipartite graphs. In *Proc. of APPROX*, LNCS 2462, pp. 135–145, 2002.

[7] K. Giaro and M. Kubale. Edge-chromatic sum of trees and bounded cyclicity graphs. *Inform. Process. Lett.*, 75(1–2):65–69, 2000.

[8] H. Hajiabolhassan, M. L. Mehrabadi, and R. Tusserkani. Minimal coloring and strength of graphs. *Discrete Math.*, 215(1–3):265–270, 2000.

[9] M. M. Halldórsson, G. Kortsarz, and H. Shachnai. Sum coloring interval and k-claw free graphs with application to scheduling dependent jobs. *Algorithmica*, 37(3):187–209, 2003.

[10] D. König. Gráfok és alkalmazásuk a determinánsok és a halmazok elméletére. *Matematikai és Természettudományi Értesítö*, 34:104–119, 1916.

[11] E. Kubicka and A. J. Schwenk. An introduction to chromatic sums. In *Proc. of the ACM Computer Science Conf.*, pp. 15–21, 1989.

[12] D. Marx. Complexity results for minimum sum edge coloring, 2004. Manuscript.

[13] D. Marx. The complexity of chromatic strength and chromatic edge strength. *Comput. Complex.*, 14(4):308–340, 2006.

[14] J. Mitchem, P. Morriss, and E. Schmeichel. On the cost chromatic number of outerplanar, planar, and line graphs. *Discuss. Math. Graph Theory*, 17(2):229–241, 1997.

[15] S. Nicoloso, M. Sarrafzadeh, and X. Song. On the sum coloring problem on interval graphs. *Algorithmica*, 23(2):109–126, 1999.

[16] M. Salavatipour. On sum coloring of graphs. *Discrete Appl. Math.*, 127(3):477–488, 2003.

[17] T. Szkaliczki. Routing with minimum wire length in the dogleg-free manhattan model is NP-complete. *SIAM J. Comput.*, 29(1):274–287, 1999.

[18] D. West. Open problems section. *The SIAM Activity Group on Discrete Mathematics Newsletter*, 5(2), Winter 1994–95.

Partial Characterizations of Circular-Arc Graphs

F. Bonomo [a,1,3], G. Durán [b,2,4], L.N. Grippo [a,5], M.D. Safe [a,6]

[a] *CONICET and Departamento de Computación, FCEyN, UBA, Buenos Aires, Argentina*

[b] *Departamento de Ingeniería Industrial, FCFM, U. de Chile, Santiago, Chile and Departamento de Matemática, FCEyN, UBA, Buenos Aires, Argentina*

Abstract

A circular-arc graph is the intersection graph of a family of arcs on a circle. A characterization by forbidden induced subgraphs for this class of graphs is not known, and in this work we present a partial result in this direction. We characterize circular-arc graphs by a list of minimal forbidden induced subgraphs when the graph belongs to the following classes: diamond-free graphs, P_4-free graphs, paw-free graphs, and claw-free chordal graphs.

Keywords: circular-arc graphs, claw-free chordal, cographs, diamond-free graphs, paw-free graphs.

[1] Partially supported by UBACyT Grant X184, Argentina and CNPq under PROSUL project Proc. 490333/2004-4, Brazil.

[2] Partially supported by FONDECyT Grant 1050747 and Millennium Science Institute "Complex Engineering Systems", Chile and CNPq under PROSUL project Proc. 490333/2004-4, Brazil.

[3] Email: `fbonomo@dc.uba.ar`

[4] Email: `gduran@dii.uchile.cl,gduran@dm.uba.ar`

[5] Email: `lgrippo@dc.uba.ar`

[6] Email: `mdsafe@dc.uba.ar`

1 Introduction

A graph G is a *circular-arc* (CA) graph if it is the intersection graph of a set $\mathcal{S}$ of arcs on a circle, i.e., if there exists a one-to-one correspondence between the vertices of G and the arcs of $\mathcal{S}$ such that two vertices of G are adjacent if and only if the corresponding arcs in $\mathcal{S}$ intersect. Such a family of arcs is called a *circular-arc* (CA) *model* of G. CA graphs can be recognized in linear time [6]. A graph is *proper circular-arc* (PCA) if it admits a CA model in which no arc is contained in another arc. Tucker gave a characterization of PCA graphs by minimal forbidden induced subgraphs [8]. Furthermore, this subclass can be recognized in linear time [2]. A graph is *unit circular-arc* (UCA) if it admits a CA model in which all the arcs have the same length. Tucker gave a characterization by minimal forbidden induced subgraphs for this class [8]. Recently, linear and quadratic time recognition algorithms for this class were presented [5,3]. Finally, CA graphs that are complements of bipartite graphs were characterized by forbidden induced subgraphs [7].

Nevertheless, the problem of characterizing the whole class of CA graphs by forbidden induced subgraphs remains open. In this work we present some steps in this direction by providing characterizations of CA graphs by minimal forbidden subgraphs when the graph belongs to one of four different classes.

Denote by $N(v)$ the set of neighbours of $v \in V(G)$; by $G|W$ the subgraph of G induced by W, for any $W \subseteq V(G)$; by $\overline{G}$ the complement of G; and by G^* the graph obtained from G by adding an isolated vertex. If t is a nonnegative integer, then tG will denote the disjoint union of t copies of G. A graph G is a *multiple* of another graph H if G can be obtained from H by replacing each vertex x of H by a complete graph K_x and adding all possible edges between K_x and K_y if and only if x and y are adjacent in H.

The graph P_4 is an induced path on 4 vertices. A *paw* is the graph obtained from a complete K_3 by adding a vertex adjacent to exactly one of its vertices. A *diamond* is the graph obtained from a complete K_4 by removing exactly one edge. A *claw* is the complete bipartite graph $K_{1,3}$. A *hole* is an induced cycle of length at least 4. A graph is *chordal* if it does not contain any hole.

Let $A, B \subseteq V(G)$; A *is complete to* B if every vertex of A is adjacent to every vertex of B; and A *is anticomplete to* B if A is complete to B in $\overline{G}$. Let G and H be two graphs; we say that G is an *augmented* H if G is isomorphic to H or if G can be obtained from H by repeatedly adding a universal vertex; and G is a *bloomed* H if there exists a subset $W \subseteq V(G)$ such that $G|W$ is isomorphic to H and $V(G) - W$ induces in G a disjoint union of complete graphs $B_1, B_2, \ldots, B_j$ for some $j \geq 0$, and each B_i is complete to one vertex

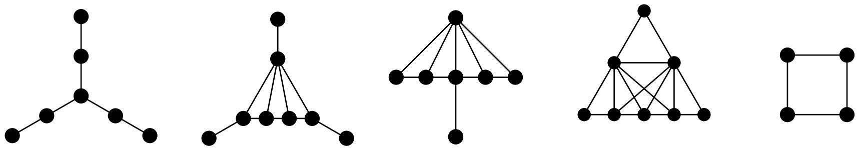


Fig. 1. Minimal forbidden induced subgraphs for the class of interval graphs

of H but anticomplete to the other vertices of H. If each vertex in W is complete to at least one of $B_1, B_2, \ldots, B_j$, we say that G is a *fully bloomed* H. The graphs $B_1, \ldots, B_j$ are the *blooms*. A bloom is *trivial* if it is composed by only one vertex.

Special graphs needed throughout this work are depicted in Figures 1 and 2. We use *net* and *tent* as abbreviations for 2-net and 3-tent, respectively.

Lekkerkerker and Boland determined all the minimal forbidden induced subgraphs for the class of interval graphs, a known subclass of CA graphs.

Theorem 1.1 [4] *The minimal forbidden induced subgraphs for the class of interval graphs are: bipartite claw, n-net for every $n \geq 2$, umbrella, n-tent for every $n \geq 3$, and C_n for every $n \geq 4$ (cf. Figure 1).*

This characterization yields some minimal forbidden induced subgraphs for the class of CA graphs.

Corollary 1.2 [7] *The following graphs are minimally non-CA graphs: bipartite claw, net*, n-net for all $n \geq 3$, umbrella*, (n-tent)* for all $n \geq 3$, and C_n^* for every $n \geq 4$. Moreover, any other minimally non-CA graph is connected.*

We call these graphs *basic* minimally non-CA graphs. Any other minimally non-CA graph will be called *nonbasic*. The following result is a corollary of Theorem 1.1 and Corollary 1.2, and gives a structural property for all nonbasic minimally non-CA graphs.

Corollary 1.3 *If G is a nonbasic minimally non-CA graph, then G has an induced subgraph H which is isomorphic to an umbrella, a net, a j-tent for some $j \geq 3$, or C_j for some $j \geq 4$. In addition, each vertex v of $G - H$ is adjacent to at least one vertex of H.*

2 Partial characterizations

A *cograph* is a graph with no induced P_4. We will call *semicircular graphs* to the intersection graphs of open semicircles on a circle. By definition, semicircular graphs are UCA graphs.

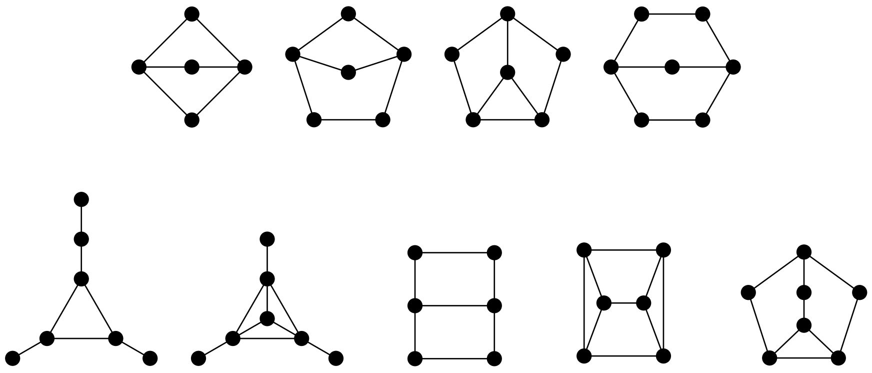


Fig. 2. Some minimally non-CA graphs.

Theorem 2.1 *Let G be a graph. The following conditions are equivalent:*

(i) *G is an augmented multiple of $\overline{tK_2}$ for some nonnegative integer t.*

(ii) *G is a semicircular graph.*

Theorem 2.2 *Let G be a cograph that contains an induced C_4, and such that all its proper induced subgraphs are CA graphs. Then, exactly one of the following conditions holds:*

(i) *G is isomorphic to G_1 or C_4^*.*

(ii) *G is an augmented multiple of $\overline{tK_2}$ for some integer $t \geq 2$.*

Corollary 2.3 *Let G be a cograph. Then, G is a CA graph if and only if G contains neither G_1 nor C_4^* as induced subgraphs.*

Proof. Suppose that H is a cograph minimally non-CA graph and H is not isomorphic to G_1 or C_4^*. Since H is not an interval graph and is P_4-free then, by Theorem 1.1, H contains an induced C_4. By Theorem 2.2, H is an augmented multiple of $\overline{tK_2}$, for some $t \geq 2$. Thus, by Theorem 2.1, H is a circular-arc graph, a contradiction. □

A *paw-free* graph is a graph with no induced paw.

Theorem 2.4 *Let G be a paw-free graph that contains an induced C_4 and such that all its proper induced subgraphs are CA graphs. Then, at least one of the following conditions holds:*

(i) *G is isomorphic to G_1, G_2, G_7, or C_4^*.*

(ii) *G is a bloomed C_4 with trivial blooms.*

(iii) *G is an augmented multiple of $\overline{tK_2}$ for some $t \geq 2$.*

Theorem 2.5 *Let G be a paw-free graph that contains an induced C_j for some $j \geq 5$, and such that all its proper induced subgraphs are CA graphs. Then, exactly one of the following conditions holds:*

(i) *G is isomorphic to G_2, G_4, or C_j^*.*

(ii) *G is a bloomed C_j with trivial blooms.*

Corollary 2.6 *Let G be a paw-free graph. Then G is a CA graph if and only if G contains no induced bipartite claw, G_1, G_2, G_4, G_7, or C_n^* $(n \geq 4)$.*

Proof. Suppose that H is not any of those graphs but it is still a paw-free minimally non-CA graph. In particular, H is not basic. Since H is paw-free then, by Corollary 1.3, H contains an induced C_j for some $j \geq 4$. By Theorem 2.4 and Theorem 2.5, H is an augmented multiple of $\overline{tK_2}$ for some $t \geq 2$ or H is a bloomed C_j with trivial blooms. In both cases H would be a CA graph, a contradiction. □

A graph is *claw-free chordal* if it is chordal and contains no induced claw.

Theorem 2.7 *Let G be a claw-free chordal graph that contains an induced net, and such that all its proper induced subgraphs are CA graphs. Then, exactly one of the following conditions holds:*

(i) *G is isomorphic to a net*, G_5 or G_6.*

(ii) *G is a CA graph.*

Theorem 2.8 [1] *Let G be a connected graph which contains no induced claw, net, C_4, or C_5. If G contains an induced tent, then G is a multiple of a tent.*

Corollary 2.9 *Let G be a claw-free chordal graph. Then, G is CA if and only if G contains no induced tent*, net*, G_5 or G_6.*

Proof. Suppose that H is not any of those graphs but it is still a claw-free chordal minimally non-CA graph. In particular, H is not basic. By Corollary 1.3, H contains an induced net or an induced tent. If H contains an induced net then, by Theorem 2.7, H would be isomorphic to a net*, G_5 or G_6, a contradiction. Thus H contains no induced net but an induced tent. If H is connected, by Theorem 2.8, H is a multiple of a tent and, in particular, a CA graph. Otherwise, H contains a tent*, a contradiction. □

A *diamond-free* graph is a graph with no induced diamond.

Theorem 2.10 *Let G be a diamond-free graph that contains a hole, and such that all its proper induced subgraphs are CA graphs. Then, exactly one of the following conditions holds:*

(i) *G is isomorphic to G_1, G_2, G_3, G_4, G_7, G_8, G_9, or C_j^* for some $j \geq 4$.*

(ii) *G is a CA graph.*

Theorem 2.11 *Let G be a diamond-free chordal graph that contains an induced net, and such that all its proper induced subgraphs are CA graphs. Then, exactly one of the following conditions holds:*

(i) *G is isomorphic to a net*, G_5, or G_6.*

(ii) *G is a fully bloomed triangle, and in consequence, it is a CA graph.*

Corollary 2.12 *Let G be a diamond-free graph. G is CA if and only if G contains no induced bipartite claw, net*, G_1, G_2, G_3, G_4, G_5, G_6, G_7, G_8, G_9, or C_n^* for every $n \geq 4$.*

Proof. Suppose that H is not isomorphic to any of those graphs but it is still a diamond-free minimally non-CA graph. Since H is not an interval graph and it is diamond-free, by Theorem 1.1, H contains either a hole or an induced net. If H contains a hole, it contradicts Theorem 2.10. If H is chordal, it contains an induced net, and so H contradicts Theorem 2.11. □

References

[1] Bang-Jensen, J. and P. Hell, *On chordal proper circular arc graphs*, Discrete Math. **128** (1994), pp. 395–398.

[2] Deng, X., P. Hell and J. Huang, *Linear time representation algorithms for proper circular-arc graphs and proper interval graphs*, SIAM J. Comput. **25** (1996), pp. 390–403.

[3] Durán, G., A. Gravano, R. McConnell, J. Spinrad and A. Tucker, *Polynomial time recognition of unit circular-arc graphs*, J. Algorithms **58** (2006), pp. 67–78.

[4] Lekkerkerker, C. and D. Boland, *Representation of finite graphs by a set of intervals on the real line*, Fund. Math. **51** (1962), pp. 45–64.

[5] Lin, M. and J. Szwarcfiter, *Efficient construction of unit circular-arc models*, in: *Proc. 17th SODA*, 2006, pp. 309–315.

[6] McConnell, R., *Linear-time recognition of circular-arc graphs*, Algorithmica **37** (2003), pp. 93–147.

[7] Trotter, W. and J. Moore, *Characterization problems for graphs, partially ordered sets, lattices, and families of sets*, Discrete Math. **16** (1976), pp. 361–381.

[8] Tucker, A., *Structure theorems for some circular-arc graphs*, Discrete Math. **7** (1974), pp. 167–195.

Partial characterizations of clique-perfect and coordinated graphs: superclasses of triangle-free graphs

Flavia Bonomo [a,c,1], Guillermo Durán [b,2],
Francisco Soulignac, [a,1,3], and Gabriel Sueiro [a,1]

[a] *Departamento de Computación, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina.* {fbonomo,fsoulign,gsueiro}@dc.uba.ar

[b] *Departamento de Ingeniería Industrial, FCFM, Universidad de Chile, Santiago, Chile and Departamento de Matemática, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina.* gduran@dii.uchile.cl

[c] *CONICET, Argentina.*

Abstract

A graph is *clique-perfect* if the cardinality of a maximum clique-independent set equals the cardinality of a minimum clique-transversal, for all its induced subgraphs. A graph G is *coordinated* if the chromatic number of the clique graph of H equals the maximum number of cliques of H with a common vertex, for every induced subgraph H of G. Coordinated graphs are a subclass of perfect graphs. The complete lists of minimal forbidden induced subgraphs for the classes of clique-perfect and coordinated graphs are not known, but some partial characterizations have been obtained. In this paper, we characterize clique-perfect and coordinated graphs by minimal forbidden induced subgraphs when the graph is either paw-free or {gem,W_4,bull}-free, two superclasses of triangle-free graphs.

Keywords: Clique-perfect graphs, coordinated graphs, {gem,W_4,bull}-free graphs, paw-free graphs, perfect graphs, triangle-free graphs.

1 Introduction

A graph G is *perfect* if the chromatic number equals the clique number for every induced subgraph H of G. A graph G is perfect if and only if no induced subgraph of G is an odd hole or an odd antihole [7]. This class of graphs can be recognized in polynomial time [6].

A graph G is *clique-Helly* (CH) if its cliques satisfy the Helly property, and it is *hereditary clique-Helly* (HCH) if H is clique-Helly for every induced subgraph H of G. The *clique graph* $K(G)$ of G is the intersection graph of the cliques of G. A graph G is *K-perfect* if $K(G)$ is perfect.

A *paw* is a triangle with a leaf attached to one of its vertices. A *gem* is a graph of five vertices, such that four of them induce a chordless path and the fifth vertex is universal. A *bull* is a triangle with two leafs attached to different vertices of it. A *wheel* W_j is a graph of $j+1$ vertices, such that j of them induce a chordless cycle and the last vertex is universal. We say that a graph is *anticonnected* if its complement is connected. An *anticomponent* of a graph is a connected component of its complement.

A *clique-transversal* of a graph G is a subset of vertices that meets all the cliques of G. A *clique-independent set* is a collection of pairwise vertex-disjoint cliques. The *clique-transversal number* of G, $\tau_C(G)$, and the *clique-independence number* of G, $\alpha_C(G)$, are the sizes of a minimum clique-transversal and a maximum clique-independent set of G, respectively. Clearly, $\alpha_C(G) \geq \tau_C(G)$, for any graph G. A graph G is *clique-perfect* [10] if $\tau_C(H) = \alpha_C(H)$ for every induced subgraph H of G. The only clique-perfect graphs which are minimally imperfect are the odd antiholes of length $6j+3$, for any $j \geq 1$ [4]. The complexity of the recognition problem for clique-perfect graphs is still not known.

A *K-coloring* of a graph G is a coloring of $K(G)$. A *Helly K-complete* of a graph G is a collection of cliques of G with common intersection. The *K-chromatic number* and the *Helly K-clique number* of G, denoted by $F(G)$ and $M(G)$, are the sizes of a minimum K-coloring and a maximum Helly K-complete of G, respectively. It is easy to see that $F(G) \geq M(G)$ for any graph G. A graph G is *C-good* if $F(G) = M(G)$. A graph G is *coordinated* if

[1] Partially supported by UBACyT Grant X184, Argentina and CNPq under PROSUL project Proc. 490333/2004-4, Brazil.

[2] Partially supported by FONDECyT Grant 1050747 and Millennium Science Institute "Complex Engineering Systems", Chile and CNPq under PROSUL project Proc. 490333/2004-4, Brazil.

[3] The work of this author has been supported by a grant of the YPF Foundation.

every induced subgraph of G is C-good. Coordinated graphs were defined and studied in [3], where it was proved that they are a subclass of perfect graphs. The recognition problem for coordinated graphs is NP-hard and remains NP-complete when restricted to some subclasses of graphs with $M(G) = 3$ [18].

A class of graphs $\mathcal{C}$ is *hereditary* if for every $G \in \mathcal{C}$, every induced subgraph of G also belongs to $\mathcal{C}$.

Finding the complete lists of minimal forbidden induced subgraphs for the classes of clique-perfect and coordinated graphs turns out to be a difficult task ([1,17]). However, some partial characterizations have been obtained in previous works (see [1,2,5,11]). In this paper, we characterize clique-perfect and coordinated graphs by minimal forbidden induced subgraphs when the graph is either paw-free or $\{$gem,W_4,bull$\}$-free, two superclasses of triangle-free graphs. In particular, we prove that in these cases clique-perfect and coordinated graphs are equivalent to perfect graphs and, in consequence, the only forbidden subgraphs are the odd holes. As a direct corollary, we can deduce polynomial time algorithms to recognize clique-perfect and coordinated graphs when the graph belongs to these classes.

2 Main results

Triangle-free graphs were widely studied in the literature, usually in the context of graph coloring problems (see for example [12,13]). It is easy to see that if a graph is triangle-free then it is perfect if and only if it is clique-perfect, if and only if it is coordinated. We shall extend this result by analyzing two superclasses of triangle-free graphs: paw-free and $\{$gem,W_4,bull$\}$-free graphs.

Paw-free graphs were studied in [14]. In this work we prove that the characterization mentioned above for clique-perfect and coordinated graphs on triangle-free graphs also holds on paw-free graphs.

Lemma 2.1 [14] *Let G be a paw-free graph. If G is not anticonnected then the anticomponents of G are stable sets. If G is connected and anticonnected then G is triangle-free.*

We first prove the following auxiliary results.

Theorem 2.2 *Let G be a paw-free, connected and anticonnected graph. Then G is perfect if and only if G is bipartite.*

Theorem 2.3 *Let G be a paw-free graph. If G is not anticonnected, then G is coordinated.*

Now we can prove the main result for paw-free graphs.

Theorem 2.4 *Let G be a paw-free graph. The following statements are equivalent:*

(i) *G is perfect.*

(ii) *G is clique-perfect.*

(iii) *G is coordinated.*

Proof:

(i) ⇒ (ii)) Since the class of paw-free perfect graphs is hereditary, it is enough to see that $\alpha_c(G) = \tau_c(G)$. We can assume that G is connected. If G is anticonnected, then by Theorem 2.2, G is bipartite, and so G is clique-perfect. If G is not anticonnected, then by Lemma 2.1, G has $A_1, \ldots, A_s$ anticomponents with A_i being an stable set, for all $1 \leq i \leq s$. Without loss of generality, we can assume that $|A_1| \leq |A_i|$ $(2 \leq i \leq s)$. Denote $a = |A_1|$. Every clique of G is composed by exactly one vertex of each A_i. Let $v_1^i, \ldots, v_{|A_i|}^i$ be an enumeration of the vertices of A_i (for $1 \leq i \leq s$). For each j $(1 \leq j \leq a)$, let $K_j = \{v_j^1, \ldots, v_j^s\}$. Clearly, K_j is a clique and for $1 \leq i < j \leq a$, $K_j \cap K_i = \emptyset$. Therefore, $K_1, \ldots, K_a$ is a clique-independent set, which implies that $\alpha_c(G) \geq a$. On the other hand, since every clique has a vertex of A_1, then A_1 is a clique-transversal of G. Therefore $\tau_c(G) \leq a$. So, $a \leq \alpha_c(G) \leq \tau_c(G) \leq a$, and hence $\alpha_c(G) = \tau_c(G)$.

(ii) ⇒ (iii)) We can assume that G is connected. If G is not anticonnected, then by Theorem 2.3, G is coordinated. If G is anticonnected, then by Lemma 2.1, G has no triangles and therefore G does not have odd antiholes with length greater than 5. On the other hand, since odd holes are not clique-perfect, G has no odd holes. We conclude that G is perfect. Let G' be an induced subgraph of G. To see that G' is C-good, it is enough to prove that every connected component of G' is C-good. Let H be a connected component of G'. If H is not anticonnected, then by Theorem 2.3, H is coordinated; in particular it is C-good. If H is anticonnected, since it is also connected and perfect, by Theorem 2.2 it follows that H is bipartite. Then H is C-good.

(iii) ⇒ (i)) Coordinated graphs are a subclass of perfect graphs. □

Corollary 2.5 *Clique-perfect and coordinated graphs can be recognized in linear time when the graph is paw-free.*

Bull-free graphs were studied in the context of perfect graphs [8,16], and $\{$gem,$W_4\}$-free graphs in the context of clique-perfect graphs [9]. It is not difficult to build examples of $\{$gem,$W_4\}$-free perfect graphs which are neither clique-perfect nor coordinated. So, the equivalence of Theorem 2.4 does not hold on this class. But we can prove the same equivalence if we also forbid bulls.

First we will show that if {gem,W_4,bull}-free graphs are perfect, they are K-perfect. We prove the following auxiliary results.

Theorem 2.6 *If G is a $\{gem,W_4\}$-free graph then $K(G)$ is a $\{gem,W_4\}$-free graph, hence $K(G)$ contains no odd antihole of length greater than 5.*

Let G be a graph. A *K-hole* $Q_1, \ldots, Q_k$ ($k \geq 4$) is a set of cliques of G which induces a hole in $K(G)$ (i.e., $Q_i \cap Q_j \neq \emptyset \Leftrightarrow i = j$ or $i \equiv j \pm 1 \mod k$). An *intersection cycle* of a K-hole $Q_1, \ldots, Q_k$ is a cycle $v_1, \ldots, v_k$ of G such that $v_i \in Q_i \cap Q_{i+1}$ for every $i, 1 \leq i \leq k$. Let C be a cycle of a graph G. An edge (v, w) of C is *improper* if there is a vertex $z \in C$ such that v, w, z is a triangle. An edge of C is *proper* if it is not improper.

Lemma 2.7 *Let G be a perfect $\{gem,W_4,bull\}$-free graph and $C = v_1, \ldots, v_{2k+1}$ ($k \geq 2$) an intersection cycle of a K-hole $Q_1, \ldots, Q_{2k+1}$. Then C contains neither two consecutive improper edges nor two consecutive proper edges.*

Now we can prove that a perfect {gem,W_4,bull}-free graph is K-perfect.

Theorem 2.8 *If G is a perfect $\{gem,W_4,bull\}$-free graph then G is K-perfect.*

Proof: Suppose G is not K-perfect. By Theorem 2.6, $K(G)$ contains no odd antihole of length greater than 5. Therefore, $K(G)$ contains an odd hole, so there is an odd-length intersection cycle $v_1, \ldots, v_{2k+1}$ ($k \geq 2$) in G. Call $e_i = (v_i, v_{i+1})$ for every $i, 1 \leq i \leq 2k + 1$. By Lemma 2.7 we may assume that e_1 is an improper edge and e_2 is a proper edge. By a repeated application of the same lemma (note that the cycle is odd) we obtain that e_{2k+1} is improper and therefore e_1 is proper, which is a contradiction. □

By the characterization of HCH graphs by forbidden subgraphs [15], {gem,W_4,bull}-free graphs are also HCH. It is known that if $\mathcal{C}$ is an hereditary class of K-perfect clique-Helly graphs, every graph in $\mathcal{C}$ is clique-perfect and coordinated [1,5]. So, since {gem,W_4,bull}-free graphs is an hereditary class of graphs, we obtain as a corollary of Theorem 2.8 the following equivalence.

Theorem 2.9 *Let G be a $\{gem,W_4,bull\}$-free graph. Then G is perfect, if and only if G is clique-perfect, if and only if G is coordinated.*

Corollary 2.10 *Clique-perfect and coordinated graphs can be recognized in polynomial time when the graph is $\{gem,W_4,bull\}$-free.*

It remains as an open problem to determine the “biggest” superclass of triangle-free graphs where perfect, clique-perfect and coordinated graphs are equivalent.

References

[1] Bonomo, F., M. Chudnovsky and G. Durán, *Partial characterizations of clique-perfect graphs I: sublcasses of claw-free graphs*, Discrete Appl. Math. (2007), to appear.

[2] Bonomo, F., M. Chudnovsky and G. Durán, *Partial characterizations of clique-perfect graphs II: diamond-free and Helly circular-arc graphs* (2005), submitted.

[3] Bonomo, F., G. Durán and M. Groshaus, *Coordinated graphs and clique graphs of clique-Helly perfect graphs*, Util. Math. **72** (2007), pp. 175–191.

[4] Bonomo, F., G. Durán, M. Groshaus and J. Szwarcfiter, *On clique-perfect and K-perfect graphs*, Ars Combin. **80** (2006), pp. 97–112.

[5] Bonomo, F., G. Durán, F. Soulignac and G. Sueiro, *Partial characterizations of coordinated graphs: line graphs and complements of forests* (2006), submitted.

[6] Chudnovsky, M., G. Cornuéjols, X. Liu, P. Seymour and K. Vušković, *Recognizing Berge Graphs*, Combinatorica **25** (2005), pp. 143–187.

[7] Chudnovsky, M., N. Robertson, P. Seymour and R. Thomas, *The Strong Perfect Graph Theorem*, Ann. Math. **164** (2006), pp. 51–229.

[8] Chvátal, V. and N. Sbihi, *Bull-free Berge graphs are perfect*, Graphs Combin. **3** (1987), pp. 127–139.

[9] Durán, G., M. Lin and J. Szwarcfiter, *On clique-transversal and clique-independent sets*, Ann. Oper. Res. **116** (2002), pp. 71–77.

[10] Guruswami, V. and C. Pandu Rangan, *Algorithmic aspects of clique-transversal and clique-independent sets*, Discrete Appl. Math. **100** (2000), pp. 183–202.

[11] Lehel, J. and Z. Tuza, *Neighborhood perfect graphs*, Discrete Math. **61** (1986), pp. 93–101.

[12] Maffray, F. and M. Preissmann, *On the NP-completeness of the k-colorability problem for triangle-free graphs*, Discrete Math. **162** (1996), pp. 313–317.

[13] Nilli, A., *Triangle-free graphs with large chromatic numbers*, Discrete Math. **211** (2000), pp. 261–262.

[14] Olariu, S., *Paw-free graphs*, Inf. Process. Lett. **28** (1988), pp. 53–54.

[15] Prisner, E., *Hereditary clique-Helly graphs*, J. Combin. Math. Combin. Comput. **14** (1993), pp. 216–220.

[16] Reed, B. and N. Sbihi, *Recognizing bull-free perfect graphs*, Graphs Combin. **11** (1995), pp. 171–178.

[17] Soulignac, F. and G. Sueiro, *Exponential families of minimally non-coordinated graphs* (2006), submitted.

[18] Soulignac, F. and G. Sueiro, *NP-hardness of the recognition of coordinated graphs*, in: *Ann. XIII CLAIO*, Montevideo, Uruguay, 2006.

On the Packing Chromatic Number of Trees, Cartesian Products and Some Infinite Graphs

Douglas F. Rall [1]

Department of Mathematics
Furman University
Greenville SC, U.S.A.

Boštjan Brešar [2,3]

FEECS
University of Maribor
Maribor, Slovenia

Arthur S. Finbow [4,5]

Department of Mathematics and Computing Science
Saint Mary's University
Halifax, Canada

Sandi Klavžar [3,6]

Department of Mathematics and Computer Science
University of Maribor
Maribor, Slovenia

Abstract

A subset A of vertices in a graph G is an r-packing if the distance between every pair of distinct vertices in A is more than r. The packing chromatic number, $\chi_\rho(G)$, is the smallest k for which there exists some partition $V_1, V_2, \ldots, V_k$ of the vertex set of G such that V_r is an r-packing. We obtain lower and upper bounds for the packing chromatic number of Cartesian products and subdivisions of finite graphs and study the existence of monotone colorings in trees. The infinite, planar triangular lattice and the three dimensional square lattice are shown to have unbounded packing chromatic number.

1 Introduction

There are several equivalent ways to view a proper k-coloring of a graph $G = (V, E)$. One is that of a function $c : V \to \{1, 2, \ldots, k\}$ such that $c(u) \neq c(v)$ for every edge uv of G. In this definition the positive integers are used simply as distinct symbols. Such a coloring function corresponds in a natural way to a partition $V_1, V_2, \ldots, V_k$ of V into independent sets that is obtained by letting $V_i = \{v \mid c(v) = i\}$.

A subset A of V is an r-*packing* if the distance in G between each pair of distinct vertices in A is more than r. The cardinality of a largest r-packing is denoted $\rho_r(G)$. An independent set is thus a 1-packing and a largest such set has cardinality $\alpha(G) = \rho_1(G)$, while a 2-packing is a collection of vertices with pairwise disjoint closed neighborhoods. One application of these distance packings is to coding theory. If G is the n-dimensional hypercube consisting of all bit strings of length n, then a $(2s+1)$-packing in G is a code C that can correct s or fewer transmission errors using the method of closest-distance decoding. See [4]. Perhaps it is desired to place broadcasting stations having signals of various powers at the vertices of a graph. Stations using the same frequency must be placed far enough apart in such a way that the power of their signals do not allow them to propagate so as to interfere with one another. In this model the vertex set must be partitioned with a distance restriction on vertices in the same cell of the partition.

This more restrictive notion of coloring was introduced by Goddard, et al, in [3] under the name broadcast coloring. Since both packings and colorings are involved, we have chosen to use the more descriptive name pack-

[1] Email: doug.rall@furman.edu

[2] Email: bostjan.bresar@uni-mb.si

[3] Support from the Ministry of Science of Slovenia under grant PI-0297

[4] Email: art.finbow@stmarys.ca

[5] NSERC Discovery Grant 8382

[6] Email: sandi.klavzar@uni-mb.si

ing chromatic. Specifically, the *packing chromatic number* of a graph G, denoted $\chi_\rho(G)$, is the smallest positive integer k such that there exists a map $c : V(G) \to \{1, 2, \ldots, k\}$ such that $V_r = c^{-1}(r)$ is an r-packing in G for each $1 \le r \le k$. In this paper we use the term *coloring* to refer either to c or to the accompanying induced partition $V_1, V_2, \ldots, V_k$.

2 Monotone Colorings and Trees

One graphical invariant that has an influence on the size of $\chi_\rho(G)$ is the diameter of G. If $c : V(G) \to \mathbb{N}$ is a packing coloring, then $|c^{-1}(k)| \le 1$ for any $k \ge \text{diam}(G)$. In particular, if G has order n and diameter two, then $\chi_\rho(G) \ge n - \rho_1(G) + 1$. In [3] the authors were able to give an explicit formula for the packing chromatic number of trees of diameter at most four. In addition, with the exception of two special cases, they proved a sharp upper bound of $(n+7)/4$ for $\chi_\rho(T)$ when T is a tree of order n. It is NP-hard to decide if an arbitrary graph G satisfies $\chi_\rho(G) \le 4$, but the complexity of computing the packing chromatic number of a tree is not known. In considering this question we were led to the notion of colorings with the following property. A packing coloring $V_1, V_2, \ldots, V_k$ of G is said to be *monotone* if

$$|V_1| \ge |V_2| \ge \cdots \ge |V_k| .$$

Since $\rho_{r+1}(G) \le \rho_r(G)$ for any G, it seemed reasonable that a graph—in particular, any tree—would have an optimal coloring that is monotone. To the contrary, we discovered an infinite family $\{T_k\}$ of trees that each possess a unique optimal coloring V_1, V_2, V_3 such that $|V_3| = k + 1$ but $|V_2| = 2$. On the other hand, we proved the following result that shows every graph has an optimal coloring that is at least partially monotone.

Proposition 2.1 ([1]) *Let G be any graph and let m be a positive integer such that $m \le \lfloor \frac{\chi_\rho(G)}{2} \rfloor$. Then there exists a packing coloring $c : V(G) \to \{1, 2, \ldots, \chi_\rho(G)\}$ such that $|c^{-1}(m)| \ge |c^{-1}(n)|$ for all $n \ge 2m$.*

3 Cartesian Products and Subdivisions

The *Cartesian product* of two graphs G and H is the graph, $G \,\Box\, H$, whose vertex set is (set) Cartesian product $V(G) \times V(H)$. Two vertices of $G \,\Box\, H$ are adjacent if they are equal in one coordinate and adjacent in the other.

For finite graphs we prove the following generalization of a result of [3].

Theorem 3.1 ([1]) *Let G and H be finite graphs of order at least two. Then*

$$\chi_\rho(G \,\square\, H) \geq (\chi_\rho(G)+1)|V(H)| - \mathrm{diam}(G \,\square\, H)(|V(H)|-1) - 1\,.$$

The *subdivision graph*, $S(G)$, of $G = (V, E)$ is obtained from G be replacing each edge $e = v_i v_j$ of G by a new vertex $v_{i,j}$ and two edges $v_i v_{i,j}$ and $v_j v_{i,j}$. We prove sharp bounds for the packing chromatic number of the subdivision graph $S(G)$ in terms of the clique number and packing chromatic number of G. These bounds are achieved when G is complete.

Theorem 3.2 ([1]) *For any connected graph G of order at least three,*

$$\omega(G) + 1 \leq \chi_\rho(S(G)) \leq \chi_\rho(G) + 1\,.$$

4 Infinite Graphs

The two-way infinite path with the integers as vertex set will be denoted P_∞. Let $\mathcal{R}_2$ denote the planar square lattice (i.e., $\mathcal{R}_2 = P_\infty \,\square\, P_\infty$) while $\mathcal{R}_3 = P_\infty \,\square\, P_\infty \,\square\, P_\infty$. The (planar) hexagonal lattice, $\mathcal{H}$, is isomorphic to the spanning subgraph of $\mathcal{R}_2$ obtained by removing all edges of the form $[(2j-1, 2k), (2j, 2k)]$ and $[(2j, 2k-1), (2j+1, 2k-1)]$. The planar triangular lattice, $\mathcal{T}$, is the plane dual of $\mathcal{H}$. Alternately, $\mathcal{T}$ is isomorphic to the graph obtained from $\mathcal{R}_2$ by adding all edges of the form $[(j, k), (j+1, k-1)]$.

Since the packing chromatic number of a subgraph H of G is bounded above by $\chi_\rho(G)$ we have the following chain of inequalities,

$$3 = \chi_\rho(P_\infty) \leq \chi_\rho(\mathcal{H}) \leq \chi_\rho(\mathcal{R}_2) \leq \chi_\rho(\mathcal{R}_3). \tag{1}$$

In fact, we proved in [1] that the packing chromatic number of $\mathcal{H}$ is either 6, 7 or 8 and in [3] it was shown that $9 \leq \chi_\rho(\mathcal{R}_2) \leq 23$. For the other two infinite graphs defined above we show the following.

Theorem 4.1 ([2]) *The planar triangular lattice, $\mathcal{T}$, and the three dimensional square lattice, $\mathcal{R}_3$ have unbounded packing chromatic number.*

References

[1] Brešar, Bostjan, Sandi Klavžar, and Douglas F. Rall, *On the packing chromatic number of Cartesian products, hexagonal lattice, and trees*, submitted, 1–13.

[2] Finbow, Arthur and Douglas F. Rall, *On the packing chromatic number of some infinite graphs*, manuscript, 2007.

[3] Goddard, Wayne, Sandra M. Hedetniemi, Stephen T. Hedetniemi, John M. Harris, and Douglas F. Rall, *Broadcast chromatic numbers of graphs*, Ars Comb., to appear, 1–21.

[4] Liu, C.L., “Elements of Discrete Mathematics,” 2nd Ed., McGraw-Hill Book Company, New York, N.Y., 1985.

4-cycles in mixing digraphs

Omid Amini [1,2]

I3S (CNRS-UNSA) INRIA
Sophia Antipolis, France

Simon Griffiths [1,2]

DPMMS
Cambridge University, England

Florian Huc [1,2]

I3S (CNRS-UNSA) INRIA
Sophia Antipolis, France

Abstract

It is known that every simple graph with $n^{3/2}$ edges contains a 4-cycle. A similar statement for digraphs is not possible since no condition on the number of arcs can guarantee an (oriented) 4-cycle. We find a condition which does guarantee the presence of a 4-cycle and our result is tight. Our condition, which we call f-mixing, can be seen as a quasirandomness condition on the orientation of the digraph. We also investigate the notion of mixing for regular and almost regular digraphs. In particular we determine how mixing a random orientation of a random graph is.

Keywords: oriented 4-cycles, digraphs, pseudo random digraphs.

[1] This research was supported by the European project AEOLUS, the french region PACA, école Polytechnique and EPSRC.

[2] Email: `omid.amini@sophia.fr,sg332@cam.ac.uk,florian.huc@sophia.fr`

1 Introduction

Throughout this paper we use the notation G for an undirected graph and D for a digraph, V for a vertex set and E for an edge (or arc) set. Unless otherwise stated all digraphs are without loops, multiple arcs or 2-cycles. We write $e(D)$, or simply e, for the number of arcs in a digraph D. We use the notation xy or yx to represent an edge $\{x, y\}$. An arc (x, y) will be denoted $\overrightarrow{xy}$. For two not necessarily disjoint subsets $A, B \subset V$, we denote by $E(A, B) = \{\overrightarrow{uv} \in E : u \in A,\ v \in B\}$ the set of arcs from A to B, and by $e(A, B) = |E(A, B)|$ its cardinality. The *out-neighbours* (resp. *in-neighbours*) of u is the set $\Gamma^{+}(u) = \{v : \overrightarrow{uv} \in E\}$ (resp. $\Gamma^{-}(u) = \{v : \overrightarrow{vu} \in E\}$). The *out-degree* (resp. *in-degree*) of u is $d^{+}(u) = |\Gamma^{+}(u)|$ (resp. $d^{-}(u)$). We shall also mention the *out-out-neighbours* $\Gamma^{++}(u) = \{w : \exists v \in V \text{ such that } \overrightarrow{uv}, \overrightarrow{vw} \in E\}$ and the *in-in-neighbours* $\Gamma^{--}(u)$.
In a simple graph G (resp. a digraph D) a k-cycle consists of k distinct vertices $v_0, \ldots, v_{k-1}$ such that for all $0 \leq i \leq k-1$, $v_i v_{i+1\ modulo\ k}$ $(resp\ \overrightarrow{v_i v_{i+1}}) \in E$.

Finally, for a more complete version, please refer to [1].

1.1 Our problem

It is known that every simple graph with $n^{3/2}$ edges contains a 4-cycle. Specifically, writing $ex(n, C_4)$ for the maximum number of edges a graph G on n vertices can have without containing a 4-cycle, it was shown by Erdős, Rényi and Sós [2] that $ex(n, C_4) = (\frac{1}{2} + o(1))n^{3/2}$, (see also [3] which shows that the example of [2] is best possible).

A similar result for digraphs is not possible, consider the digraph D on vertex set $V = \{1, ..., n\}$, with an arc $\overrightarrow{ij}$ whenever $i < j$. D has $\binom{n}{2}$ arcs (the most possible) but contains no (oriented) 4-cycle. So no condition on the number of arcs can guarantee the existence of an 4-cycle. However, in this example D has an extreme 'bias' in its orientation: we can find subsets $A, B \subset V$ with $e(A, B) = n^2/4$ but $e(B, A) = 0$. Our main result will be that only strongly biased digraphs can avoid containing 4-cycles.

Definition 1.1 Given a digraph D and $f \in \mathbb{R}$, we say D is f*-mixing* if for every (not-necessarily disjoint) pair of subsets $A, B \subset V$ with $e(A, B) \geq f$, we have $e(B, A) > e(A, B)/2$.

In words, a digraph D is f-mixing if whenever there are many arcs from A to B (ie. at least f arcs) then we have more than half as many back. This gives us a large range of mixing conditions getting stronger as f decreases. We can now formulate the main theorem of this paper:

Theorem 1.2 *i There exists $\varepsilon > 0$, such that every $(\varepsilon e^2/n^2)$-mixing digraph D contains a 4-cycle.*

ii In fact D has at least ce^4/n^4 of them (for a constant $c > 0$).

Remark 1.3 i The number of 4-cycles in a typical simple graph is of the order e^4/n^4. Hence the number of (oriented) 4-cycles given to us by the above theorem is (up to multiplication by constant) the largest we could possibly hope for. **In fact any $(\varepsilon e^2/n^2)$-mixing digraph contains the correct order of any orientation of a 4-cycle.**

ii In Section 3, we give examples of digraphs which are (Ke^2/n^2)-mixing (where K is a large constant) and do not contain 4-cycles. This means that the mixing condition cannot be weakened, so Theorem 1.2 is best possible.

iii In the case where e is of order n^2, it is not too difficult to deduce Theorem 1.2 using Szemeredi's regularity Lemma (cf [4] for a survey).

Given a result like Theorem 1.2 it is important to ask: Do there exist digraphs which are $(\varepsilon e^2/n^2)$-mixing? Lemma 1.4, which is obtained by a simple application of Chernoff's inequality, answers this question:

Lemma 1.4 *There exists K such that for any simple graph G on n vertices, the digraph D obtained by orienting the edges of G at random, is Kn-mixing with positive probability (always) and with high probability (as n tends to infinity). In particular, if e is much larger than $n^{3/2}$ then Kn is less than $(\varepsilon e^2/n^2)$ and so D is $(\varepsilon e^2/n^2)$-mixing with positive probability and with high probability.*

In Section 2, we sketch the proof of the first part of Theorem 1.2, the complete proof can be found in [1] as well as all the omitted proofs. In Section 3, we show that the condition $(\varepsilon e^2/n^2)$-mixing can not be weakened to (Ke^2/n^2)-mixing, where K is a large constant. In Section 4, we consider the question of 'how mixing' a digraph can be. In particular, we find a constant $c > 0$ such that randomly oriented random graphs are (with high probability) not cn-mixing. This provides a converse to Lemma 1.4.

2 Proof of Theorem 1.2(i)

Given a digraph D on $V = \{1, ..., n\}$, it simplifies the proof to consider (D, X, Y), *the double cover* of D. (D, X, Y) is defined on vertex set $X \cup Y$ where $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_n\}$, and has arc set $E(D, X, Y) = \{\overrightarrow{x_i y_j}, \overrightarrow{y_i x_j} : \overrightarrow{ij} \in E(D)\}$. If D is f-mixing, then (D, X, Y) has Property 1:

Property 1 *For any pair $A \subset X$ and $B \subset Y$ (or $A \subset Y$ and $B \subset X$) such that $e(A, B) \geq f$, we have $e(B, A) > e(A, B)/2$.*

We take $\varepsilon = 1/32$, if D is $\varepsilon e^2/n^2$-mixing then (D, X, Y) has Property 1 with $f = e^2/32n^2$, we show that this implies the presence of a four cycle in (D, X, Y), which in turn implies the presence of a four cycle in D. We begin by defining for each vertex $x \in X$ the quantity $e_x = e(\Gamma^+(x), \Gamma^{++}(x))$.

Lemma 2.1 *With (D, X, Y) as above, we have:* $\sum_{x\in X} e_x \geq e^2/8n$

We now give a compact version of our proof of Theorem 1.2(i), for the complete version cf [1]. We focus on the set of vertices $W = \{x \in X : e_x \geq e^2/16n^2\}$. In doing so we keep most of the sum $\sum_{x\in X} e_x$:

$$\sum_{x\in W} e_x \geq e^2/16n.$$

Now for each $x \in W$ we have $e(\Gamma^+(x), \Gamma^{++}(x)) \geq \varepsilon e^2/n^2$. Property 1 implies $e(\Gamma^{++}(x), \Gamma^+(x)) \geq e_x/2$. In other words, writing $d^{++}(x, u)$ for $|\Gamma^+(x) \cap \Gamma^+(u)|$, we have for each $x \in W$ that, $\sum_{u\in\Gamma^{++}(x)} d^{++}(x, u) \geq e_x/2$.

Hence: $\displaystyle\sum_u \sum_{x\in\Gamma^{--}(u)} d^{++}(x, u) = \sum_x \sum_{u\in\Gamma^{++}(x)} d^{++}(x, u) \geq \tfrac{1}{2} \sum_{x\in W} e_x \geq \tfrac{e^2}{32n}$

So there exists u with $\sum_{x\in\Gamma^{--}(u)} d^{++}(x, u) \geq e^2/32n^2$, ie. $e(\Gamma^{--}(u), \Gamma^+(u)) \geq e^2/32n^2$. By Property 1, we have: $e(\Gamma^+(u), \Gamma^{--}(u)) \geq e^2/64n^2$. Each arc from $\Gamma^+(u)$ to $\Gamma^{--}(u)$ gives us a 4-cycle so we are done because $e^2/64n^2 > 0$ (as the empty graph is not $\varepsilon e^2/n^2$-mixing). The proof works for $\varepsilon = \frac{1}{32}$.

3 (Ke^2/n^2)-mixing without a 4-cycle

We have proved that every $(\varepsilon e^2/n^2)$-mixing digraph contains a 4-cycle. It is natural to ask whether this result would fail if ε was replaced by a large constant K. In this Section, we give examples of digraphs which are (Ke^2/n^2)-mixing but do not contain any oriented 4-cycle.

Using Erdős-Rényi graphs, we can find for all m, a graph G on m vertices with at least $\frac{1}{20}m^{3/2}$ edges which does not contain a 4-cycle. By Lemma 1.4, a random orientation of G is Lm-mixing with positive probability, where L is a constant. Taking D' as a Lm-mixing orientation of G gives our first example, for D contains no (oriented) 4-cycle and is (Ke^2/m^2)-mixing, for $K = 400L$.

Theorem 1.2 holds for all pairs n, e, the examples above all have $e = \Theta(n^{3/2})$. Are there examples for other pairs n, e? In fact taking a stronger variant of the above example, and then taking a blow up of it in which each vertex is replaced by l vertices and each arc by the corresponding l^2 arcs, we obtain a digraph D with $|V(D)| = ml$ and $e(D) = e(D')l^2$, which is $K'e(D)^2/|V(D)|^2$-mixing (for some K', proof sketched below) which does not contain 4-cycles. Hence, we obtain a wide class of examples, one for each pair m, l. Now for any pair n, e with $e > n^{3/2}$ a suitable choice of m, l yields a digraph D with about n vertices and about e edges which is $K'e^2/n^2$-mixing and contains no 4-cycles.

D is $K'e(D)^2/|V(D)|^2$-mixing for, if there were sets $A, B \subset V(D)$ with $e(A, B) \geq K'e^2/n^2$ and $e(B, A) \leq e(B, A)/2$, then one can find $A', B' \subset V(D)$

each a union of cells of the blow up which also display a strong bias, this contradicts the mixing property of D', and thus no such pair A, B can exist, ie. D is $K'e^2/n^2$-mixing. See [1].

4 How mixing can digraphs be?

Lemma 1.4 says that, if K is a large constant, randomly oriented digraphs are Kn-mixing with high probability. The presence of non necessarily disjoint sets of vertices A and B with arcs from A to B and none from B to A ($e(A, B) > 0$ and $e(B, A) = 0$) is a natural obstruction for a digraph to be highly mixing. It is why we ask the following question:

Given a digraph D, what is $\max_D^{AB} = \max_{\{A,B:e(B,A)=0\}}(e(A, B))$? In fact this question can be asked for any digraph, with or without 2-cycles. Let now state some complexity results about it:

Theorem 4.1 *Given a digraph potentially with 2-cycles, computing* $\max_D^{AB}$ *is NP-hard and hard to approximate within a factor* $n^{1-\varepsilon}$ *for some* $\varepsilon > 0$*.*

In a digraph with multiple arcs, $\max_D^{AB}$ can be equal to zero, but this is strongly due to the presence of 2-cycles. What happens when the digraph has no 2-cycles? We prove that for *almost d-regular digraphs* (digraphs such that for all vertices $v \in V$ we have $d \leq d^-(v), d^+(v) \leq 2d$), $\max_D^{AB}$ is at least linear in n. This means that almost regular graphs cannot be εn-mixing for small ε. To prove it, we construct the sets A and B using the following algorithm:

Algorithm 1

```
Let $v \in V$, set $A = \{v\}$ and $B = \Gamma^+(v)$.
For all $u \in V \setminus A$ evaluate the function:
    $f(u) = e(\{u\}, A) + e(A, \{u, \}) \sum_{v \in A} \Big( |\Gamma^+(v) \cap \Gamma^-(u)| + |\Gamma^+(u) \cap \Gamma^-(v)| \Big).$
while there is a vertex $u$ with $f(u) < d^+(u)$ do
    Add to $A$ the vertex $v$ which maximizes $d^+(v) - f(v)$.
    Update $B$: $B = \Big(B \cup \Gamma^+(v)\Big) \setminus \Big(A \cup \big(B \cap \Gamma^-(v)\big) \cup \big(\Gamma^+(v) \cap \Gamma^-(A)\big)\Big).$
    Update $f(u)$ for all $u \in V \setminus A$.
end while
return $A$ and $B$.
```

Using this algorithm on almost d-regular digraphs, one can prove:

Lemma 4.2 *Let D be an almost d-regular digraphs, there exist subsets $A, B \subset V$ with $e(A, B) \geq \frac{n}{16}$ and $e(B, A) = 0$. So that D is emphatically not $(\frac{n}{16})$-mixing.*

As a random orientation of a random graph gives whp an almost regular digraph, we may deduce:

Corollary 4.3 *Consider the digraph D obtained by randomly orienting the edges of the random graph $G(n,p)$. If $p = \omega(\log n/n)$ then whp there exist subsets $(A,B) \subset V(G)^2$ with $e(A,B) \geq \frac{n}{16}$ and $e(B,A) = 0$. So that with high probability D is emphatically not $(\frac{n}{16})$-mixing for large n.*

5 Conclusion

We have introduced a new notion: the notion of mixing digraphs. This notion can be used to give a tight condition for guaranteeing the presence of a 4-cycle in the digraph. For a fixed digraph D' one can ask whether there exists a mixing condition which guarantees the presence of a copy of D'. So far we have only partial results. We have also investigated how mixing digraphs can be. In particular, almost d-regular digraphs are not $(\frac{n}{16})$-mixing and whp a random orientation of a random graph is Kn-mixing but not εn-mixing where $K > \varepsilon > 0$ are constants. In proving these results we have shown that it is often possible to find subsets A, B with many edges from A to B, while there are none from B to A. Many interesting questions remain such as, is $\max_D^{AB}$ NP-hard in digraphs without 2-cycles, can we find algorithms to better approximate $\max_D^{AB}$.

References

[1] www-sop.inria.fr/sloop/personnel/Florian.Huc/Publications/4cycles.pdf

[2] P. Erdős and A. Rényi and V.T. Sós, On a problem of graph theory, Studia Sci. Math. Hungar. 1, 215-235, 1966

[3] Z. Füredi, Graphs without Quadrilaterals, Journal of Comb. Theory, Series B 34, 187-190, 1983

[4] J. Komlos, A. Shokoufandeh, M. Simonovits, E. Szemeredi, The regularity lemma and its applications in graph theory, Theoretical aspects of computer science (Teheran 2000), 84–112, Springer

Sufficient conditions for a graph to be edge-colorable with maximum degree colors[1]

R. C. S. Machado[2] and C. M. H. de Figueiredo[2]

COPPE, Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil

Abstract

We study properties of graphs related to the existence of certain vertex and edge partitions. These properties give sufficient conditions for a graph to be Class 1 (i.e., edge-colorable with maximum degree colors). We apply these conditions for solving the classification problem for graphs with acyclic *core* (the subgraph induced by the maximum degree vertices), and for subclasses of join graphs and cobipartite graphs.

Keywords: edge-coloring, core of a graph, join graph, cobipartite graph.

1 Introduction

Let $G = (V, E)$ be a simple graph (i.e., without loops or multiple edges). The maximum degree of a vertex in G is denoted by Δ_G. A vertex of maximum degree is called a *delta-vertex* and we denote by Λ_G the set of delta-vertices of G. We denote by $Adj_G(v)$ the set of vertices of G adjacent to vertex v, and by $N_G(v)$ the set $Adj_G(v) \cup \{v\}$. Similarly, for $Z \subset V$, we denote $Adj_G(Z) = \cup_{v \in Z} Adj_G(v)$ and $N_G(Z) = Adj_G(Z) \cup Z$. $G[Z]$ is the subgraph induced by Z.

[1] This research was partially supported by CNPq 470420/2004-9.

[2] Email: {raphael,celina}@cos.ufrj.br

Given a partition (V_L, V_R) of V, with $V_L, V_R \neq \emptyset$, we denote $B_G(V_L, V_R) = (V, \{uv \in E | u \in V_L \wedge v \in V_R\})$ and $D_G(V_L, V_R) = (V, \{uv \in E | u, v \in V_L \vee u, v \in V_R\})$. When the partition is clear, we write only B_G and D_G. Note that B_G is bipartite, D_G is disconnected, and $B_G \cup D_G = G$.

A *k-edge-coloring* of a graph $G = (V, E)$ is a function $\pi : E \to \mathcal{C}$, with $|\mathcal{C}| = k$. A *partial edge-coloring* π' of a graph is an edge-coloring of a subset $E' \subset E$, $\pi' : E' \to \mathcal{C}$. An edge-coloring is *proper* if no two edges incident to the same vertex receive the same color. The *chromatic index* of G, denoted by $\chi'(G)$, is the least k for which G has a proper k-edge-coloring. In the present work, the edge-colorings are assumed to be proper.

Vizing's theorem [1] states that $\chi'(G) = \Delta_G$ or $\chi'(G) = \Delta_G + 1$, defining the *classification problem*. Graphs with $\chi'(G) = \Delta_G$ are said to be *Class 1*; graphs with $\chi'(G) = \Delta_G + 1$ are said to be *Class 2*. It is NP-complete even to determine [3] if a cubic graph has chromatic index 3. The goal of this paper is to find sufficient conditions for classifying graphs as Class 1.

2 Union of Graphs

We say that $G = (V, E)$ is the *union* of $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ if $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$. We write $G = G_1 \cup G_2$, we say that G is a *union graph*, and that G_1 and G_2 are the *operands* of the union operation. Observe that $V_1 \cap V_2$ and $E_1 \cap E_2$ are not necessarily empty.

2.1 Coloring an uncolored edge of a graph

We use Vizing's theorem for determining sufficient conditions for being possible to color an uncolored edge in a partially colored graph without using new colors. The following useful lemma identifies these conditions.

Lemma 2.1 *Let $\pi : E \setminus \{uv\} \to \mathcal{C}$ be a partial Δ_G-edge-coloring of $G = (V, E)$. If $(Adj_G(u) \setminus \{v\}) \cap \Lambda_G = \emptyset$, then G is Δ_G-edge-colorable.*

2.2 Union operations where the maximum degrees of the operands are added

Let $G = G_1 \cup G_2$ with $\Delta_G = \Delta_{G_1} + \Delta_{G_2}$. We consider two cases according to the classification of the operands G_1 and G_2.

Case 1: Both G_1 and G_2 are Class 1. This condition easily implies that no recoloring is needed — we just color the edges of G_1 with Δ_{G_1} colors and the edges of G_2 with additional Δ_{G_2} colors.

Proposition 2.2 *Let $G = G_1 \cup G_2$ be a union of two graphs such that $\Delta_G = \Delta_{G_1} + \Delta_{G_2}$. If G_1 and G_2 are Class 1, then G is Class 1.*

Case 2: Only G_1 is Class 1. Now we state a sufficient condition for the union graph $G = G_1 \cup G_2$ to be Class 1 regardless the classification of G_2. First, we consider the special case where $\Delta_{G_1} = 1$; then, the general case.

Lemma 2.3 *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where E_1 is a matching such that every edge has not both endvertices in $N_{G_2}(\Lambda_{G_2})$. If $G = G_1 \cup G_2$ has maximum degree $\Delta_G = \Delta_{G_2} + 1$, then G is Class 1.*

Proof (Sketch) Consider a $(\Delta_{G_2} + 1)$-edge-coloring of G_2 and add to G_2 all edges of E_1, one at a time. We show that, independently of the order the edges are added, they will always satisfy the conditions of Lemma 2.1. So, no new color will be needed to color $G = G_1 \cup G_2$. Since $\Delta_G = \Delta_{G_2} + 1$, G is Class 1. □

Proposition 2.4 *Let $G = G_1 \cup G_2$ be the union of a Class 1 graph $G_1 = (V_1, E_1)$ and a graph $G_2 = (V_2, E_2)$ such that $\Delta_G = \Delta_{G_1} + \Delta_{G_2}$. If every edge of E_1 has not both endvertices in $N_{G_2}(\Lambda_{G_2})$, then G is Class 1.*

Proof (Sketch) Let $\pi_{G_1} : E_1 \to \{1, ..., \Delta_{G_1}\}$ be an edge-coloring of G, $M_1 = \{e \in E_1 | \pi_{G_1}(e) = 1\}$, $G_1' = (V_1, M_1)$, and $G_1'' = (V_1, E_1 \setminus M_1)$. We show the proposition in two steps: first, we use Lemma 2.3 for showing that $G' = G_1' \cup G_2$ is Class 1; then, by Proposition 2.2, $G' \cup G_1'' = G$ is Class 1. □

2.3 Union graph maximum degree equals the maximum degree of the operands

We investigate union graphs whose maximum degree is equal to the largest of the maximum degrees of the operands. If this operand of largest maximum degree is Class 2, then the union graph is also Class 2. So, we consider unions where the operand of largest maximum degree is Class 1. We show the following sufficient condition for the union graph to be Class 1:

Proposition 2.5 *Let $G = G_1 \cup G_2$ be the union graph of a Class 1 graph $G_1 = (V_1, E_1)$ and a graph $G_2 = (V_2, E_2)$ such that $\Delta_G = \Delta_{G_1}$. If $\Lambda_G = \Lambda_{G_1}$ and all edges of G_2 have not both endvertices in $N_{G_1}(\Lambda_{G_1})$, then G is Class 1.*

Proof (Sketch) We show that we can add to G_1 the edges in E_2 one at a time and they will always be in the conditions of Lemma 2.1. □

3 Applications

We investigate the subgraph induced by the delta-vertices of a graph — named *core* — and by their neighborhoods. We apply our results for solving the classification problem in subclasses of join graphs and cobipartite graphs.

3.1 Subgraphs induced by the delta-vertices and their neighborhoods

The following sufficient condition was established in [2] as a particular case of a theorem about coloring edges of a multigraph. We show that this condition follows from Lemma 2.1:

Proposition 3.1 *Let G be a graph. If $G[\Lambda_G]$ is acyclic, then G is Class 1.*

Proof (Sketch) Consider G' obtained from G by removing the edges of $G[\Lambda_G]$. We use that G' is Δ_G edge-colorable and that we can add to G' the edges of $G[\Lambda_G]$ in such an order that Lemma 2.1 can always be applied.□

We also show that the chromatic index of a graph $G = (V, E)$ depends only on the induced subgraph $G[N_G(\Lambda_G)]$.

Proposition 3.2 *Let $G = (V, E)$ be a graph. Then $\chi'(G) = \chi'(G[N_G(\Lambda_G)])$.*

Proof (Sketch) The result follows from Proposition 2.5 applied to $G_1 = (V_1, E_1) = G[N_G(\Lambda_G)]$ and $G_2 = (V, E \setminus E_1)$. □

3.2 Graph classes

Join graphs. A graph $G = (V, E)$ is the join graph of two vertex disjoint graphs $G_L = (V_L, E_L)$ and $G_R = (V_R, E_R)$ if $V = V_L \cup V_R$ and $E = E_L \cup E_R \cup \{uv : u \in V_L, v \in V_R\}$. In this case, we write $G = G_L + G_R$. It is known [4] that join graphs with $|V_L| \leq |V_R|$ and $\Delta_{G_L} > \Delta_{G_R}$ are Class 1. We show here a different sufficient condition for a join graph to be Class 1.

Theorem 3.3 *Let $G = G_L + G_R$ be the join graph of graphs $G_L = (V_L, E_L)$ and $G_R = (V_R, E_R)$. If $|V_L| \leq |V_R|$ and $\Delta_{G_R} < |V_R| - |V_L|$, then G is Class 1.*

Proof. Let $D_G = D_G(V_L, V_R)$ and $G' = G_R \cup B_G$. The degree in G' of any vertex $v \in V_L$ is $|V_R|$. So, since $|V_R| > \Delta_{G_R} + |V_L|$, the delta-vertices of G' are the vertices of V_L. Observe that these vertices are an independent set of G'. So, by Proposition 3.1, G' is Class 1.

Now, observe that $G = G_L \cup G'$ and that $\Delta_G = \Delta_{G_L} + |V_R| = \Delta_{G_L} + \Delta_{G'}$. Besides, G' is Class 1 and all of its edges have at least one endvertex outside $N_{G_L}(\Lambda_{G_L})$ — those vertices in V_R. So, by Proposition 2.4, G is Class 1. □

We observe that Theorem 3.3 exhibits some Class 1 join graphs not covered by the mentioned result of [4] (for example, $P_2 + C_5$). We observe, also, that the inequality $\Delta_{G_R} < |V_R| - |V_L|$ must be proper, since there are Class 2 join graphs with $\Delta_{G_R} = |V_R| - |V_L|$ (for example, $K_1 + K_{2t} = K_{2t+1}$).

Cobipartite graphs. A graph $G = (V, E)$ is cobipartite if it is the complement of a bipartite graph. In this case, there is a partition (V_L, V_R) of V such that $G[V_L]$ and $G[V_R]$ are complete graphs. We denote $d_i(V_L, V_R) = \max\{d_{B_G(V_L,V_R)}(v)|v \in V_i\}$, $i = L, R$. We consider only connected cobipartite graphs with $V_L, V_R \neq \emptyset$ (otherwise, the problem is reduced to determining the chromatic index of complete graphs). We prove the following:

Theorem 3.4 *Let $G = (V, E)$ be a connected cobipartite graph with partition (V_L, V_R), $0 < |V_L| < |V_R|$. If $d_L(V_L, V_R) \leq d_R(V_L, V_R)$, then G is Class 1.*

Proof. Let $D_G = D_G(V_L, V_R)$ and $B_G = B_G(V_L, V_R)$. Since the delta-vertices of D_G are in V_R, all edges of B_G have one endvertex outside $N_{D_G}(\Lambda_{D_G})$ (those in V_L). Since $d_R(V_L, V_R) \geq d_L(V_L, V_R)$, we have $\Delta_{B_G \cup D_G} = \Delta_{B_G} + \Delta_{D_G}$ and we can apply Proposition 2.4 for showing that $G = D_G \cup B_G$ is Class 1. □

We prove in Theorem 3.6 that, if $|V_L| = |V_R|$, the cobipartite graph is Class 1. In the proof we use the following lemma:

Lemma 3.5 *Let $G_L = (V_L, E_L)$ and $G_R = (V_R, E_R)$ be two vertex disjoint complete graphs of equal order k and $G_{LR} = G_L \cup G_R$. Now, let $G' = (V', E')$ be a graph where $V' = V_L \cup V_R$ and E' is a matching such that each edge has one endvertex in V_L and the other in V_R. Then $G = G_{LR} \cup G'$ is Class 1.*

Proof (Sketch) If k is even, then G_{LR} is Class 1. Since G' is Class 1 and $\Delta_G = \Delta_{G_{LR}} + \Delta_{G'}$, by Proposition 2.2, G is Class 1. Now, suppose k odd. Construct a k-edge-coloring of G_{LR} in such a way that if $uv \in E'$, then u and v have the same free color. Now we can color every edge $uv \in E'$ with the free color of u and v, which gives a k-edge-coloring of G. □

Theorem 3.6 *Let $G = (V, E)$ be a connected cobipartite graph with partition (V_L, V_R), $|V_L| = |V_R|$. Then G is Class 1.*

Proof (Sketch) Denote $D_G = D_G(V_L, V_R)$ and $B_G = B_G(V_L, V_R)$ and let π_B be a Δ_{B_G}-edge-coloring of B_G. Let $G' = (V, E')$, where E' is a matching defined by edges of B_G colored with an arbitrary color c in π_B. By Lemma 3.5, $G_M = D_G \cup G'$ is Class 1. Now, consider $G'' = B_G \setminus E'$, which is Class 1 and has degree is $\Delta_{G''} = \Delta_{B_G} - 1$. Since $G = G_M \cup G''$ has degree $\Delta_G = \Delta_{D_G} + 1 + \Delta_{B_G} - 1 = \Delta_{G_M} + \Delta_{G''}$, by Proposition 2.2, G is Class 1. □

4 Conclusion

In this work we investigated decomposition tools for the edge-coloring classification problem and applied those tools in subclasses of join graphs and cobipartite graphs. We also considered the role of the core of a graph and its neighborhood in the edge-coloring problem.

Besides those immediate applications, our decomposition results lead to some problems, interesting on their own, related to the existence of certain vertex partitions and edge orderings. We describe, next, one of such problems.

C1-Partition. For showing that a given graph $G = (V, E)$ is Class 1, we have applied a number of times the strategy of finding a partition (V_1, V_2) of V with the following properties:

(i) $\Delta_{G[V_1]} > \Delta_{G[V_2]}$,

(ii) $\Lambda_{B_G(V_1,V_2)} \cap \Lambda_{G_1} \neq \emptyset$,

with $B_G(V_1, V_2)$ having a non-empty edge set. We call such partition a *C1-partition.* If G has a C1-partition, then it is Class 1. This strategy was easily applied to some join graphs and cobipartite graphs. A natural question is: can this strategy be applied to more general Class 1 graphs?

Observe that, although having a C1-partition is sufficient for being Class 1, there are Class 1 graphs that do not have a C1-partition (for example, C_4). So, it would be interesting to know for which graph classes having a C1-partition is equivalent to being Class 1, and whether there exists any polynomial time algorithm for determining if a graph has a C1-partition — at least for those classes. This problem can be viewed as a decision problem whose input is a graph $G = (V, E)$ and whose answer is YES if this graph has a C1-partition.

References

[1] Vizing, V. G. *On an estimate of the chromatic class of a p-graph. (Russian)* Diskret. Analiz **3** (1964) 25–30.

[2] Fournier, J.-C. *Méthode et théorème générale de coloration des arêtes.* J. Math. Pures Appl. **56** (1977), 437–453.

[3] Holyer, I. *The NP-completeness of edge-coloring.* SIAM J. Comput. **10** (1981), no.1, 718–720.

[4] De Simone, C., and C. P. de Mello, *Edge-colouring of join graphs.* Theoret. Comput. Sci. **355** (2006), no. 3, 364–370.

The Football Pool Polytope

Javier L. Marenco[1]

Computer Science Dept., FCEN, University of Buenos Aires, Argentina
Sciences Institute, National University of General Sarmiento, Argentina

Pablo A. Rey[2]

Departamento de Ingeniería Industrial, FCFM, Universidad de Chile, Chile

Abstract

The football pool problem asks for the minimun number of bets on the result on n football matches ensuring that some bet correctly predicts the outcome of at least $n-1$ of them. This combinatorial problem has proven to be extremely difficult, and is open for $n \geq 6$. Integer programming techniques have been applied to this problem in the past but, in order to tackle the open cases, a deep knowledge of the polytopes associated with the integer programs modeling this problem is required. In this work we address this issue, by defining and studying the football pool polytope in connection with a natural integer programming formulation of the football pool problem. We explore the basic properties of this polytope and present several classes of facet-inducing valid inequalities over natural combinatorial structures in the original problem.

Keywords: football pool, polyhedral combinatorics

[1] Partially supported by Ubacyt Grant X212, Foncyt Grant 11-09112 (Argentina), and CNPq under Prosul project Proc. 490333/2004-4 (Brazil). Email: `jmarenco@dc.uba.ar`

[2] Partially supported by Fondecyt Grant 1040727, Millennium Science Nucleus Complex Engineering Systems F-06-04 (Chile), and CNPq under Prosul project Proc. 490333/2004-4 (Brazil). Email: `prey@dii.uchile.cl`

1 Introduction

Let $\mathcal{S} = \{0, 1, 2\}$ be a set of symbols and, given a positive integer n, define $\mathcal{A}_n$ to be the set of all strings of length n over the alphabet $\mathcal{S}$. The elements of $\mathcal{A}_n$ are called *codewords*. If $a, b \in \mathcal{A}_n$, we say that *a covers b* if they differ in at most one symbol, i.e., if the Hamming distance between them is 0 or 1. The *football pool problem* asks for the minimum-cardinality subset $C \subset \mathcal{A}_n$ such that every codeword in $\mathcal{A}_n$ is covered by some codeword in C [1].

The football pool problem has been solved for $n \leq 5$ [3], and for $n = (3^k - 1)/2$ for any natural number k [4] (i.e., when there exists a ternary Hamming code of length n). The football poll problem for $n \geq 6$ is an important open combinatorial problem [5]. Integer programming techniques have been applied to this problem in the past but, in order to tackle the open cases, a deep knowledge of the polytopes associated with the integer programs modeling this problem is required. In this work we address this issue, by defining and studying the *football pool polytope* in connection with a natural formulation of the football pool problem as an integer program.

If the codeword length n is not explicitly needed, we write $\mathcal{A}$ instead of $\mathcal{A}_n$. If $a, b \in \mathcal{A}$, we denote by $\text{dist}(a, b)$ the number of symbols in which a and b differ (i.e, the Hamming distance between a and b). For $a \in \mathcal{A}$, we define the *neighborhood* of a to be $N(a) = \{x \in \mathcal{A} : \text{dist}(a, x) = 1\}$. Moreover, if $C \subseteq \mathcal{A}$, we define the *neighborhood* of C to be $N(C) = \{x \in \mathcal{A} \backslash C : \text{dist}(a, x) = 1$ for some $a \in C\}$. Finally, for $a \in \mathcal{A}$ we define the *closed neighborhood* of a to be $N[a] = N(a) \cup \{a\}$ and, similarly, for $C \subseteq \mathcal{A}$ we define $N[C] = N(C) \cup C$.

2 The football pool polytope

If $C \subseteq \mathcal{A}$ is a set of codewords such that every codeword in $\mathcal{A}$ is covered by some codeword in C, we say that C is a *feasible set*. The *incidence vector* associated with a feasible set C is the vector $x^C \in \{0, 1\}^{|\mathcal{A}|}$ such that $x^C_a = 1$ if and only if $a \in C$, for every $a \in \mathcal{A}$.

Definition 2.1 [football pool polytope] We define the *football pool polytope* $\textsc{Fpp}_n \subseteq \mathbf{R}^{|\mathcal{A}_n|}$ to be the convex hull of the incidence vectors associated to all the feasible sets of $\mathcal{A}_n$.

When the codeword length n is not explicitly needed, we write $\textsc{Fpp}$ instead of $\textsc{Fpp}_n$. With this definition, we can state the football pool problem as

$$\min \Big\{ \sum_{a \in \mathcal{A}} x_a : x \in \textsc{Fpp} \Big\}.$$

The football poll polytope FPP is a special set covering polytope, hence we can explore the basic properties and facets of FPP by resorting to known results on set covering polytopes [2,6]. It is not difficult to verify that the polytope FPP is full-dimensional [2]. For every $a \in \mathcal{A}$, the binary bounds $0 \leq x_a$ and $x_a \leq 1$ are trivial valid inequalities of FPP and, moreover, a straightforward argument shows that these inequalities define facets of FPP.

Definition 2.2 [point inequalities] If $a \in \mathcal{A}$, we define

$$\sum_{z \in N[a]} x_z \ \geq \ 1 \tag{1}$$

to be the *point inequality* associated with the codeword a.

Theorem 2.3 *The point inequalities (1) are valid and facet-inducing for* FPP.

Note that the point inequalities (1) together with the binary constraints $x_a \in \{0,1\}$ for every $a \in \mathcal{A}$ define an integer programming formulation for FPP, i.e., this polytope equals the convex hull of all points $x \in \{0,1\}^{|\mathcal{A}|}$ satisfying the point inequalities.

3 Box inequalities

We define a set $B = \{a,b,c,d\} \subseteq \mathcal{A}$ to be a *box* if $\text{dist}(a,b) = 2$, and c and d are the only two codewords with $\text{dist}(a,c) = \text{dist}(c,b) = 1$ and $\text{dist}(a,d) = \text{dist}(d,b) = 1$ (i.e., $N(a) \cap N(b) = \{c,d\}$). For example, if $a = 00000$ and $b = 11000$, then $c = 10000$ and $d = 01000$. Note that c and d are the only two codewords which differ from a in only one symbol and differ from b in the other symbol separating a from b.

Definition 3.1 [box inequalities] If $B = \{a,b,c,d\}$ is a box, we define

$$\sum_{z \in B} x_z + \sum_{z \in N(B)} x_z \ \geq \ 2 \tag{2}$$

to be the *box inequality* associated with B.

Theorem 3.2 *The box inequalities (2) are valid and facet-inducing for* FPP.

For $x \in \{a,b\}$, define $N_B(x) = N(x) \backslash B$ and, for $x \in \{c,d\}$, define $N_B(x) = N(x) \backslash (B \cup N_B(a) \cup N_B(b))$. Note that $N_B(x)$ and $N_B(y)$ are disjoint if $x \neq y$.

Definition 3.3 [reinforced box inequalities] If $B = \{a,b,c,d\}$ is a box, we define

$$2x_d + \sum_{z \in B \backslash \{d\}} x_z + (x_{N_B(a)} + x_{N_B(b)} + x_{N_B(d)}) \ \geq \ 2 \tag{3}$$

to be the *reinforced box inequality* associated with B and d.

Theorem 3.4 *The reinforced box inequalities (3) are valid and facet-inducing for* FPP.

If $\pi x \leq \pi_0$ is a valid inequality for FPP, we define its *support* to be $\mathrm{supp}(\pi) = \{z \in \mathcal{A} : \pi_z \neq 0\}$, i.e., the set of codewords with nonzero coefficients in the inequality. Let $B \subseteq \mathcal{A}$ be a box, and let $\pi x \leq \pi_0$ be a valid inequality for FPP. We say that $\pi x \leq \pi_0$ is *contained in the box structure* $N[B] = B \cup N(B)$ if $\mathrm{supp}(\pi) \subseteq N[B]$ and, for every $x \in B$, all the codewords in $N_B(x)$ have the same coefficient in π.

Theorem 3.5 *The only facet-inducing inequalities of* FPP *contained in a box structure are the point inequalities (1), the box inequalities (2), and the reinforced box inequalities (3).*

4 Diamond inequalities

Let $B = \{a, b, d, c\} \subseteq \mathcal{A}$ be a box. Let $N(a) \cap N(c) = \{e\}$, $N(c) \cap N(b) = \{f\}$, $N(a) \cap N(d) = \{g\}$, and $N(b) \cap N(d) = \{h\}$. We call the set $D = B \cup \{e, f, g, h\}$ a *diamond* of $\mathcal{A}$. For $x \in D$, we define $N_D(x) = N(x) \backslash D$. Note that $N_D(x)$ and $N_D(y)$ are disjoint if $x \neq y$. The codewords in B are called the *inner* codewords of the diamond, and the codewords in $D \backslash B$ are called the *outer* codewords of the diamond.

The diamond structure generates two further classes of facet-inducing inequalities for FPP, based on the codewords required for covering different subsets of a diamond. The first family is based on the covering of two inner codewords and one outer codeword, and the second family is based on the covering of one inner codeword and two outer codewords.

Definition 4.1 [inner diamond inequalities] If $D = \{a, \ldots, h\}$ is a diamond, we define

$$\sum_{z \in D} x_z + (x_{N_D(b)} + x_{N_D(d)} + x_{N_D(e)}) \ \geq \ 2 \tag{4}$$

to be the *inner diamond inequality* associated with D.

Theorem 4.2 *The inner diamond inequalities (4) are valid and facet-inducing for* FPP.

Definition 4.3 [outer diamond inequalities] If $D = \{a, \dots, h\}$ is a diamond, we define

$$\sum_{z \in D} x_z + (x_{N_D(a)} + x_{N_D(f)} + x_{N_D(h)}) \;\geq\; 2 \tag{5}$$

to be the *outer diamond inequality* associated with D.

Theorem 4.4 *The outer diamond inequalities (5) are valid and facet-inducing for* Fpp.

Let $B = \{a, b, c, d\}$ be a box and let $D = B \cup \{e, f, g, h\}$ be a diamond of $\mathcal{A}$. If $\pi x \leq \pi_0$ is a valid inequality for Fpp, we say that the inequality is *contained in the diamond structure* $N[D] = D \cup N(D)$ if $\operatorname{supp}(\pi) \subseteq N[D]$ and, for every $x \in D$, all the codewords in $N_D(x)$ have the same coefficient in π.

Theorem 4.5 *The only facet-inducing inequalities contained in any diamond structure are the point inequalities (1), the box inequalities (2), the reinforced box inequalities (3), the inner diamond inequalities (4), and the outer diamond inequalities (5).*

5 3D-Box inequalities

We define a *3D-Box* to be a set $T = B \cup B'$, where $B = \{a, b, c, d\}$ and $B' = \{a', b', c', d'\}$ are two boxes such that $\operatorname{dist}(a, a') = \operatorname{dist}(b, b') = \operatorname{dist}(c, c') = \operatorname{dist}(d, d') = 1$.

A 3D-Box is given by the eigth codewords differing in three fixed positions. For example, we may take $T = \{xyz00 : x, y, z \in \{0, 1\}\}$. In this case, we have $a = 00000$, $b = 11000$, $c = 10000$, and $d = 01000$ as in the standard box structure and, furthermore, $a' = 00100$, $b' = 11100$, $c' = 10100$, and $d' = 01100$.

For $x \in \{a, b\}$, define $N_T(x) = N(x) \backslash T$ and, for $x \in \{a', b', c, d\}$, define $N_T(x) = N(x) \backslash (T \cup N_T(a) \cup N_T(b))$. Finally, for $x \in \{c', d'\}$, define $N_T(x) = N(x) \backslash (T \cup N_T(a') \cup N_T(b') \cup N_T(c) \cup N_T(d))$. Note that $N_T(x)$ and $N_T(y)$ are disjoint if $x \neq y$.

Definition 5.1 [3D-Box 1-2 inequalities] If $T = \{a, b, c, d\} \cup \{a', b', c', d'\}$ is a 3D-Box, we define

$$\begin{aligned} & 2(x_{a'} + x_{b'} + x_d + x_{d'}) + (x_a + x_b + x_c + x_{c'}) + \\ & + (x_{N_T(a)} + x_{N_T(b)} + x_{N_T(d)}) + (x_{N_T(a')} + x_{N_T(b')} + x_{N_T(c')}) \;\geq\; 3 \end{aligned} \tag{6}$$

to be the *3D-Box 1-2 inequality* associated with T and the 6-cycle $C = \{a, a', c', b', b, d\}$.

Definition 5.2 [3D-Box 2-3 inequalities] If $T = \{a, b, c, d\} \cup \{a', b', c', d'\}$ is a 3D-Box, we define

$$\begin{aligned} & 3(x_{a'} + x_{b'} + x_d + x_{d'}) + 2(x_a + x_b + x_c + x_{c'}) + \\ & + (x_{N_T(a)} + x_{N_T(b)} + x_{N_T(d)}) + (x_{N_T(a')} + x_{N_T(b')} + x_{N_T(c')}) \ \geq \ 5 \end{aligned} \tag{7}$$

to be the *3D-Box 2-3 inequality* associated with T and the 6-cycle $C = \{a, a', c', b', b, d\}$.

Note that the 3D-Box 1-2 inequality (6) and the 3D-Box 2-3 inequality (7) are defined over the same supporting codewords, the only difference between them being the assignment of coefficients 1 and 2 resp. 2 and 3 within the boxes B and B'.

Theorem 5.3 *The 3D-Box 1-2 inequalities (6) and the 3D-Box 2-3 inequalities (7) are valid and facet-inducing for* Fpp.

As in the previous sections, we can define a valid inequality to be *contained in a 3D-Box structure*. Again, it is possible to show that the only facet-inducing valid inequalities contained in any 3D-Box structure are the point inequalities (1), the box and reinforced box inequalities (2) and (3), the diamond inequalities (4) and (5), and the 3D-Box inequalities (6) and (7).

References

[1] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes.* North-Holland, 1997.

[2] E. Balas and S. Ng, *On the set covering polytope I: All the facets with coefficients in* $\{0, 1, 2\}$. Mathematical Programming 43 (1989) 57–69.

[3] H. Kamps and J. van Lint, *The football pool problem for 5 matches.* Journal of Combinatorial Theory 2 (1967) 315–325.

[4] J. van Lint, *Introduction to coding theory.* Springer-Verlag, 1982.

[5] P. Östergard, *A combinatorial proof of the football pool problem for six matches.* Journal of Combinatorial Theory A-76 (1996) 160–163.

[6] M. Sánchez-García, M. Sobrón and B. Vitoriano, *On the set covering polytope: Facets with coefficients in* $\{0, 1, 2, 3\}$. Annals of Operations Research 81 (1998) 343–356.

The polynomial dichotomy for three nonempty part sandwich problems [1]

Rafael B. Teixeira [a,2], Simone Dantas [b,2],
Celina M. H. de Figueiredo [a,2]

[a] *COPPE, Universidade Federal do Rio de Janeiro, Brazil*

[b] *Instituto de Matemática, Universidade Federal Fluminense, Brazil*

Abstract

We classify into polynomial time or *NP*-complete all three nonempty part sandwich problems. This solves the polynomial dichotomy for this class of problems.

Keywords: Graph algorithms, Computational complexity, Structural characterization of graphs

1 Introduction

Consider an undirected, finite, simple graph $G = (V(G), E(G))$ and the problem of finding a partition of $V(G)$ into nonempty subsets satisfying constraints *internal* or *external*. An internal constraint refers to constraints within the parts as to be a clique, or an independent set. An external constraint refers to constraints between different parts as to be completely adjacent or nonadjacent to other parts. The SKEW PARTITION PROBLEM was defined [3] as finding a partition of the vertex set of a given graph into four nonempty parts A, B,

[1] This research was partially supported by CNPq 470420/2004-9.

[2] Email:rafaelbt,sdantas,celina@cos.ufrj.br

C, D such that there are all possible edges between A and B, and no edges between C and D. It has a key role in the proof of the strong perfect graph theorem [4], and it admits a polynomial-time algorithm [11]. The SKEW PARTITION PROBLEM has only external constraints, variations with additional internal constraints ou larger number of parts have been considered [6,8].

Given a graph G and a positive integer k, consider the problem of partitioning the vertex set into at most k parts A_1, A_2, …, A_k, subject to constraints specified by a symmetric $k \times k$ matrix M over $\{0, 1, *\}$ [9] such that for $i \neq j$, if entry $m_{i,j} = 0$ (resp., 1, $*$) then we require 'no edges' (resp., 'all edges', 'no restriction') between a vertex placed in part A_i and a vertex placed in part A_j; if entry $m_{i,i} = 0$ (resp., 1, $*$) then we require A_i to induce a stable set (resp., clique, arbitrary subgraph). An M-partition of graph G is a partition of its vertex set into at most k parts so that all the constraints specified by M are respected. The M-PARTITION PROBLEM asks: "Given a graph G, does G admit an M-partition?". In the LIST M-PARTITION PROBLEM, we are given a graph G, and each vertex v of G has a nonempty list $\mathcal{L}(v) \subseteq \{A_1, A_2, \ldots, A_k\}$. The problem asks: "Does G admit an M-partition in which each vertex v of G is assigned to a part in $\mathcal{L}(v)$?". In particular, we note that if, for all v, $\mathcal{L}(v) = \{A_1, A_2, \ldots, A_k\}$, then we have the M-PARTITION PROBLEM.

Every LIST M-PARTITION PROBLEM with M of dimension at most 4 was classified by the quasi-dichotomy as either solvable in quasi-polynomial time or NP-complete and every LIST M-PARTITION PROBLEM with M of dimension at most 3 was classified as either solvable in polynomial time or NP-complete [9]. Recently, every LIST M-PARTITION PROBLEM with M of dimension 4 was classified as either solvable in polynomial time or NP-complete [1], with the single exception of the STUBBORN PROBLEM and its complement. The H-PARTITION PROBLEM considers [7] a 4×4 matrix M with only $*$s in its main diagonal, it does not impose internal constraints, and requires the four parts of the partition to be nonempty. The SKEW PARTITION PROBLEM is an H-PARTITION PROBLEM.

Graph sandwich problems [12] are generalized recognition problems arising from applications in computational biology. Say that a graph $G^1 = (V, E^1)$ is a *spanning* subgraph of $G^2 = (V, E^2)$ if $E^1 \subseteq E^2$; and that a graph $G = (V, E)$ is a *sandwich* graph for the pair G^1, G^2 if $E^1 \subseteq E \subseteq E^2$. For notational simplicity in the sequel, we let E^3 be the set of all edges in the complete graph with vertex set V which are not in E^2. Thus every sandwich graph for the pair G^1, G^2 satisfies $E^1 \subseteq E$ and $E \cap E^3 = \emptyset$. We call E^1 the *forced edge set*, $E^2 \setminus E^1$ the *optional edge set*, E^3 the *forbidden edge set*. The GRAPH SANDWICH PROBLEM FOR PROPERTY Π asks, given a vertex set V, a forced edge set E^1, and a forbidden edge set E^3, whether there is a graph $G = (V, E)$ such that $E^1 \subseteq E$ and $E \cap E^3 = \emptyset$ that satisfies property Π.

Graph sandwich problems for properties Π related to decompositions arising in perfect graph theory have been considered: homogeneous set [2], join composition [10], (k, l) graphs [5], clique and star cutsets [13]. Homogeneous set, $(2, 1)$ graph, and clique cutset are three dimensional M-PARTITION PROBLEMS, with the additional constraint that the three parts of the partition are required to be nonempty.

In this paper, we consider all GRAPH SANDWICH PROBLEMS corresponding to three dimensional M-PARTITION PROBLEMS, with the additional constraint that the parts of the partition are required to be nonempty. We completely solve the polynomial dichotomy for this class of problems, by classifying each problem into polynomial time or NP-complete.

THREE NONEMPTY PART M-PARTITION SANDWICH PROBLEM (3NPMSP)
Instance: Vertex set V, forced edge set E^1, forbidden edge set E^3.
Question: Is there a graph $G = (V, E)$ such that $E^1 \subseteq E$ and $E \cap E^3 = \emptyset$ that admits a three nonempty part M-partition?

2 Listing the 61 interesting matrices

The complement $\overline{M}_x$ of a matrix M_x is the matrix obtained from M_x by replacing each 1 by 0 and each 0 by 1 (the $*$ constraints remain unchanged). Two matrices M_x, M_ℓ are *isomorphic* if M_ℓ represents the same partition as M_x or $\overline{M}_x$. This means that M_ℓ is obtained from M_x or $\overline{M}_k$ by a permutation of its part names A_i, i.e., its lines and columns.

If all entries of a matrix M are 0 or $*$, then M defines a hereditary property, and the sandwich problem is a recognition problem, for which it is sufficient to test whether G^1 admits a three nonempty part M-partition. If all entries of a matrix M are 1 or $*$, then M defines an ancestral property, and the sandwich problem is a recognition problem, for which it is sufficient to test whether G^2 admits a three nonempty part M-partition. Since all three nonempty part M-partition recognition problems are classified, we focus on *interesting* matrices containing at least one entry 0 and one entry 1.

Figure 1 depicts all, up to isomorphisms, 61 interesting 3×3 matrices M_x, each matrix defines its corresponding decision sandwich problem 3NPM_xSP. The 61 matrices are sorted by increasing number of internal constraints, and then by increasing number of external constraints. In case the matrix contains an internal constraint, we fix $m_{11} = 0$.

$$\begin{array}{cccccccc}
\begin{bmatrix}\ast&0&1\\0&\ast&\ast\\1&\ast&\ast\end{bmatrix} & \begin{bmatrix}\ast&0&0\\0&\ast&1\\0&1&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&\ast\\\ast&\ast&1\\\ast&1&\ast\end{bmatrix} & \begin{bmatrix}0&0&1\\0&\ast&\ast\\1&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&0&\ast\\0&\ast&1\\\ast&1&\ast\end{bmatrix} & \begin{bmatrix}0&1&1\\1&\ast&\ast\\1&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&1&\ast\\1&\ast&0\\\ast&0&\ast\end{bmatrix} & \begin{bmatrix}0&1&\ast\\1&\ast&1\\\ast&1&\ast\end{bmatrix}\\
(1) & (2) & (3) & (4) & (5) & (6) & (7) & (8)\\
\begin{bmatrix}0&0&0\\0&\ast&1\\0&1&\ast\end{bmatrix} & \begin{bmatrix}0&0&1\\0&\ast&0\\1&0&\ast\end{bmatrix} & \begin{bmatrix}0&0&1\\0&\ast&1\\1&1&\ast\end{bmatrix} & \begin{bmatrix}0&1&1\\1&\ast&0\\1&0&\ast\end{bmatrix} & \begin{bmatrix}0&1&1\\1&\ast&1\\1&1&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&\ast\\\ast&0&1\\\ast&1&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&1&\ast\\0&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&1\\\ast&1&\ast\\1&\ast&\ast\end{bmatrix}\\
(9) & (10) & (11) & (12) & (13) & (14) & (15) & (16)\\
\begin{bmatrix}0&0&1\\0&0&\ast\\1&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&1&0\\1&0&\ast\\0&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&1&1\\1&0&\ast\\1&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&0&1\\0&1&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&1\\\ast&0&1\\1&1&\ast\end{bmatrix} & \begin{bmatrix}0&0&0\\0&1&\ast\\0&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&0&1\\0&1&\ast\\1&\ast&\ast\end{bmatrix} & \begin{bmatrix}0&0&\ast\\0&1&0\\\ast&0&\ast\end{bmatrix}\\
(17) & (18) & (19) & (20) & (21) & (22) & (23) & (24)\\
\begin{bmatrix}0&0&\ast\\0&1&1\\\ast&1&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&1&0\\0&0&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&1&1\\0&1&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&1\\\ast&1&0\\1&0&\ast\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&1\\0&1&\ast\end{bmatrix} & \begin{bmatrix}0&0&1\\0&0&1\\1&1&\ast\end{bmatrix} & \begin{bmatrix}0&1&0\\1&0&0\\0&0&\ast\end{bmatrix} & \begin{bmatrix}0&1&0\\1&0&1\\0&1&\ast\end{bmatrix}\\
(25) & (26) & (27) & (28) & (29) & (30) & (31) & (32)\\
\begin{bmatrix}0&1&1\\1&0&1\\1&1&\ast\end{bmatrix} & \begin{bmatrix}0&0&0\\0&1&0\\0&0&\ast\end{bmatrix} & \begin{bmatrix}0&0&0\\0&1&1\\0&1&\ast\end{bmatrix} & \begin{bmatrix}0&0&1\\0&1&0\\1&0&\ast\end{bmatrix} & \begin{bmatrix}0&1&0\\1&1&0\\0&0&\ast\end{bmatrix} & \begin{bmatrix}0&\ast&\ast\\\ast&0&\ast\\\ast&\ast&1\end{bmatrix} & \begin{bmatrix}0&1&\ast\\1&0&\ast\\\ast&\ast&0\end{bmatrix} & \begin{bmatrix}0&0&\ast\\0&0&\ast\\\ast&\ast&1\end{bmatrix}\\
(33) & (34) & (35) & (36) & (37) & (38) & (39) & (40)\\
\begin{bmatrix}0&1&\ast\\1&0&\ast\\\ast&\ast&1\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&0&\ast\\0&\ast&1\end{bmatrix} & \begin{bmatrix}0&\ast&\ast\\\ast&0&1\\\ast&1&1\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&0&1\\0&1&0\end{bmatrix} & \begin{bmatrix}0&\ast&1\\\ast&0&1\\1&1&0\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&\ast\\0&\ast&1\end{bmatrix} & \begin{bmatrix}0&0&\ast\\0&0&1\\\ast&1&1\end{bmatrix} & \begin{bmatrix}0&1&0\\1&0&\ast\\0&\ast&1\end{bmatrix}\\
(41) & (42) & (43) & (44) & (45) & (46) & (47) & (48)\\
\begin{bmatrix}0&1&\ast\\1&0&1\\\ast&1&1\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&0&0\\0&0&1\end{bmatrix} & \begin{bmatrix}0&\ast&0\\\ast&0&1\\0&1&1\end{bmatrix} & \begin{bmatrix}0&\ast&1\\\ast&0&1\\1&1&1\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&1\\0&1&0\end{bmatrix} & \begin{bmatrix}0&0&1\\0&0&1\\1&1&0\end{bmatrix} & \begin{bmatrix}0&1&1\\1&0&1\\1&1&0\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}\\
(49) & (50) & (51) & (52) & (53) & (54) & (55) & (56)
\end{array}$$

$$\begin{array}{ccccc}
\begin{bmatrix}0&0&0\\0&0&1\\0&1&1\end{bmatrix} & \begin{bmatrix}0&0&1\\0&0&1\\1&1&1\end{bmatrix} & \begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix} & \begin{bmatrix}0&1&0\\1&0&1\\0&1&1\end{bmatrix} & \begin{bmatrix}0&1&1\\1&0&1\\1&1&1\end{bmatrix}\\
(57) & (58) & (59) & (60) & (61)
\end{array}$$

Fig. 1. All interesting 3×3 matrices up to isomorphisms

3 Tools

For some matrices M_x of Figure 1, the corresponding 3NPM_xSP has already been classified: M_1 corresponds to HOMOGENEOUS SET SANDWICH PROBLEM, proved polynomial [2]; M_3 corresponds to CLIQUE CUTSET SANDWICH PROBLEM, and M_{38} corresponds to $(2,1)$-GRAPH SANDWICH PROBLEM, both proved *NP*-complete [13,5]. The remaining problems are classified by applying the seven tools defined next.

Tool 1(Two part reducible): M_x has two equal lines, which implies that 3NPM_xSP is a two part problem, polynomially reducible to 2-SAT.

Tool 2(Stable Cutset and 3-coloring): Let M_S and M_C be, respectively, the 3×3 matrices of the only *NP*-complete three nonempty part recognition problems: STABLE CUTSET and 3-COLORING. Suppose M_x is obtained from M_S by changing some entries $\ast$ to 1, then a polynomial reduction shows 3NPM_xSP is *NP*-

complete. If G is and instance of STABLE CUTSET, then (V, E^1, E^3) such that $E^1 = E$ and $E^3 = \emptyset$ is the required instance for 3NPM_xSP. An analogous construction holds for M_x obtained from M_C by changing some entries $*$ to 1.

Tool 3(Universal vertex): M_x contains a line i with all entries 0 or with all entries 1. If 3NPM_xSP has a solution, then every vertex placed in part A_i is universal with respect to G^2. Vertices that cannot be placed in A_i must be placed in the two remaining parts, which is decided by 2-SAT.

Tool 4(Disconnected partition): M_x satisfies: for some i, $m_{ij} = m_{ik} = 0$ and $m_{ii} \neq 0$. If 3NPM_xSP has a solution, then G_1 must be disconnected. Solve 3NPM_xSP by considering the connected components of G_1 and solving a polynomial number of 2-part problems.

Tool 5(Homogeneous set): M_x is obtained from M_1 by the addition of a constraint that allows a polynomial solution.

Tool 6(Singleton part): If 3NPM_xSP has a solution, then it has a solution with a singleton part A_i. A polynomial algorithm solves n 2-SAT problems obtained by setting $A_i = \{x\}$, for each $v \in V$.

Tool 7(3-SAT): Each such 3NPM_xSP is proved *NP*-complete by a reduction from 1-in-3 3-SAT (without negative literals), for an example see Figure 2.

Tools	Problems	Class
Two part reducible	6, 30, 40, 54, 56, 58.	Polynomial
Stable cutset, 3-colour	7, 12, 16, 23, 28, 36, 39, 41, 45, 48, 55, 59.	NP-Complete
Universal vertex	9, 22, 29, 34, 35, 46, 52, 53, 57, 61.	Polynomial
Disconnected partition	2, 8, 10, 11, 13, 19, 21, 24, 26, 31, 33, 37, 50.	Polynomial
Homogeneous set	4, 5, 27	Polynomial
Singleton part	15, 17, 18, 25, 32, 44, 47.	Polynomial
3-SAT	14, 20, 42, 43, 49, 51, 60.	NP-Complete

References

[1] K. Cameron, E. M. Eschen, C. T. Hoàng, R. Sritharan. The list partition problem for graphs. In Proc. of the ACM-SIAM Symp. on Disc. Algorithms, 391–399, 2004.

[2] M. Cerioli, H. Everett, C. M. H. de Figueiredo, and S. Klein. The homogeneous set sandwich problem. Inform. Process. Lett., 67 (1998), 31–35.

[3] V. Chvátal. Star-cutsets and perfect graphs. J. Combin. Theory Ser. B, 39 (1985) 189–199.

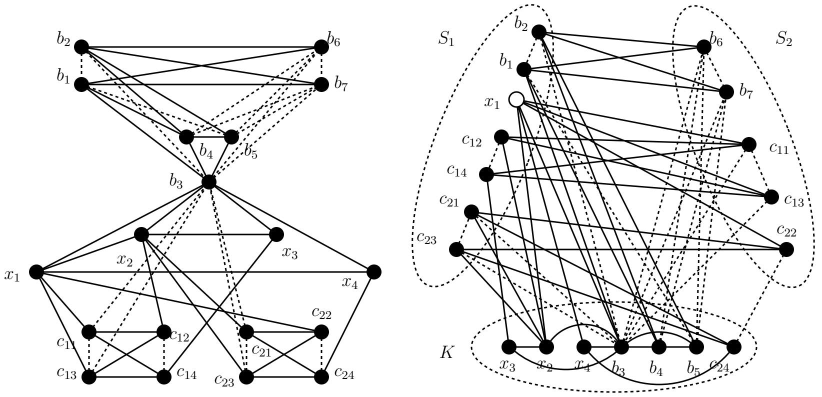


Fig. 2. Instance for 3NPM_{42}SP, obtained from satisfiable instance of 1-in-3 3SAT $\{(x_1, x_2, x_3), (x_2, x_1, x_4)\}$ and respective partition from satisfying truth assignment $x_1 = T, x_2 = F, x_3 = F, x_4 = F$. Solid edges are forced, dashed edges are forbidden, optional edges are omitted

[4] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph Theorem. Ann. of Math., 164 (2006), 51–229.

[5] S. Dantas, L. Faria, and C. M. H. Figueiredo. On decision and optimization (k, l)-graph sandwich problems. Discrete Appl. Math., 143 (2004), 155–165.

[6] S. Dantas, C. M. H. de Figueiredo, S. Gravier, S. Klein, and B. Reed. Stable skew partition problem. Discrete Appl. Math., 143 (2004), 17–22.

[7] S. Dantas, C. M. H. de Figueiredo, S. Gravier, and S. Klein. Finding H-partitions efficiently. Theor. Inform. Appl., 39 (2005), 133–144.

[8] S. Dantas, C. M. H. de Figueiredo, S. Gravier, and Klein. Extended skew partition problem. Discrete Math., 306 (2006), 2438–2449.

[9] T. Feder, P. Hell, S. Klein, and R. Motwani. List partitions. SIAM J. Discrete Math., 16 (2003), 449–478.

[10] C. M. H. de Figueiredo, S. Klein and K. Vusković. The Graph Sandwich Problem for 1-Join Composition is *NP*-complete. Discrete Appl. Math., 121 (2002), 73–82.

[11] C. M. H. de Figueiredo, S. Klein, Y. Kohayakawa, and B. Reed, Finding Skew Partitions Efficiently. J. Algorithms, 37 (2000), 449–473.

[12] M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. J. Algorithms, 19 (1995), 449–473.

[13] R. B. Teixeira and C. M. H. Figueiredo. The sandwich problem for cutsets: Clique cutset, k-star cutset. Discrete Appl. Math., 154 (2006), 1791–1798.

A New Lower Bound for the Minimum Linear Arrangement of a Graph

André R.S. Amaral [a] Alberto Caprara [b] Adam N. Letchford [c]
Juan-José Salazar-Gonzalez [d]

[a] *Departamento de Informática, Universidade Federal do Espírito Santo, Vitoria, ES 29060-970, Brazil. E-mail:* amaral@inf.ufes.br

[b] *DEIS, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy. E-mail:* acaprara@deis.unibo.it

[c] *Department of Management Science, Lancaster University, Lancaster LA1 4YX, Great Britain. E-mail:* a.n.letchford@lancaster.ac.uk

[d] *DEIOC, Universidad de La Laguna, 38271 La Laguna, Tenerife, Spain. E-mail:* jjsalaza@ull.es

Abstract

Given a graph $G = (V, E)$ on n vertices, the *Minimum Linear Arrangement Problem* (MinLA) calls for a one-to-one function $\psi : V \to \{1, \ldots, n\}$ which minimizes $\sum_{\{i,j\} \in E} |\psi(i) - \psi(j)|$. MinLA is strongly $\mathcal{NP}$-hard and very difficult to solve to optimality in practice. One of the reasons for this difficulty is the lack of good lower bounds. In this paper, we take a polyhedral approach to MinLA. We associate an integer polyhedron with each graph G, and derive many classes of valid linear inequalities. It is shown that a cutting plane algorithm based on these inequalities can yield competitive lower bounds in a reasonable amount of time. A key to the success of our approach is that our linear programs contain only $|E|$ variables. We conclude showing computational results on benchmark graphs from literature.

Keywords: linear arrangement problem, polyhedral combinatorics, cutting planes

1 Introduction

Given a graph $G = (V, E)$, with $V = \{1, \ldots, n\}$, an *arrangement* (also called a *permutation*, *labelling*, *ordering* or *layout*) is a one-to-one function $\psi : V \to V$. If we view ψ as a placing of the vertices on a line segment, the quantity $|\psi(i) - \psi(j)|$ corresponds to the Euclidean distance between vertices i and j. Several important combinatorial optimization problems, collectively known as *graph layout problems*, call for an arrangement minimizing a function of these distances (see the survey [3]). Here, we are concerned with the *Minimum Linear Arrangement Problem* (MinLA), in which the objective is to minimize $\sum_{\{i,j\} \in E} |\psi(i) - \psi(j)|$, the sum of the distances.

MinLA was originally proposed in [7]. It was proven to be strongly $\mathcal{NP}$-hard and this was later shown to hold even when G is bipartite. For general graphs, the fastest known exact algorithm is based on dynamic programming and runs in $\mathcal{O}(2^n|E|)$ time. However, MinLA is known to be solvable in polynomial time on trees, outerplanar graphs and certain Halin graphs. Indeed, for some restricted classes of graphs, optimal layouts are known explicitly [8].

Recently, some progress has been made on the approximability of MinLA. Approximation algorithms with performance guarantee $\mathcal{O}(\log n)$ were introduced in [2,11]. Very recently, an $\mathcal{O}(\sqrt{\log n} \log \log n)$ approximation algorithm was found [6]. It has been conjectured, however, that MinLA cannot be approximated to within a constant factor in polynomial time.

Some heuristics are known which give good performance in practice, see for example [9,10,12,13]. On the other hand, MinLA is very difficult to solve to proven optimality in practice and there are no good exact algorithms available at present. The main reason for this difficulty appears to be the lack of strong and efficiently computable *lower bounds*.

In [5], an interesting new lower bound was proposed, based on solving a linear program (LP) with $|E|$ variables and an exponential number of constraints. However, to our knowledge, the bound was never tested computationally. In the extended version of this paper, we explore and significantly extend the approach in [5]. We derive several new families of valid inequalities in this extended abstract, while the extended paper presents a cutting plane algorithm based on these inequalities and gives extensive computational results. The lower bounds produced by the algorithm are much stronger than other known bounds.

We do not present the proofs of the theorems in this extended abstract. Also the sections with other inequalities, separation procedures and computational results have been removed due to the limit in the number of pages.

2 The Polyhedral Study

In this section, we study certain integer polyhedra associated with MinLA.

2.1 Definitions and elementary results

Our goal is to work with only m distance variables, d_e for each $e \in E$. At first sight, it seems natural to work with the projection of the MinLA polytope onto the subspace defined by the edges in E, i.e., with the following integer polytope:

$$\mathcal{P}(G) := \operatorname{conv}\left\{d \in \mathbb{Z}_+^m : \exists \psi : \, d_{ij} = |\psi(i) - \psi(j)| \; (\{i,j\} \in E)\right\}.$$

However, we have found that $\mathcal{P}(G)$ has an extremely complex structure that makes it hard to study. In fact, even determining the dimension of $\mathcal{P}(G)$ seems difficult.

We have found it more helpful to study the *dominant* of $\mathcal{P}(G)$, which is the Minkowski sum of $P(G)$ and the non-negative orthant $\mathbb{R}_+^m$. That is:

$$\text{DOM}(G) := \left\{d \in \mathbb{R}_+^m : d \geq d' \text{ for some } d' \in \mathcal{P}(G)\right\}.$$

Since the objective function in MinLA is non-negative, optimising over DOM(G) is equivalent to optimising over $\mathcal{P}(G)$. However, DOM(G) is much easier to work with. Indeed, we have the following three elementary results, the proofs of which are more or less immediate.

Proposition 2.1 *DOM(G) is a full-dimensional, unbounded polyhedron. It is of* blocking *type, i.e., its recession cone is the non-negative orthant.*

Proposition 2.2 *If the inequality $\alpha^T d \geq \beta$ is valid for DOM(G), then α and β are non-negative. Moreover, if it is* supporting *(i.e., there exists at least one vertex d^* of DOM(G) such that $\alpha^T d^* = \beta$), then β is positive.*

Proposition 2.3 *Let $G = (V,E)$ be a general graph, let $E' \subset E$ be a subset of the edges and let $G' = (V',E')$ be the subgraph of G induced by E'. The inequality $\sum_{e \in E'} \alpha_e d_e \geq \beta$ is valid (or supporting, or facet-inducing) for DOM(G') if and only if it is valid (respectively, supporting, facet-inducing) for DOM(G).*

When $G = K_n$, the relationship between $P(G)$ and DOM(G) is clear:

Proposition 2.4 *$P(K_n)$ is the unique bounded facet of DOM(K_n), induced by the equation $\sum_{1 \leq i < j \leq n} d_{ij} = \binom{n+1}{3}$.*

From now on we concentrate on DOM(G). In the following subsections, we present various valid and facet-inducing inequalities. Our main method for deriving valid inequalities for DOM(G) is simply to look at classes of graphs for which there is an explicit formula for the cost of the optimal arrangement. If $G' = (V', E')$ is such a graph, and β is the cost of the optimal arrangement, then obviously

$$\sum_{\{i,j\} \in E} d_{ij} \geq \beta$$

is valid for DOM(G), for any graph G containing G' as a subgraph.

2.2 Clique inequalities

MinLA is trivial when $G = K_n$, since any arrangement satisfies the equation $d(E) = \binom{n+1}{3}$. This leads immediately to the following *clique* inequalities:

Theorem 2.5 *For any $n \geq 2$, and for any $S \subseteq V$ inducing a clique in G, the* clique *inequality*

$$\sum_{\{i,j\} \subset S} d_{ij} \geq \binom{|S|+1}{3}$$

is valid and facet-inducing for DOM(G).

Note that the clique inequalities with $|S| = 2$ are the trivial lower bounds $d_e \geq 1$.

2.3 Star inequalities

MinLA is also trivial when G is a *star* (i.e., a graph in which all edges are incident on a common vertex). In this case, the optimal MinLA solution has cost $\lfloor n^2/4 \rfloor$. This leads to the following *star* inequalities:

Lemma 2.6 *For any $i \in V$ and any $S \subset n(i)$, the* star *inequality*

$$\sum_{j \in S} d_{ij} \geq \lfloor (|S|+1)^2/4 \rfloor$$

is valid for DOM(G).

Star inequalities were shown to be valid for $\mathcal{P}_n$, where it was also noted that they do not in general induce facets of $\mathcal{P}_n$. In the case of DOM(G), however, we have:

Theorem 2.7 *Star inequalities induce facets of DOM(G) if and only if $|S| \neq 2$.*

2.4 Circuit inequalities

MinLA is also trivial when G is a *circuit* (i.e., a simple cycle). Clearly, if G is a circuit on n vertices, then the optimal MinLA solution has cost $2n - 2$. This leads to the following *circuit* inequalities:

Lemma 2.8 *For any $C \subset E$ inducing a circuit in G, the* circuit *inequality*

$$d(C) \geq 2|C| - 2$$

is valid for DOM(G).

Note that the circuit inequality with $|C| = 3$ is a clique inequality.

Theorem 2.9 *Circuit inequalities induce facets of DOM(G).*

2.5 Bipartite and double star inequalities

A more general class of graphs for which MinLA is polynomially solvable (though by no means trivial) is that of the *complete bipartite graphs*. We denote the complete bipartite graph by $K_{p,q}$ and assume without loss of generality that $p \leq q$. In [8] it is shown that the optimal solution to MinLA for $K_{p,q}$ is $p(3q^2+6pq-p^2+4)/12$ (when $p+q$ is even) and $p(3q^2+6pq-p^2+1)/12$ (when $p+q$ is odd). For simplicity of notation we denote this optimal value by $R(p,q)$.

The above result leads immediately to the following *bipartite* inequalities:

Lemma 2.10 *Let $F \subset E$ induce the complete bipartite subgraph $K_{p,q}$ in G, with $p \leq q$. Then the* bipartite *inequality $d(F) \geq R(p,q)$ is valid for DOM(G).*

Note that, when $p = 1$, the bipartite inequalities reduce to star inequalities, and are therefore facet-inducing for $q \neq 2$. The following theorem deals with the case $p \geq 2$.

Theorem 2.11 *Bipartite inequalities induce facets of DOM(G) if $3 \leq p \leq q$, and also if $p = 2$ and q is even.*

In the remaining case, in which $p = 2$ and q is odd, the bipartite inequalities do not induce facets. To see this, we need the following result.

Theorem 2.12 *Let $\{i,j\} \subset V$ and $T \subset V \setminus \{i,j\}$ be such that i and j are adjacent to every vertex in T, and such that $q = |T|$ is odd and at least 3.*

Then the double star *inequality*

$$\sum_{k \in T} (2d_{ik} + d_{jk}) \geq 3(q^2 + 4q - 1)/4$$

is valid and facet-inducing for DOM(G).

References

[1] D. Adolphson & T.C. Hu (1973) Optimal linear ordering. *SIAM J. Appl. Math.*, 25, 403–423.

[2] C. Bornstein & S. Vempala (2004) Flow metrics. *Theor. Comp. Sci.*, 321, 13–24.

[3] J. Díaz, J. Petit & M. Serna (2002) A survey of graph layout problems. *ACM Computing surveys*, 34, 313–356.

[4] T. Easton, S.B. Horton, R.G. Parker (1996) A solvable case of the optimal linear arrangement problem on Halin graphs. *Congr. Numer.*, 119, 3–17.

[5] G. Even, J. Naor, S. Rao & B. Schieber (2000) Divide-and-conquer approximation algorithms via spreading metrics. *J. of the ACM*, 47, 585–616.

[6] U. Feige & J.R. Lee (2007) An improved approximation ratio for the minimum linear arrangement problem. *Inf. Proc. Lett.*, 101, 26–29.

[7] L.H. Harper (1964) Optimal assignment of numbers to vertices. *SIAM Journal*, 12, 131–135.

[8] M. Juvan and B. Mohar (1992) Optimal linear labellings and eigenvalues of graphs. *Discr. Appl. Math.*, 36, 153–168.

[9] J. Petit (2003) Combining spectral sequencing and parallel simulated annealing for the MinLA problem. *Parallel Processing Letters*, 13, 77–91.

[10] J. Petit (2003b) Experiments on the linear arrangement problem. *Journal of Experimental Algorithmics*, 8, article 2.3.

[11] S. Rao & A.W. Richa (2005) New approximation techniques for some linear ordering problems. *SIAM J. on Computing*, 34, 388–404.

[12] E. Rodriguez-Tello, J.-K. Hao & J. Torres-Jimenez (2007) An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *Comp. & Oper. Res.*, to appear.

[13] I. Safro, D. Ron & A. Brandt (2006) Graph minimum linear arrangement by multilevel weighted edge contractions. *J. of Alg.*, 60, 24–41.

Constraint Programming for the Diameter Constrained Minimum Spanning Tree Problem

Thiago F. Noronha, Andréa C. Santos

Department of Computer Science, Catholic University of Rio de Janeiro
Rua Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil

Celso C. Ribeiro

Department of Computer Science, Universidade Federal Fluminense
Rua Passo da Pátria 156, Niterói, RJ 24210-240, Brazil
{*tfn,cynthia,celso*}*@inf.puc-rio.br*

Abstract

We propose a new formulation for the Diameter Constrained Minimum Spanning Tree Problem using constraint programming. Computational results have shown that this formulation combined with an appropriate search procedure solves larger instances and is faster than the other approaches in the literature.

1 Introduction

Given an undirected connected graph $G = (V, E)$ with a set V of vertices, a set E of edges, and costs c_{ij} associated to every edge $[i, j] \in E$, with $i < j$, the *Diameter Minimum Spanning Tree Problem* (DCMST) consists in finding a minimum spanning tree $T = (V, E')$, with $E' \subseteq E$, where the *diameter* required does not exceed a given positive integer value D, where $2 \leq D \leq |V| - 1$. The *diameter* of a tree T is equal to the number of edges in the longest path between any two nodes $i, j \in V$ in T. This problem is NP-hard when $D \geq 4$. DCMST applications appear in telecommunications, data compression and distributed mutual exclusion in parallel computing.

Some mixed integer programming (MIP) formulations for DCMST implicitly use a property [5] that ensures that a *central vertex* $c \in V$ exists in any feasible tree T when D is even, such that no vertex is more than $D/2$ edges away from c. If D is odd, a *central edge* $e = [a, b] \in E$ exists in T, such that no vertex is more than $(D-1)/2$ edges away from the closest extremity of e.

The first DCMST formulations, using single commodity flows, were presented in [2]. Improved formulations with valid inequalities and a lifting procedure are found in [1,6]. An alternative formulation for the odd D case was also proposed in [6]. Multicommodity flow formulations with tighter linear programming relaxations are presented in [3]. However, they require more memory and computation time to solve the linear relaxation. Formulations strengthened with valid inequalities were proposed in [4]. A comparison with other formulations appears in [4]. The computational results showed that no approach dominates any other. The formulations using single commodity flow produce weak linear programming relaxations, especially for small diameters. On the other hand, the multicommodity flow formulations give tighter lower bounds, but require much more memory and time to solve the linear relaxation, because of their large number of variables and constraints.

Constraint Programming (CP) is a programming paradigm for formulating and solving combinatorial problems. Instead of using only the linear relaxation for pruning the search tree, it uses a variety of bounding techniques based on constraint propagation, which consists in operating with the constraints to generate new constraints that reduce the domain of some variables and, consequently, the size of the search space.

In the next section, we propose a new approach based on constraint programming for solving DCMST. It is capable to handle both the odd and even diameter cases in the same formulation, contrary to most approaches in the literature. Computational experiments are reported in Section 3. Concluding remarks are drawn in Section 4.

2 Constraint programming formulation

We propose a new approach to DCMST based on constraint programming that tackles both the odd and even D cases. It was motivated by the fact that some MIP formulations have weak lower bounds, while others require a huge amount of memory because of the large number of variables and constraints.

CP reduces these drawbacks by allowing concise formulations with a small number of variables and by using constraint propagation for pruning the search space, instead of the linear relaxation. Since it is based on a backtracking procedure, CP requires less memory than branch-and-bound algorithms.

The directed graph $G' = (V', A')$ is obtained from the original undirected graph $G = (V, E)$ as follows. Let r be an artificial vertex and $V' = V \cup \{r\}$. For every edge $[i, j] \in E$, with $i < j$, there exist two arcs (i, j) and $(j, i) \in A$ with costs $c_{ij} = c_{ji}$. Then, $A' = A \cup \{(r, 1), \ldots, (r, |V|)\}$, with costs $c_{ri} = 0$ for every $i \in V$. For any $i \in V$, we define $backwardStar[i]$ as the set of all vertices $j \in V'$ such that $(j, i) \in A'$ and $forwardStar[i]$ as the set of all vertices $j \in V$ such that $(i, j) \in A'$. Let $L = \lfloor D/2 \rfloor$. The number of edges in the path from the artificial vertex r to $i \in V$ is said to be the *height* of vertex i.

We give below the formulation of DCMST using the ILOG OPL language. Variables a and b denote the central vertices of the spanning tree. When D is odd, $[a, b]$ denotes the central edge of the spanning tree. In the even D case, $a = b$ denotes the central vertex. Variable $y_i \in V'$ denotes de parent of vertex $i \in V$. Furthermore, variable $u_i \in \{0, ..., L+1\}$ represents the height of vertex $i \in V'$. Sets V and V' are designated by `nodes` and `nodes_p`, respectively. The costs c_{ij} are denoted by `cost[i,j]`. In case $i = j$, then `cost[i,j]` returns zero:

```
var nodes a;
var nodes b;
var nodes_p y[nodes];
var int u[nodes_p] in 0..L+1;
minimize
(1)    sum(i in nodes) cost[i,y[i]] + cost[a,b]
subject to {
(2)    sum(i in nodes) (y[i]=r) = 1 + (D mod 2);
(3)    forall(i in nodes) u[i]=u[y[i]]+1;
(4)    forall(i in nodes) y[i] in backwardStar[i];
(5)    if D mod 2 = 1 then a<b else a=b endif;
(6)    y[a]=y[b]=r;
};
```

The objective function is handled by (1). Constraint (2) ensures that the artificial vertex r is connected to two vertices (i.e., the extremities of the central edge) when D is odd, or to exactly one vertex (i.e., the central vertex) when D is even. Constraints (3) establish that the height of every vertex $i \in V$ in the tree is equal to one plus the height of its parent. Constraints (4) ensure that there is an edge $e \in E$ connecting every vertex i with its parent. If D is even Constraint (5) establishes that a is equal b, otherwise it ensures that a is smaller (different) than b. Constraint (6) establishes that vertex r is the parent of vertices a and b.

Solving a combinatorial optimization problem by constraint programming involves two steps: generating the set of constraints that must be satisfied and describing how to search for solutions. The above formulation gives the set of constraints that must be satisfied, i.e., it describes the search space. For sake of conciseness, the search procedure is omitted.

3 Computational results

The computational experiments were carried out on a Pentium 4 with 3.0 GHz clock and 1Mb of RAM memory, using OPL Studio 3.7.1 as the constraint programming solver. We compared our results with the best over all those obtained by any of the MIP approaches in [4,6], according with the computation times presented in Table 1 of [4] for the same 29 instances used in our study: 18 on complete graphs and 11 on sparse graph, with diameters equal to 4, 5, 6, 7, 9, and 10.

Numerical results are presented on Table 1. For each instance, the first three columns give its number of vertices, its number of edges, and its maximum diameter, respectively. The next three columns give statistics for the CP formulation: the number of nodes visited in the search tree, the amount of memory (in bytes) used by the algorithm, and the computation time in seconds to prove optimality. The next two columns give the time in seconds (on a Pentium 4 with 2.8 GHz clock and 2Mb of RAM memory, using CPLEX 8.1 as MIP solver) to prove optimality by the best MIP formulation in [4,6] and the corresponding algorithm version. The best MIP formulations of [4] and [6] are denoted by `ILP` and `Santos`, respectively (the symbol '+' denotes the use of connect cuts, while an '*' indicates the use of cycle elimination cuts [4]). The last column shows the ratios between the computation times to find the optimal solution with constraint programming and the best MIP approach.

CP performed better than the best MIP approach on all but four out of the 29 test instances. On average, the CP approach run on 45% of the time of the best MIP variant. The computation times were particularly remarkable for the instances with odd diameter: for these instances, the time needed by the CP algorithm to prove optimality was only 23% of that taken by the best MIP variant, on average. The larger is the time to prove optimality, the better is the performance of the CP algorithm when compared with all MIP variants.

				CP		Best MIP		
$\|V\|$	$\|E\|$	D	nodes	memory	time (s)	time (s)	version	CP/MIP
15	105	4	1,044	463,780	0.08	0.7	ILP	11.43%
15	105	5	2,850	463,780	0.22	3.0	ILP	7.33%
15	105	6	6,960	479,800	0.28	8.1	ILP	3.46%
15	105	7	8,240	527,860	0.38	20.0	ILP+*	1.90%
15	105	9	11,743	527,860	0.47	6.2	Santos+*	7.58%
15	105	10	11,830	511,840	0.41	1.0	Santos+	41.00%
20	190	4	3,143	607,960	0.2	2.5	ILP	8.00%
20	190	5	18,283	640,000	1.06	8.1	ILP	13.09%
20	190	6	35,383	672,040	2.03	95.0	ILP	2.14%
20	190	7	19,142	688,060	0.97	5.5	ILP*	17.64%
20	190	9	119,906	752,140	5.01	66.7	ILP+*	7.51%
20	190	10	151,969	672,040	6.08	29.7	Santos+*	20.47%
25	300	4	28,842	800,200	1.48	12.0	ILP	12.33%
25	300	5	37,608	864,280	2.83	64.3	ILP+*	4.40%
25	300	6	534,222	864,280	39.14	26.4	ILP	148.26%
25	300	7	812,957	979,424	56.06	770.5	ILP	7.28%
25	300	9	2,655,810	1,043,504	114.14	246.0	Santos+	46.40%
25	300	10	1,126,130	944,380	55.47	254.8	Santos+	21.77%
20	50	4	389	466,784	0.05	0.2	ILP	25.00%
20	50	5	3,611	466,784	0.17	1.0	ILP	17.00%
20	50	6	2,678	498,824	0.13	0.8	Santos+	16.25%
20	50	7	1,975	498,824	0.14	0.8	Santos+	17.50%
20	50	9	13,040	495,820	0.45	0.7	Santos+	64.29%
20	50	10	17,937	514,844	0.64	0.2	Santos+	320.00%
40	100	4	130,480	911,340	5.44	1.9	ILP	286.32%
40	100	5	161,961	927,360	7.31	6.4	ILP	114.22%
40	100	6	91,022	943,380	4.72	13.2	ILP+	35.76%
40	100	7	778,699	975,420	34.38	212.4	ILP	16.19%
40	100	9	769,161	1,007,460	40.16	979.8	ILP*	4.10%
							Average:	44.78 %

Table 1
Numerical results.

For dense graphs, the CP algorithm run on average in 21% of the time of the best MIP variant: the search strategy implemented within the CP algorithm significantly reduces the size of the search space, when the number of nodes is much smaller then the number of edges.

The algorithms in [4] are more appropriate to small diameter instances, while those in [6] perform better on large diameter instances. However, the CP approach surpassed both MIP approaches on small and large diameter instances. Furthermore, no instance required more than 1 Mbyte of RAM memory to be solved, because the search tree is explored by a depth first search algorithm and no node is stored to be further explored.

4 Conclusions

We proposed a new approach based on constraint programming to solve the degree constrained minimum spanning tree problem. Constraint programming was capable to overcome the main drawbacks of MIP: first, by using more concise formulations with a smaller number of variables; and second, by using constraint propagation for pruning the search space, instead of the bound provided by the linear relaxation. One single algorithm was capable to handle both the odd and even diameter instances.

Constraint programming obtained better results (i.e., smaller computation times to find exact optimal solutions and to prove their optimality) than all MIP approaches for most (25 out of the 29) of the test instances. On the average, the constraint programming computation times were only 45% of those observed with the MIP approach. The advantage of the CP approach was even stronger for the instances with odd diameter and for those on dense graphs, for which the previous ratio was equal to 23% and 21%, respectively.

References

[1] Achuthan, N., L. Caccetta, P. Caccetta and J. Geelen, *Computational methods for the diameter restricted minimum weight spanning tree problem*, Australasian Journal of Combinatorics **10** (1994), pp. 51–71.

[2] Achuthan, N., L. Caccetta, P. Caccetta and J. F. Geelen, *Algorithms for the minimum weight spanning tree with bounded diameter problem*, in: K. Phua, C. Wand, W. Yeong, T. Leong, H. Loh, K. Tan and F. Chou, editors, *Optimization Techniques and Applications*, World Scientific, Singapore, 1992 pp. 297–304.

[3] Gouveia, L. and T. Magnanti, *Network flow models for designing diameter-constrained minimum-spanning and Steiner trees*, Networks **41** (2003), pp. 159–173.

[4] Gruber, M. and G. Raidl, *A new 0-1 ILP approach for the bounded diameter minimum spanning tree problem*, in: L. Gouveia and C. Mourao, editors, *Proceedings of the 2nd International Network Optimization Conference*, Lisbon, 2005, pp. 178–185.

[5] Handler, G., *Minimax location of a facility in an undirected graph*, Transportation Science **7** (1978), pp. 287–293.

[6] Santos, A., A. Lucena and C. Ribeiro, *Solving diameter constrained minimum spanning tree problem in dense graphs*, Lecture Notes in Computer Science **3059** (2004), pp. 458–467.

Finding Folkman Numbers via MAX CUT Problem

Andrzej Dudek[1] and Vojtěch Rödl[2]

Department of Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA

Abstract

In this note we report on our recent work, still in progress, regarding Folkman numbers. Let $f(2,3,4)$ denote the smallest integer n such that there exists a K_4–free graph of order n having that property that any 2–coloring of its edges yields at least one monochromatic triangle. It is well–known that such a number must exist [4,10]. For almost twenty years the best known upper bound, given by Spencer, was $f(2,3,4) < 3 \cdot 10^9$ [13]. Recently, the authors and Lu showed that $f(2,3,4) < 130\,000$ [2] and $f(2,3,4) < 10\,000$ [9]. However, it is commonly believed that, in fact, $f(2,3,4) < 100$. All previous bounds are based on an idea of Goodman [6]. It seems that such methods will not yield substantial further improvement. In this note we will generalize this idea by giving a necessary and sufficient condition for a graph G to yield a monochromatic triangle for every edge coloring. In particular, for any graph G we construct a graph H such that G is Folkman if and only if the value of the maximum cut of H is less than twice the number of triangles in G. We believe this technique may be used to find a new upper bound on $f(2,3,4)$.

Keywords: graph coloring, Folkman numbers, MAX CUT problem

[1] Email: `adudek@emory.edu`
[2] Email: `rodl@mathcs.emory.edu`

1 Introduction

Let r, k, l be positive integers with $k < l$, and let $\mathcal{F}(r, k, l)$ be a family of K_l–free graphs having the following property that if $G \in \mathcal{F}(r, k, l)$, then every r–coloring of edges of G must yield at least one monochromatic copy of K_k. J. Folkman showed in [4] that $\mathcal{F}(2, k, l) \neq \emptyset$. The general case, i.e. $\mathcal{F}(r, k, l) \neq \emptyset$, $r \geq 2$, was settled by J. Nešetřil and the second author in [10]. Let $f(r, k, l) = \min_{G \in \mathcal{F}(r,k,l)} |V(G)|$. The problem of determining the numbers $f(r, k, l)$ in general includes the classical Ramsey numbers and thus is not easy. In this note we focus to the case where $r = 2$ and $k = 3$. We will write $G \rightarrow (\triangle)$ and say that G arrows a triangle if every 2–coloring of G yields a monochromatic triangle. Since the Ramsey number $R(3, 3) = 6$ clearly $f(2, 3, l) = 6$, for $l > 6$. The value of $f(2, 3, 6) = 8$ was determined by R. Graham in [7], and $f(2, 3, 5) = 15$ by K. Piwakowski, S. Radziszowski and S. Urbański in [11]. In the remaining case, the upper bounds on $f(2, 3, 4)$ obtained from [4] and [10] are extremely large (iterated tower function). Consequently, in 1975, P. Erdös [3] offered \$100 for proving or disproving that $f(2, 3, 4) < 10^{10}$. Based on the idea of Goodman of counting triangles in a graph and in its complement [6] applied to random graphs P. Frankl and the second author came relatively close to the desired bound showing in [5] that $f(2, 3, 4) < 10^{12}$. Subsequently, J. Spencer in [13] refined this argument and proved $f(2, 3, 4) < 3 \cdot 10^9$ giving a positive answer to the question of Erdös [3]. Subsequently, F. Chung and R. Graham in [1] conjectured that $f(2, 3, 4) < 10^6$ and offered \$100 for a proof of disproof. Recently, L. Lu and independently the authors gave the computer assisted proof of $f(2, 3, 4) < 10\,000$ [9] and of $f(2, 3, 4) < 130\,000$ [2], respectively. Similarly as in [5] and [13] the proofs from [2] and [9] are based on the modification of the idea from Goodman's paper [6]. The idea explores the local property of every vertex neighborhood in a graph (See Corollary 2.2). While this property easily yields that a graph contains a monochromatic triangle in every edge coloring, it seems to be stronger than needed. We believe that this method may not yield substantial further improvement without additional modifications. We will give a necessary and sufficient condition for a graph G to yield a monochromatic triangle for every edge coloring. More precisely, for every graph G we will construct a weighted graph H such that G arrows a triangle if and only if the corresponding value to the maximum cut of H is less than twice number of triangles in G.

2 Counting blue and red triangles

In order to establish a necessary and sufficient condition for a graph G to yield a monochromatic triangle for every edge coloring, we will use a modification of an idea of [6]. For any blue–red coloring of G let $T_{BR}(v)$, $T_{BB}(v)$ and $T_{RR}(v)$ count the number of triangles containing vertex v, for which two edges incident to v are colored blue–red, blue–blue and red–red, respectively. The sum $\sum_{v\in V(G)} T_{BR}(v)$ counts 2 times the number of nonmonochromatic triangles. This is because each such triangle is counted once for two different vertices. On the other hand, the sum $\sum_{v\in V(G)} \big(T_{BB}(v)+T_{RR}(v)\big)$ counts 3 times the number of monochromatic triangles and once the number of nonmonochromatic triangles. Consequently, $G \to (\triangle)$ if and only if for every edge coloring of G the following holds

$$\sum_{v\in V(G)} T_{BR}(v) < 2 \sum_{v\in V(G)} \big(T_{BB}(v)+T_{RR}(v)\big). \tag{1}$$

Denote by $N(v)$ the set of neighbors of a vertex $v \in V$ and let $G[N(v)]$ be a subgraph of G induced on $N(v)$. Also we denote by $M(G)$ the size of the maximum cut of G. One can show using (1) the following Proposition.

Proposition 2.1 *Let $G=(V,E)$ be a graph that satisfies*

$$\sum_{v\in V(G)} M(G[N(v)]) < \frac{2}{3} \sum_{v\in V(G)} \big|E(G[N(v)])\big|. \tag{2}$$

Then, $G \to (\triangle)$.

A special case of Proposition 2.1 was used to determine the upper bounds on $f(2,3,4)$ in [2,5,9,13].

Corollary 2.2 (Frankl and Rödl [5]; Spencer [13]) *Let $G=(V,E)$ be a graph which satisfies*

$$M(G[N(v)]) < \frac{2}{3}\big|E(G[N(v)])\big| \tag{3}$$

for every vertex $v \in V(G)$. Then, $G \to (\triangle)$.

We extend the idea of Goodman [6] and give a necessary and sufficient condition for a graph G to yield a monochromatic triangle for every edge coloring. More precisely, for every graph $G=(V,E)$ with $t_\triangle = t_\triangle(G)$ triangles, we construct a weighted graph H with $2|E|$ vertices such that $G \to (\triangle)$ if and only if the value of the maximum cut of H is less than $2t_\triangle$.

Let G be a graph with the vertex set $V(G) = \{1, 2, ..., n\}$. For every vertex $i \in V(G)$, let G_i be a graph with $V(G_i) = \{(i, j) \mid j \in N(i)\}$ and $E(G_i) = \{\{(i, j), (i, k)\} \mid \{j, k\} \in E(G)\}$. Clearly G_i is isomorphic to the subgraph $G[N(i)]$ of G induced on the neighborhood $N(i)$. Now we define a weighted graph H as follows: $V(H) = \{(i, j) \in V(G) \times V(G) \mid (i, j) \in V(G_i)\}$ and $E(H) = E^+(H) \cup E^-(H)$, where $E^+(H) = \{\{(i, j), (i, k)\} \mid \{j, k\} \in G[N(i)]\}$ and $E^-(H) = \{\{(i, j), (j, i)\} \mid (i, j) \in V(G_i) \text{ and } (j, i) \in V(G_j)\}$. To every edge in E^+ and E^- we assign the weight 1 or $-\infty$, respectively. Clearly $|V(H)| = 2|E(G)|$, $|E^+(H)| = 3t_\triangle(G)$ and $|E^-(H)| = |E(G)|$. Note that the adjacency matrix of H is is a $2|E(G)| \times 2|E(G)|$ matrix with adjacency matrices of $G_i \cong G[N(i)]$ around the diagonal.

We say that H has a *positive cut* if the value of this cut is positive. Let $V(H) = V_1 \cup V_2$ be a positive cut. Since the value of each edge $\{(i, j), (j, i)\} \in E(H)$ is $-\infty$, we infer that $\{(i, j), (j, i)\} \in \binom{V_1}{2} \cup \binom{V_2}{2}$, whenever $\{i, j\} \in E(G)$. Consequently, each blue–red coloring of edges of G defines a bipartition of vertices of H and vice versa. Summarizing, the following holds.

Proposition 2.3 *There is a one to one correspondence between edge colorings of G and positive cuts of H.*

Based on Proposition 2.3 we proved the main result of this note, which we state here without the proof. Now $M(H)$ denotes the value of the maximum cut for a weighted graph H.

Theorem 2.4 *Let G be a graph. Then, $G \to (\triangle)$ if and only if $M(H) < 2t_\triangle(G)$.*

We will show how Theorem 2.4 can be used in the following simple example. Let G_{17} be a graph with the vertex set $V(G_{17}) = \{1, 2, ..., 17\}$ and the edge set defined as follows: $\{i < j\} \in E(G_{17})$ if $j - i$ is a quadratic residue of 17. One can check that G_{17} is K_4–free. Let G_{18} be a graph obtained from G_{17} by adding one additional vertex, say 18, connected to all vertices from $V(G_{17})$. Then, $|V(G_{18})| = 18$, $|E(G_{18})| = 85$, $t_\triangle(G_{18}) = 136$. Clearly G_{18} is K_5–free. In [8], R. Irving proved that $G \to (\triangle)$, thus establishing $f(2, 3, 5) \leq 18$. An alternative (computer assisted) proof of Irving's result is based on Theorem 2.4. Let H be the graph of order 170 from Theorem 2.4 that corresponds to G_{18}. Since $M(H) < 272 = 2 \cdot 136$,[3] Theorem 2.4 yields that $G_{18} \in \mathcal{F}(2, 3, 5)$. Note that we could not apply a simpler condition given by Proposition 2.1. This is because the maximum cut of $G_{18}[N(i)]$, $i = 1, ..., 17$, equals 14 and the maximum cut of $G_{18}[N(18)] \cong G_{17}$ equals 44. Hence, $\sum_{i \in V(G_{18})} M(G_{18}[N(i)]) =$

[3] The authors used Biq Mac solver [12] to compute $M(H)$.

$17 \cdot 14 + 44 = 282$. Also, $\frac{2}{3}\sum_{i \in V(G_{18})} \left|E(G_{18}[N(i)])\right| = \frac{2}{3}(17 \cdot 20 + 68) = 272$. We observe that due to the above equalities condition (2) fails and hence Proposition 2.1 cannot be applied.

3 Acknowledgment

We would like to thank to F. Rendl and A. Wiegele who helped us in using Biq Mac solver [12].

References

[1] F. Chung and R. Graham, *Erdös on graphs. His legacy of unsolved problems*, A K Peters, Wellesley, Massachusetts (1998), 46–47.

[2] A. Dudek and V. Rödl, *On the Folkman number* $f(2, 3, 4)$, submitted.

[3] P. Erdös, *Problems and results in finite and infinite graphs*, Recent advances in graph theory (Proceedings of the Symposium held in Prague), edited by M. Fiedler, Academia Praha (1975), 183–192.

[4] J. Folkman, *Graphs with monochromatic complete subgraphs in every edge coloring*, SIAM J. Appl. Math., 18 (1970), 19–24.

[5] P. Frankl and V. Rödl, *Large triangle–free subgraphs in graphs without* K_4, Graphs and Combinatorics, 2 (1986), 135–144.

[6] A.W. Goodman, *On sets of acquaintances and strangers at any party*, Amer. Math. Mon., 66 (1959), 778–783.

[7] R. Graham, *On edgewise 2–colored graphs with monochromatic triangles and containing no complete hexagon*, J. Comb. Theory, 4, (1968), 300–300.

[8] R. W. Irving, *On a bound of Graham and Spencer for a graph–coloring constant*, J. Comb. Theory Ser. B, 15, (1973), 200–203.

[9] L. Lu, *Explicit construction of small Folkman graphs*, submitted.

[10] J. Nešetřil and V. Rödl, *The Ramsey property for graphs with forbidden complete subgraphs*, J. Comb. Theory Ser. B, 20, (1976), 243–249.

[11] K. Piwakowski, S. Radziszowski and S. Urbański, *Computation of the Folkman number* $F_e(3, 3; 5)$, J. Graph Theory, 32, (1999), 41–49.

[12] F. Rendl, G. Rinaldi and A. Wiegele, *Biq Mac Solver: Binary quadratic and max cut solver*, http://biqmac.uni-klu.ac.at/

[13] J. Spencer, *Three hundred million points suffice*, J. Comb. Theory Ser. A, 49 (1988), 210–217. Also see erratum by M. Hovey in Vol. 50, p. 323.

Bounding the pseudoachromatic index of the complete graph via projective planes

Gabriela Araujo-Pardo [1,2]

Instituto de Matemáticas
Universidad Nacional Autónoma de México
Ciudad Universitaria 04510 México, D.F.

Ricardo Strausz [3]

Instituto de Matemáticas
Universidad Nacional Autónoma de México
Ciudad Universitaria 04510 México, D.F.

.

Abstract

Let $q = 2^\beta$ and $n = q^2 + q + 1$. Further, let $G = L(K_n)$ be the *complete line graph* and $\psi(G)$ its pseudoachromatic number. By exhibiting an explicit colouring of $E(K_n)$, we show that $\psi(G) \geq q^3 + q$. This result improves the bound $\psi(G) \geq q^3 + 1$ due to Jamison (1989) [9].

Keywords: Graph colourings, pseudoachromatic number, projective planes.

[1] Partially supported by PAPIIT-UNAM, México, Proyecto IN106305-3

[2] Email: `garaujo@math.unam.mx`

[3] Email: `strausz@math.unam.mx`

1 Introduction

Let $G = (V, E)$ be a (simple) graph. A *complete colouring* of G is a function $\varsigma\colon V \to [k]$ from its vertices onto a k-set $[k] := \{0, 1, \ldots, k-1\}$ —the set of colours— such that for each pair of (different) colours $i, j \in [k]$ there is an edge $e = uv \in E$ which uses the two colours in its vertices

$$\varsigma(e) = \{\varsigma(u), \varsigma(v)\} = \{i, j\}.$$

The *pseudoachromatic number* $\psi(G)$ is the maximum k for which there exists such a complete colouring. The *achromatic number* $\alpha(G)$ is defined analogously to the pseudoachromatic number, but the colouring is required also to be *proper*, i.e., that no edge is coloured monochromatic.

The pseudoachromatic number was introduced by Gupta [7] while proving that

$$\chi(G) + \psi(G^c) \leq n + 1,$$

where $n = |V|$ is the order of G and G^c denotes its complement. This result refines that of Harary and Hedetniemi [8] which asserts that

$$\chi(G) + \alpha(G^c) \leq n + 1.$$

Both inequalities where motivated by the well-known one due to Nordhaus and Gaddum [11]:

$$\chi(G) + \chi(G^c) \leq n + 1.$$

Clearly, $\chi(G) \leq \alpha(G) \leq \psi(G)$.

Further interest of such invariants can be found in [2,3,5] (see also [10] for a surprising connection with abstract convexity).

We are mainly interested in the pseudoachromatic number of the complete graph's line graph; that is, we will bound, from below, $\psi(L(K_n))$ exhibiting a complete colouring of the edges of K_n. For this, we use some combinatorial properties of the projective plane of even order (see [4] for similar results on the achromatic number); explicitly, we prove that

Theorem 1.1 *If $q = 2^\beta$ and $n = q^2 + q + 1$, then $\psi(L(K_n)) \geq q^3 + q$.*

In the case $\beta = 1$ (i.e., in the case of the Fano Plane) our colouring does not improve the known one; indeed, it is easy to see that $\psi(L(K_7)) = \alpha(L(K_7)) = 11$ (see [1]). However, for $\beta > 1$ this result improves the bound $\psi(L(K_n)) \geq \alpha(L(K_n)) \geq q^3 + 1$ due to Jamison [9], whenever both theorems can be applied —e.g., whenever $q - 1$ is an odd prime power.

2 Preliminaries

If $n = q^2 + q + 1$ and $q = p^\beta$ is a prime power, then there exists an algebraic projective plane $\Pi_q = PG(2, q)$ which consists of n points and n lines, each line being a subset of $q + 1$ points, such that every two lines intersect in a unique point and every pair of points is contained in a unique line. Such a projective plane defines a partition on the edges of K_n induced by the following equivalence relation:

$$ab \sim cd \iff \overline{ab} = \overline{cd},$$

where $\overline{ab}$ denotes the (unique) line which contains the points a and b. Here, and in the sequel, we identify the points of Π_q with the vertices of K_n, and the lines with the complete subgraph induced by theirs points. We denote by $[q] = \{0, 1, \ldots, q-1\}$ the canonic q-set.

We will use the following useful description of Π_q. There is an *affine* set of q^2 points, which we identify with the set $[q] \times [q]$, and another set of $q + 1$ points which are elements of the so-called *line at infinity*

$$\ell_\infty := \{P_0, \ldots, P_{q-1}, P_\infty\}.$$

Besides the line at infinity, there are other q lines which contains the point P_∞; we say that they have *slope equal to infinity*, and denote them by

$$[x = i] := \{(i, 0), (i, 1), \ldots, (i, q-1), P_\infty\},$$

for each $i \in [q]$. The rest of the lines are defined, in the affine part, by equations of the form $y = mx + b$ — all the arithmetic is done in the Galois field $GF(q)$ — and we add, at infinity, the point P_m; explicitly, we write

$$[m, b] := \{(x, y) : y = mx + b\} \cup \{P_m\},$$

to denote the remaining q^2 lines.

3 The colouring

Let $n = q^2 + q + 1$, $q = 2^\beta$ and $\beta \in \mathbb{N}$ be natural numbers, and denote by $G = L(K_n)$ the line graph of K_n. To show that

$$\psi(G) \geq q^3 + q = (q-1)n + (q+1)$$

it is enough to exhibit a complete colouring of G with that number of colours. For, let $C = [q-1]$ be a set of colours — we will use them as a "pattern box of colours" — and let $\widehat{C} = C_1 \sqcup \cdots \sqcup C_n$ be the disjoint union of n copies of C. Further, consider an extra box of colours $D = [q+1]$. We will exhibit a complete colouring $\varsigma\colon V(G) \to \widehat{C} \sqcup D$ where each line ℓ of Π_q will use a different box of colours C_ℓ and an extra colour $d_\ell \in D$ from the last extra box; that is, $\varsigma(\ell) = C_\ell \cup \{d_\ell\}$, where $\varsigma(\ell) = \{\varsigma(uv) : \overline{uv} = \ell\}$. Here, and in the sequel, we identify the vertices of K_n and the points of Π_q (see Section 2).

It is well known that the complete line graph $L(K_q)$ of even order q accepts a complete proper colouring with $q-1$ colours; that is, $\alpha(L(K_q)) \geq q-1$ if q is even (see Füredi [6] for a nice discussion and generalisation of the subject). In such a colouring, each colour class induces a matching into the complete graph. We extend that colouring to $L(K_{q+1})$ by, first coping the colours from the edges incident to a fixed vertex $v \in V(K_q)$, to the new vertex $u \in V(K_{q+1})$, and then, adding a new colour to the edge $uv \in E(K_{q+1})$ (see Figure 1 for the case $q = 4$).

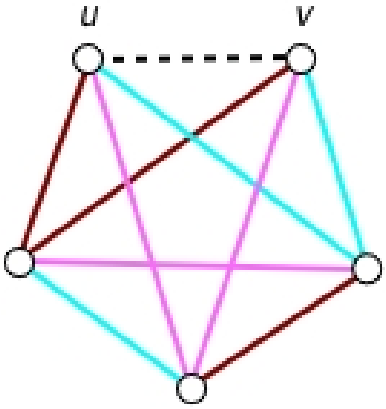


Fig. 1. A complete colouring of $L(K_5)$

Now, for each line ℓ of Π_q we can copy such a colouring into the edges of the complete subgraph induced by its points — we just have to choose the "special" edge uv and pick-up colours from the box C_ℓ, which has order $q-1$; the colour of uv will be picked-up from the box D.

To select the special edges we proceed as follows. For each line with slope equal to infinity, say $[x = i]$, we pick the point $(i, 0)$ and the colour $i \in D$ to assign it to the edge $P_\infty(i, 0)$; we choose different colours for different lines, i.e., $d_i \neq d_j$ if $i \neq j$. We use the remaining colour $q \in D$ for the pair $P_\infty P_0$.

To choose the remaining special edges, to be coloured from the box D, consider the line $\ell = [m, b] \neq [0, 0]$ and let $Q = \ell \cap [x = m]$. If $Q \in [0, 0]$ then we choose the edge QP_m as the special edge and colour it with $q \in D$;

otherwise we colour such an edge with colour $m \in D$. Finally, colour the line $\ell = [0,0]$ arbitrarily; i.e., choose any edge as the special one (see Figure 2).

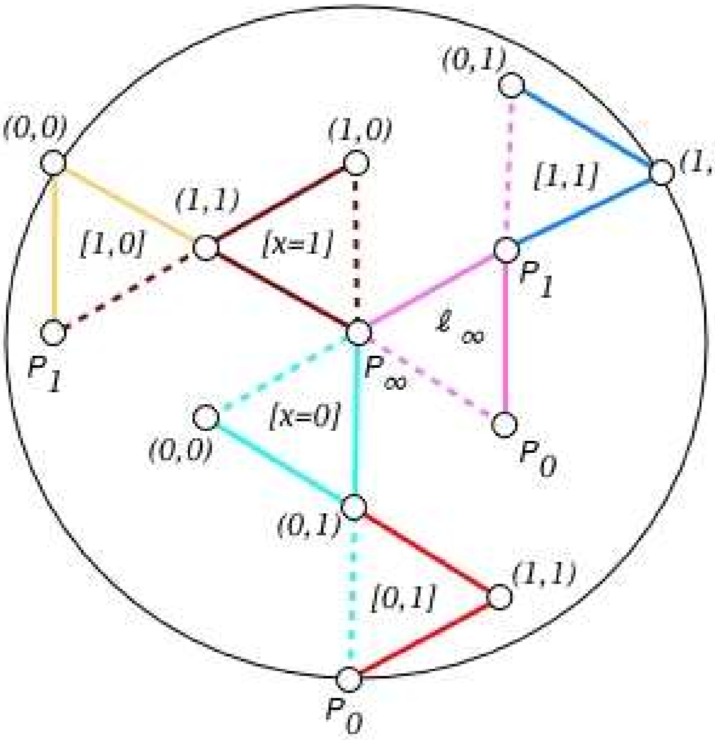


Fig. 2. The colouring of $L(K_7)$ related to the Fano plane.

From here, it is not hard to see that $\varsigma\colon V(G) \to \widehat{C} \sqcup D$, as described above, is a complete colouring of $L(K_n)$ with $|\widehat{C} \sqcup D| = (q-1)n + (q+1) = q^3 + q$ colours (see [1] for details). □

References

[1] Araujo-Pardo, G., Strausz, R.; *Bounding the pseudoachromatic number via finite geometries* (in preparation).

[2] Bhave, V.N.; *On the pseudoachromatic number of a graph*, Fundementa Mathematicae **102** (1979) 159–164.

[3] Bollobás, B., Reed, B., Thomason A.; "An extremal function for the achromatic number", *Graph Structure Theory*, Contemporary Mathematics Vol. 147, American Mathematical Society, Providence, Rhode Island, pp. 161–165, 1993.

[4] Bouchet, A.; *Indice achromatique des graphes multiparti complets et reguliers*, Cahiers centre Etudes Rech. Oper. **20(3–4)** (1978) 331–340.

[5] Farber, M., Hann, G., Hell, P., Miller, D.; *Concerning the achromatic number of graphs*, J. Comb. Theory Ser. B. **40(1)** (1986) 21–39.

[6] Füredi, Z.; *Covering the complete graph by partitions*, Discrete Math., **75** (1989) 217–226.

[7] Gupta, R.P.; "Bounds on the chromatic and achromatic numbers of complementary graphs", *Recent trends in combinatorics*, W.T. Tutte (Ed.), Proceedings of the Third Conference on Combinatorics, Waterloo, 1968, Academic Press, New York, 1969, pp. 229–235.

[8] Harary, F., Hedetniemi, S.T.; *The achromatic number of a graph*, J. Combin. Theory **8** (1970) 154–161.

[9] Jamison, R.E.; *On the edge achromatic numbers of complete graphs*, Discrete. Math. **74** (1989) 99–115.

[10] Montellano-Ballesteros, J.J., Pór, A., Strausz, R.; *Tverberg-type theorems for separoids*, Discrete Comput. Geom. **35** (2006) 513–523.

[11] Nordhaus, E.A., Gaddum, J.W.; *On complementary graphs*, Amer. Math. Monthly **63** (1956) 175–177.

Linear-Interval Dimension and PI Orders

Márcia R. Cerioli[a,c,1], Fabiano de S. Oliveira[c,2] and
Jayme L. Szwarcfiter[a,b,c,3]

[a] *Instituto de Matemática*

[b] *Núcleo de Computação Eletrônica*

[c] *COPPE/Sistemas e Computação,*
Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21945-970,
Rio de Janeiro, Brazil

Abstract

A PI graph G is the intersection graph of a family of triangles ABC between two distinct parallel lines L_1 and L_2, such that A is on L_1 and $\overline{BC}$ is on L_2. We study the orders defined by transitive orientations of the complement of G, the PI orders. We describe a characterization for such orders in terms of a special order dimension called linear-interval dimension. We show that the linear-interval dimension of an order is a comparability invariant, which generalizes the well-known result that the interval dimension is a comparability invariant.

Keywords: Order dimensions, intersection graphs, PI graphs.

[1] Partially supported by CAPES, CNPq and FAPERJ. Email: `cerioli@cos.ufrj.br`

[2] Supported by a CNPq fellowship. Email: `fabsoliv@cos.ufrj.br`

[3] Partially supported by CAPES, CNPq and FAPERJ. Email: `jayme@nce.ufrj.br`

1 Introduction

A graph G is an *intersection graph* if we can associate a family of sets to G, each set corresponding to a vertex, such that $(u, v) \in E(G)$ if and only if the sets corresponding to u and v have non-empty intersection (or simply that they are intersecting). We call such a family a *model* of G. An *interval graph* is the intersection graph of a family of intervals of the real line, called an *interval model*. Let L_1 and L_2 be two distinct parallel lines. A *permutation graph* is the intersection graph of a family of line segments, such that each segment has an extreme point on L_1 and the other on L_2. A *trapezoid graph* is the intersection graph of a family of trapezoids ABCD, such that $\overline{AB}$ is on L_1 and $\overline{CD}$ is on L_2 [2]. A *point-interval graph* or *PI graph* is the intersection graph of a family of triangles ABC, such that A is on L_1 and $\overline{BC}$ is on L_2, called a *PI model* [2]. Let T be a triangle ABC. Denote $t(T) = A$ and $b(T) = \overline{BC}$. Figure 1 illustrates a PI graph and a PI model of it.

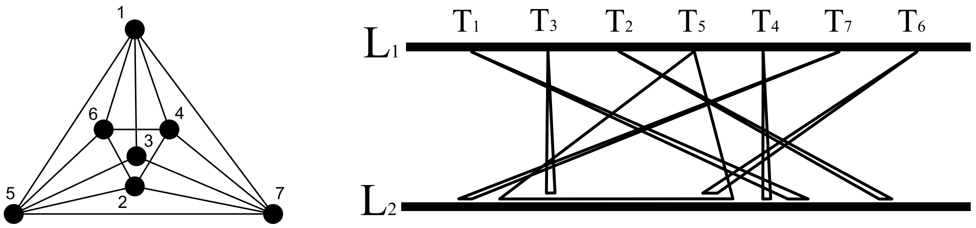


Figure 1. A PI graph and a PI model.

The left and right extreme points of an interval I are denoted by $\ell(I)$ and $r(I)$, respectively. When $\ell(I) = r(I)$, we say that I is *trivial*. Given a set of distinct parallel lines $L_1, \ldots, L_n$, a *figure* is a set of intervals $\{I^1, \ldots, I^n\}$, where $I^i \subset L_i$ for all $1 \leq i \leq n$. Two figures $F_1 = \{I_1^i \mid 1 \leq i \leq n\}$ and $F_2 = \{I_2^i \mid 1 \leq i \leq n\}$ are *disjoint* if one of them, say F_1, is such that $r(I_1^i) < \ell(I_2^i)$ for all $1 \leq i \leq n$, denoted by $F_1 \ll F_2$. Two figures F_1 and F_2 *are intersecting* if they are not disjoint, denoted by $F_1 \times F_2$.

An *order* $P = (X, \prec)$ is an irreflexive and transitive binary relation $\prec$ on the set X. If $x \prec y$ or $y \prec x$, we say that x and y are *comparable*. Otherwise, they are *incomparable* and we denote this relation by $x \parallel y$. An order is a *linear order* if any two distinct elements of it are comparable. An order $(X, \prec)$ is an *interval order* if it can be associated to an interval model $\{I_x \mid x \in X\}$, such that $x \prec y$ if and only if $I_x \ll I_y$. An order $P' = (X, \prec')$ is an *extension* of an order $P = (X, \prec)$ if $x \prec y \implies x \prec' y$, for all $x, y \in X$. If an extension is a linear order, we call it a *linear extension*. Similarly, we call an extension an *interval extension* if it is an interval order.

Let $P = (X, \prec_P)$ and $Q = (X, \prec_Q)$ be orders. The relation $P \cap Q$ is defined to be the order $(X, \prec_R)$ such that $x \prec_R y \iff x \prec_P y$ and $x \prec_Q y$. Let L be a set $\{P_1, \ldots, P_k\}$ of extensions of P. We call L a *realizer* of P if $P = \bigcap_{i=1}^{k} P_i$. The *linear (interval) dimension* of an order P is the least k such that there exists a realizer of P containing precisely k linear (interval) extensions [9].

Let G be a graph and P the order defined by some transitive orientation of G. We call G and $\overline{G}$ respectively the *comparability* and the *incomparability graph* of P. A property about orders is said to be a *comparability invariant* if either all orders with the same comparability graph have such a property or none of them does. It is known that both the linear and the interval dimensions are comparability invariants [4,5].

It is clear that the class of PI graphs generalizes both classes of interval and permutation graphs and is generalized by the class of trapezoid graphs. Both interval and permutation graph classes are well-known, and there are algorithms to recognize both of them in linear time [8]. Trapezoid graphs are also recognized efficiently, and the fastest algorithm is due to Ma and Spinrad [7], which runs in $O(n^2)$ time. Another approach for the recognition of such graphs is based on the following steps. First, characterize trapezoid orders (transitive orientations of the complements of trapezoid graphs) as those having interval dimension at most 2. Then, formulate a polynomial time algorithm to recognize orders of interval dimension at most 2 [3,6]. The recognition problem of PI graphs has been open since 1987 [2,1,8]. In this work, we reduce the recognition problem of PI graphs to that of recognizing PI orders, and characterize such orders as those having linear-interval dimension at most $(2, 1)$. If there is a polynomial algorithm to recognize such orders, PI graphs would be recognized efficiently.

2 Linear-Interval Dimension and PI Orders

Let P be an order and F be a realizer of P. We say that F is a (p, q)*-linear-interval realizer* of P, if F is an interval realizer with p elements and precisely q of them are non-linear. We define $(p, q) \leq (p', q')$ if (p, q) is lexicographically smaller than or equal to (p', q'). A *linear-interval dimension* of an order P, denoted by *lidim*(P), is the lexicographically smallest ordered pair (p, q) such that there exists a (p, q)-linear-interval realizer of P. We show that the linear-interval dimension of an order is a comparability invariant, as follows.

Given a graph $G = (V, E)$, we say that $A \subseteq V$ is a *homogeneous set* if every vertex in $V \setminus A$ is adjacent either to all of the vertices in A or to none of them. Let $P_1 = (X, \prec_1)$ and $P_2 = (X, \prec_2)$ be orders with the same comparability graph G. We say that P_2 is obtained from P_1 by an *elementary reversal* if there is a homogeneous set $A \subseteq X$ of G that satisfies the following properties: (i) A is not an independent set of G; (ii) if x, y are not both in A, then $x \prec_1 y \iff x \prec_2 y$; and (iii) if $x, y \in A$, then $x \prec_1 y \iff y \prec_2 x$.

Theorem 2.1 (Gallai [4]) *Let π be a property about orders. In order to prove that π is a comparability invariant, it suffices to prove that if an order Q is obtained from an order P by an elementary reversal and π holds for P, then π holds for Q.*

The proof of the next Lemma is straightforward.

Lemma 2.2 ([4]) *Let $P_1 = (X, \prec_1)$ and $P_2 = (X, \prec_2)$ be orders such that P_2 is obtained from P_1 by an elementary reversal of the homogeneous set $A \subseteq X$. Then, $X \setminus A$ is partitioned into the sets $P_1^-(A) = \{x \in X \setminus A \mid x \prec_1 a$ for all $a \in A\}$, $P_1^+(A) = \{x \in X \setminus A \mid a \prec_1 x$ for all $a \in A\}$ and $P_1^\times(A) = \{x \in X \setminus A \mid x \parallel_1 a$ for all $a \in A\}$.*

Given distinct parallel lines $L_1, \ldots, L_p$ and a set X, a family of figures $\{F_x \mid x \in X\}$, where $F_x = \{I_x^i \mid 1 \leq i \leq p\}$ for all $x \in X$, is a (p, q)-*linear-interval model* if: (i) for all $1 \leq i \leq q$, I_x^i is non-trivial for some $x \in X$; (ii) for all $q < i \leq p$, I_x^i is trivial for all $x \in X$. Figure 2 illustrates a $(3, 2)$-linear-interval model. To make the picture clearer, $\ell(I_x^i)$ is joined to $\ell(I_x^{i+1})$ and $r(I_x^i)$ to $r(I_x^{i+1})$ both by a line segment, for all $1 \leq i < p$.

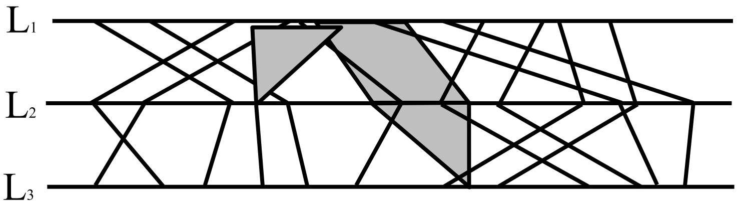


Figure 2. A $(3, 2)$-linear-interval model.

An order $P = (X, \prec)$ is (p, q)-*linear-interval representable* if there exists a (p, q)-linear-interval model $\{F_x \mid x \in X\}$ such that $x \prec y \iff F_x \ll F_y$. It can be shown the equivalence between these two concepts.

Lemma 2.3 *Let $P = (X, \prec)$ be an order. Then P has a (p, q)-linear-interval realizer if and only if P is (p, q)-linear-interval representable.*

Let S be a set of points and $M < N$ be real constants such that $s \geq M$ for all $s \in S$. Let $W(S) = max\{s - M \mid s \in S\}$. The operation of *fitting* S between M and N is that of making the width of $W(S)$ equal to $N - M$, such that $s \in S$ is moved to $(s - M)(N - M)/W(S) + M$. Let $\mathcal{R} = \{F_x \mid x \in X\}$ be a (p, q)-linear-interval model and $M < N$ be real constants. The operation of *resizing* $Y \subseteq X$ between M and N consists of fitting F_x into the region delimited by the vertical lines at M and N of the model, for all $x \in Y$, as follows. Let R be the rectangle with minimum width in which all figures corresponding to Y are included. First, slide horizontally those figures (and therefore R) such that the left side of R is on the vertical line at M. Next, apply fitting operation on $\{\ell(I_y^i), r(I_y^i) \mid y \in Y, 1 \leq i \leq p\}$ between M and N.

Theorem 2.4 *Being (p, q)-linear-interval representable is a comparability invariant.*

Proof (Sketch) By Theorem 2.1, it is sufficient to show that if $P = (X, \prec_P)$ is (p, q)-linear-interval representable, then $Q = (X, \prec_Q)$ is (p, q)-linear-interval representable, where Q is obtained from P by an elementary reversal of $A \subseteq X$.

Let $\mathcal{R} = \{F_x \mid x \in X\}$ be a (p, q)-linear-interval model of P. By the property (i) of an elementary reversal, there exist $b, c \in A$ such that $b \prec_P c$. Let $M = r(I_b^1)$ and $N = \ell(I_c^1)$. It is possible to adjust $\mathcal{R}$ so that $r(I_b^i) = M$ and $\ell(I_c^i) = N$, for each $1 \leq i \leq p$, by sliding conveniently the extreme points in each horizontal line of $\mathcal{R}$. Then, apply the resizing operation on A between M and N. Next, flip horizontally the figures corresponding to vertices in A through a vertical line at $(M + N)/2$, obtaining the final model $\mathcal{R}' = \{F_x' \mid x \in X\}$. We claim that $\mathcal{R}'$ is a (p, q)-linear-interval model of Q.

Since the resizing operation is composed with a horizontal flip, it follows that for all $x, y \in A$, $x \prec_Q y \iff y \prec_P x \iff F_x' \ll F_y'$, which is according to our claim. It also holds that for all $x, y \in X \setminus A$, $x \prec_Q y \iff x \prec_P y \iff F_x' \ll F_y'$, again consistent with our claim. Finally, for all $x \in X \setminus A$ and $y \in A$, consider the subcases where $x \prec_Q y$, $y \prec_Q x$ or $x \parallel_Q y$. If $x \prec_Q y$, then $x \prec_P y$. By Lemma 2.2, $x \prec_P a$, for all $a \in A$, and in particular, $x \prec_P b$. Therefore, $F_x \ll F_b$ and then $F_x' \ll F_a'$, for all $a \in A$, and in particular, $F_x' \ll F_y'$. The other two cases are similar. □

Corollary 2.5 *Linear-interval dimension is a comparability invariant.*

As a consequence, we have the well-known result:

Corollary 2.6 *Interval dimension is a comparability invariant.*

Proof Let P and Q be orders with the same comparability graph. By Corollary 2.5, $lidim(P) = lidim(Q) = (p, q)$. Then, $idim(P) = idim(Q) = p$. □

An order $(X, \prec)$ is a *PI order* if there exists a PI model $\mathcal{R} = \{T_x \mid x \in X\}$, such that $x \prec y$ if and only if $T_x \ll T_y$.

Theorem 2.7 *An order P is a PI order if and only if $lidim(P) \leq (2,1)$.*

Proof Let $P = (X, \prec)$ be a PI order and $\mathcal{R} = \{T_x \mid x \in X\}$ be a PI model of P. Consider the orders $P_I = (X, \prec_I)$ and $P_L = (X, \prec_L)$ such that $x \prec_I y \iff b(T_x) \ll b(T_y)$ and $x \prec_L y \iff t(T_x) \ll t(T_y)$. Since $x \prec y \iff x \prec_L y$ and $x \prec_I y$, then $\{P_L, P_I\}$ is either a $(2,1)$- or a $(2,0)$-linear-interval realizer of P. Conversely, let $P = (X, \prec)$ be an order such that $lidim(P) \leq (2,1)$. If $lidim(P) \leq (2,0)$, the result holds. Suppose $lidim(P) = (2,1)$ and then let $\{P_I, P_L\}$ be a $(2,1)$-linear-interval realizer of P, such that $P_I = (X, \prec_I)$ is the non-linear interval order. Let $\mathcal{R}_I = \{I_x \mid x \in X\}$ be an interval model of P_I. We build a PI model $\mathcal{R} = \{T_x \mid x \in X\}$ such that $b(T_x) = I_x$ for all $x \in X$, and $t(T_x) \ll t(T_y) \iff x \prec_L y$. Thus, $x \prec y \iff T_x \ll T_y$. □

Corollary 2.8 *Being a PI order is a comparability invariant.*

References

[1] Brandstädt, A., V. B. Le, and J. P. Spinrad, "Graph Classes: a Survey," SIAM, Philadelphia, 1999.

[2] Corneil, D. J. and P. A. Kamula, *Extensions of permutation and interval graphs*, Congressus Numerantium **58** (1987), 267–275.

[3] Felsner, S., M. Habib, and R. Möhring, *On the interplay between interval dimension and ordinary dimension*, SIAM Journal of Discrete Mathematics, **7** (1994), 32–40.

[4] Golumbic, M., and A. Trenk, "Tolerance Graphs," Cambridge University Press, 2003.

[5] Habib, M., D. Kelly, and R. Möhring, *Interval dimension is a comparability invariant*, Discrete Applied Mathematics **88** (1992), 211–229.

[6] Langley, L., *Recognition of orders of interval dimension 2*, Discrete Applied Mathematics, **60** (1995), 257–266.

[7] Ma, T. H., and J. P. Spinrad, *On the 2-chain subgraph cover and related problems*, Journal of Algorithms **17:2** (1994), 251–268.

[8] Spinrad, J. P., "Efficient Graph Representations," AMS, 2003.

[9] Trotter, W. T., "Combinatorics and Partially Ordered Sets," The Johns Hopkins University Press, 1992.

On cliques of Helly Circular-arc Graphs

Min Chih Lin[a,1], Ross M. McConnell[b],
Francisco J. Soulignac[a,2] and Jayme L. Szwarcfiter[c,3]

[a] *Departamento de Computación, FCEN, Universidad de Buenos Aires, Buenos Aires, Argentina.* {oscarlin,fsoulign}@dc.uba.ar

[b] *Computer Science Department, Colorado State University, Fort Collins, CO 80523-1873, USA.* rmm@cs.colostate.edu

[c] *Universidade Federal do Rio de Janeiro, Inst. de Matemática, NCE and COPPE, Caixa Postal 2324, 20001-970 Rio de Janeiro, RJ, Brasil.* jayme@nce.ufrj.br

Abstract

A *circular-arc graph* is the intersection graph of a set of arcs on the circle. It is a *Helly circular-arc graph* if it has a *Helly model*, where every maximal clique is the set of arcs that traverse some *clique point* on the circle. A *clique model* is a Helly model that identifies one clique point for each maximal clique. A Helly circular-arc graph is *proper* if it has a Helly model where no arc is a subset of another. In this paper, we show that the clique intersection graphs of Helly circular-arc graphs are precisely the proper Helly circular-arc graphs. This yields the first polynomial (linear) time recognition algorithm for the clique graphs of Helly circular-arc graphs. Next, we develop an $O(n)$ time algorithm to obtain a clique model of Helly model, improving the previous $O(n^2)$ bound. This gives a linear-time algorithm to find a proper Helly model for the clique graph of a Helly circular-arc graph. As an application, we find a maximum weighted clique of a Helly circular-arc graph in linear time.

Keywords: algorithms, Helly circular-arc graphs, proper Helly circular-arc graphs, clique graphs, maximum weight cliques.

1 Introduction

If $G = (V(G), E(G))$ is a graph, we denote by n and m the values of $|V(G)|$ and $|E(G)|$. A *complete set* is a subset of pairwise adjacent vertices, while a *clique* is a maximal complete set. If the vertices of G are weighted, let us say that G is a *weighted graph*. The *weight* of a clique is the sum of the weights of its vertices. The *clique graph* $K(G)$ of G is the intersection graph of its cliques.

A graph is a *clique graph* if it is isomorphic to $K(G)$ for some graph G [5,8]. Clique graphs of several classes have been characterized and several algorithms are known for testing if a graph is a clique graph of some class. Some of these can be recognized in polynomial time [11]. However, in the last year the complexity of recognition of clique graphs of arbitrary graphs was proved to be NP-Hard [1].

A *circular-arc* (CA) model $\mathcal{M}$ is a pair $(C, \mathcal{A})$, where C is a circle and $\mathcal{A}$ is a collection of arcs of C. When traversing the circle C, we will always choose the clockwise direction. If s, t are points of C, write (s, t) to mean the arc of C defined by traversing the circle from s to t. Call s, t the *extremes* of (s, t), while s is the *beginning point* and t the *ending point* of the arc. For $A \in \mathcal{A}$, write $A = (s(A), t(A))$. The *extremes* of $\mathcal{A}$ are those of all arcs in $\mathcal{A}$. Without loss of generality, we assume that all arcs of C are open arcs, no two extremes coincide, and no single arc covers C. We will say that $\epsilon > 0$ is *small enough* if ϵ is smaller than the minimum arc distance between two consecutive extremes of $\mathcal{A}$.

When no arc of $\mathcal{A}$ contains any other, $\mathcal{M}$ is a *proper* circular-arc (PCA) model. When every set of pairwise intersecting arcs share a common point, $\mathcal{M}$ is called a *Helly* circular-arc (HCA) model. If no two arcs of $\mathcal{A}$ cover C, then the model is called *normal*. A *proper Helly* circular-arc model (PHCA) is one which is both HCA and PCA. Finally, an *interval model* is a CA model where $\bigcup_{A \in \mathcal{A}} A \neq C$. A *CA (PCA) (HCA) (PHCA) (interval) graph* is the intersection graph of a CA (PCA) (HCA) (PHCA) (interval) model. A graph is *K(HCA)* if it is the clique graph of some HCA graph. Two CA models are *equivalent* when they have the same intersection graph.

Denote by $\mathcal{A}(p)$ the collection of arcs that contain p. Clearly, the vertices

[1] Partially supported by UBACyT Grants X184 and X212, CNPq under PROSUL project Proc. 490333/2004-4.

[2] Partially supported by UBACyT Grant X184 and CNPq under PROSUL project Proc. 490333/2004-4, and by a grant of the YPF Foundation.

[3] Partially supported by CNPq and FAPERJ, Brasil.

corresponding to $\mathcal{A}(p)$ form a complete set. If this set is a clique, then p is called a *clique point*. For points p, p' on the circle, p *(properly) dominates* p' if $\mathcal{A}(p')$ is (properly) contained in $\mathcal{A}(p)$. When $\mathcal{A}(p) = \mathcal{A}(p')$ then p, p' are *equivalent*. Point p is a *complete point* if it is not properly dominated by any other point. In HCA graphs there is a one-to-one correspondence between cliques and non-equivalent complete points. An *intersection segment* (s, t) is a pair of consecutive extremes where s is a beginning point and t is an ending point. Points inside intersection segments are called *intersection points*. Every complete point is an intersection point, but the converse is not necessarily true [4], because there can be multiple intersection segments that are contained in exactly the same set of arcs. However, when $\mathcal{M}$ is a PHCA model, then every intersection point is also a complete point. A *complete (intersection) (clique) point representation* of $\mathcal{M}$ is a maximal set of non-equivalent complete (intersection) (clique) points. Let $I = (s(A_i), t(A_j))$ be an intersection segment and $p \in I$. The *arc reduction* of p is the arc $(s(A_i), t(A_k))$ where $A_k \in \mathcal{A}(p)$ and $t(A_k)$ is the ending point farthest from p when traversing C. Observe that when $\mathcal{M}$ is PHCA then the arc reduction of p is A_i. Let Q be a clique (complete) point representation, the *clique (complete) model* (with respect to Q) is the model formed by the arc reductions of Q. In particular, any clique model of an HCA graph G is a PCA model of $K(G)$ [4].

Circular-arc graphs and its subclasses have been receiving much attention recently ([2,10]). For CA, PCA, HCA and PHCA graphs, there are several characterizations and linear linear time recognition algorithms which construct a model (see [6,7]). In [3], K(HCA) graphs are studied. It is proved that K(HCA) graphs are both PCA and HCA graphs. In the same paper some characterizations are shown, but these characterizations did not lead to a polynomial time recognition algorithm. On the other hand, in [4] an $O(n^2)$ time algorithm that outputs the clique graph of an HCA graph is described.

In Section 2, we prove that the class of K(HCA) graphs is precisely the class of PHCA graphs. A PHCA model can be obtained in $O(n)$ time from any PCA model of a PHCA graph [7], so PHCA graphs can be recognized in linear time. This implies a linear time recognition algorithm for K(HCA) graphs. In the last section, we also describe a new simple linear time algorithm for constructing a clique model of an HCA graph. In fact, we describe a more general linear time algorithm that finds a maximal dominating subset of some set of points in a CA model. This algorithm can be easily extended to find the weighted clique graph of an HCA graph, solving the maximum weighted clique problem in linear time. For CA graphs, the maximum weight clique problem can be solved in $O(n \log n + m \log \log n)$ time [9].

2 Characterization of K(HCA) graphs

Theorem 2.1 [4] *Let G be a PHCA graph. Then $K(G)$ is PHCA and every complete point model of G is a PHCA model of $K(G)$.*

Theorem 2.2 [7] *Let G be a PCA graph. Then G is a PHCA graph if and only if G contains neither W_4 nor 3-sun as induced subgraphs.*

Theorem 2.3 *Let G be a graph. Then the following are equivalent:*

(i) $G = K(H)$ *for some PHCA graph* H.

(ii) $G = K(H)$ *for some HCA graph* H.

(iii) G *is a PHCA graph.*

Proof (Sketch). Clearly (i) implies (ii). To see that (ii) implies (iii), consider an HCA graph H. Graph $G = K(H)$ is PCA [3], contains no 3-sun as an induced subgraph [5] and it is not a difficult task to check that G contains no W_4 as an induced subgraph. Hence, by Theorem 2.2, G is PHCA. Finally, we show that (iii) implies (i). Let G be a PHCA graph. If G is a proper interval graph, then (i) follows (see [11]). Otherwise, let $\mathcal{M} = (C, \mathcal{A})$ be a PHCA model of G and assume that it is also normal by [7]. By Theorem 2.1, it suffices to find a PHCA supermodel of $\mathcal{M}$ whose clique model is $\mathcal{M}$. Let $\mathcal{Q}$ be the set of arc reductions of $\mathcal{A}$ and $\mathcal{N} = \mathcal{A} \setminus \mathcal{Q}$. Observe that since $\mathcal{M}$ is PHCA, $\mathcal{Q}$ is a subset of $\mathcal{A}$. Also note that since G is not an interval graph then every arc of $\mathcal{A}$ contains at least one ending point of some other arc. Now, fix a small enough ϵ. For each arc $A_i \in \mathcal{N}$ let B_i be the arc $(s(A_j) - \epsilon, s(A_i) + \epsilon)$ where $t(A_j)$ is the first ending point that appears when traversing C from $s(A_i)$. If two arcs B_i, B_j share their beginning points, then modify one of them so that none of them is included in the other. We claim that $\mathcal{M}' = (C, \mathcal{A} \cup \{B_i : A_i \in \mathcal{N}\})$ is PHCA and has $\mathcal{M}$ as a clique model. To observe this, check that $\mathcal{M}'$ is a PHCA model and that the set of arc reductions of $\mathcal{M}'$ is precisely $\mathcal{A}$. □

To check whether a graph is K(HCA) is the same as to check if the graph is PHCA. This can be done in linear time [7].

3 Construction of a clique model of an HCA graph

In [4] an $O(n^2)$ algorithm for constructing a clique model of an HCA model is described. The algorithm consists of two well defined procedures: 1) Find a clique point representation Q of the model and 2) build the clique model with respect to Q. The first procedure is the bottleneck of the algorithm, and takes

$O(n^2)$ time, while the second procedure can be done in $O(n)$ time. In this section we develop a linear-time algorithm that reduces the bottleneck step to $O(n)$. Given a set P of points in an arbitrary CA model, the algorithm finds a minimum subset of the points that dominates all members of P in $O(n + |P|)$ time. Letting P be one intersection point in each of the $O(n)$ intersection segments solves the bottleneck step. From now on, let $\mathcal{M} = (C, \mathcal{A})$ be a CA model.

The *ascendant (descendant) semi-dominating sequence* of P is the subsequence $SD^+(P) = \{p_i \in P : \mathcal{A}(p_i) \not\subseteq \mathcal{A}(p_j)$ for all $p_j \in P$ and $j > i\}$ ($SD^-(P) = \{p_i \in P : \mathcal{A}(p_i) \not\subseteq \mathcal{A}(p_j)$ for all $p_j \in P$ and $j < i\}$).

Lemma 3.1 *Let $\mathcal{M} = (C, \mathcal{A})$ be a CA model and $P = \{p_1, \ldots, p_k\}$ be a sequence of circularly ordered points from C. Then both $SD^-(SD^+(P))$ and $SD^+(SD^-(P))$ are P-dominating sequences.*

Algorithms to find SD^+ and SD^- are symmetric. We describe the one to find SD^+. The algorithm works by induction on the size of a $P_i = \{p_1, p_2, \ldots, p_i\}$. After step i, we have a partition of $SD^+(P_i)$ into the following two sets:

- The members D_i that are contained in some arc that is a subset of the open interval (p_k, p_i), hence that cannot be dominated by any arc in $\{p_{i+1}$, p_{i+2}, $\ldots$, $p_k\}$. These are already known to be members of $SD^+(P)$.
- The set $Q_i = SD^+(P_i) - D_i$ in clockwise order $(q_1, q_2, \ldots q_j)$ of appearance in $[p_1, p_i]$. Their status as members of $SD^+(P)$ is uncertain; though they are in $SD^+(P_i)$, they might be dominated by points in $\{p_{i+1}, p_{i+2}, \ldots p_k\}$.

After step k, the algorithm returns $SD^+(P) = D_k \cup Q_k$. It remains to describe how to obtain D_{i+1} and Q_{i+1} from D_i and Q_i. We first find D_{i+1} by finding the members of Q_i that must be added to D_i. We need only consider the effect of arcs that begin in (p_k, p_{i+1}) and that end in $[p_i, p_{i+1})$, since arcs in (p_k, p_{i+1}) that end earlier have already been considered in determining D_i. Of these arcs, let D be the one that reaches farthest to the left, that is, whose beginning point is closest to p_k; D is the one that covers the most members of Q_i, so $D_{i+1} - D_i$ is just the points of Q_i that are covered by D.

The remaining members $Q_{i+1} = SD^+(P_{i+1}) - D_{i+1}$ are just the members of $Q_i - D_{i+1}$ that aren't dominated by p_{i+1}. A point $p \in Q_i - D_{i+1}$ is in Q_{i+1} if and only if it is contained in an arc A that doesn't contain p_{i+1}, since otherwise it would be dominated by p_{i+1}, and that contains p_k, since otherwise it would already be identified as a member of D_{i+1}. Of all such arcs, let D be the one that reaches farthest to the right of p_k; since this is the one that covers the most members of $Q_i - D_{i+1}$, Q_{i+1} is just the points of $Q_i - D_{i+1}$ that are contained in D.

It is easy to see that the algorithm can be implemented to run in $O(n+|P|)$ time using elementary techniques, since, at each step, the points of Q_i that are moved to D_{i+1} or discarded are a suffix of $(q_1, q_2, ..., q_j)$.

Theorem 3.2 *Let $\mathcal{M} = (C, \mathcal{A})$ be an HCA model of a graph G. Then a PHCA model of $K(G)$ can be found in $O(n)$ time.*

Theorem 3.3 *Let $\mathcal{M} = (C, \mathcal{A})$ be an HCA model of a graph G. Then the maximum weight clique of G can be found in $O(n)$ time.*

References

[1] Alcón, L., L. Faria, C. Figueiredo and M. Gutiérrez, *Clique graph recognition is NP-complete*, Proc. of WG 2006, Lect. Notes in Comput. Sci. **4271** (2006), 269–277.

[2] Brandstädt, A., V. Le and J. Spinrad, "Graph Classes: A Survey", SIAM, 1999.

[3] Durán, G. and M. Lin, *Clique graphs of Helly circular-arc graphs*, Ars Combin. **60** (2001), 255–271.

[4] Durán, G., M. Lin, S. Mera and J. Szwarcfiter, *Algorithms for clique-independent sets on subclasses of circular-arc graphs*, Discrete Appl. Math. **154** (2006), 1783–1790.

[5] Hamelink, R., *A partial characterization of clique graphs*, J. Combin. Theory Ser. B **5** (1968), 192–197.

[6] Lin, M., and J. Szwarcfiter, *Characterizations and recognition of circular-arc graphs and subclasses: a survey*, manuscript, 2006.

[7] Lin, M., F. Soulignac and J. Szwarcfiter, *Proper Helly circular-arc graphs*, Proc. of WG 2007 (2007), 235–246, Lect. Notes in Comput. Sci., to appear.

[8] Roberts, F. and J. Spencer, *A characterization of clique graphs*, J. Combin. Theory Ser. B **10** (1971), 102–108.

[9] Shih, W. and W. Hsu, *An $O(n \log n + m \log\log n)$ algorithm for finding a maximum weight clique in circular-arc graphs*, Inf. Process. Lett. **31** (1989), 129–134.

[10] Spinrad, J., "Efficient Graph Representations", Amer. Math. Soc., 2003.

[11] Szwarcfiter, J., *A survey on Clique Graphs*, in: "Recent Advances in Algorithms and Combinatorics", Springer, 109–136.

Arithmetic relations in the set covering polyhedron of circulant clutters

Néstor E. Aguilera [1,2]

Abstract

We study the structure of the set covering polyhedron of circulant clutters, $P(\mathscr{C}_n^k)$, especially the properties related to contractions that yield other circulant clutters. Building on work by Cornuéjols and Novick, we show that if $\mathscr{C}_n^k/N$ is isomorphic to $\mathscr{C}_{n'}^{k'}$, then certain algebraic relations must hold and N is the union of particular disjoint simple directed cycles. We also show that this property is actually a characterization. Based on a result by Argiroffo and Bianchi, who characterize the set of null coordinates of vertices of $P(\mathscr{C}_n^k)$ as being one of such N's, we then arrive at other characterizations, one of them being the conditions that hold between the existence of vertices and algebraic relations of certain parameters. With these tools at hand, we show how to obtain by algebraic means some old and new results, without depending on Lehman's work as is traditional in the field.

Keywords: circulant clutter, set covering polyhedron, directed cycle, relative prime numbers.

[1] Consejo Nacional de Investigaciones Científicas y Técnicas and Universidad Nacional del Litoral, Argentina. This work is supported in part by CONICET Grant PIP 5810.

[2] Email: `aguilera@ceride.gov.ar`

1 Background and results

Given a clutter $\mathscr{C}$ with vertices $V(\mathscr{C})$ and edges $E(\mathscr{C})$, we denote by

$$P(\mathscr{C}) = \{x \in \mathbb{R}^n : M(\mathscr{C})x \geq \mathbf{1},\, x \geq \mathbf{0}\},$$

the corresponding set covering polyhedron. Our work is concerned with the circular clutters $\mathscr{C}_n^k$, having vertex set $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ and edges $\{i, i \oplus 1, \ldots, i \oplus (k-1)\}$ for $i \in \mathbb{Z}_n$ ($\oplus$ being addition modulo n), so that it will convenient to regard vectors in $\mathbb{R}^n$ as having coordinates $(x_0, \ldots, x_{n-1})$. $G(\mathscr{C}_n^k)$ will denote the directed graph having vertex set $V(\mathscr{C}_n^k) = \mathbb{Z}_n$, and (i, i') is an arc of $G(\mathscr{C}_n^k)$ if and only if $i' \ominus i \in \{k, k+1\}$.

Cornuéjols and Novick [5] described many ideal and minimally non ideal (mni for short) clutters, in particular finding all of the clutters $\mathscr{C}_n^k$ which are ideal or mni. Their results are based on the work by Lehman [6,7], and the following lemma, which is central to our work:

Lemma 1.1 (lemma 4.5 in [5]) *Suppose $2 \leq k \leq n-2$. If a subset N of $V(\mathscr{C}_n^k)$ induces a simple directed cycle, D, in $G(\mathscr{C}_n^k)$, then there exists $n_1, n_2, n_3 \in \mathbb{Z}_+$, $n_1 \geq 1$, such that*

(i) $n n_1 = k n_2 + (k+1)\, n_3$,

(ii) $\gcd(n_1, n_2, n_3) = 1$,

(iii) If $k - n_1 \leq 0$, then $E(\mathscr{C}_n^k / N) = \emptyset$ or $\{\emptyset\}$. If $k - n_1 \geq 1$, then $\mathscr{C}_n^k / N$ is of the form $\mathscr{C}_{n-n_2-n_3}^{k-n_1}$.

In a previous paper [1] we have shown that the condition $\gcd(m, n_1) = 1$ in lemma 1.1 is not only necessary, but also sufficient, for the existence of a simple directed cycle, giving a constructive proof:

Theorem 1.2 *Let n, k and m be given, $1 \leq k \leq n-1$, $1 \leq m \leq n-1$. Then there exists a simple directed cycle D in $G(\mathscr{C}_n^k)$ with $|V(D)| = m$ if and only if the following two conditions are satisfied:*

$$\left\lceil \frac{km}{n} \right\rceil = \left\lfloor \frac{(k+1)\, m}{n} \right\rfloor, \tag{1}$$

and if n_1 is its common value, then $\gcd(m, n_1) = 1$.

Moreover, such a cycle can be constructed in $O(m)$ steps.

The following result—with little changes—is part of the proof given by Cornuéjols and Novick of lemma 1.1, and reveals an important part of the structure of cycles in $G(\mathscr{C}_n^k)$:

Proposition 1.3 *Suppose $1 \leq k \leq n-1$, D is a simple directed cycle in $G(\mathscr{C}_n^k)$, having n_2 arcs of length k and n_3 arcs of length $k+1$, n_1 is defined as in lemma 1.1(i), $N = V(D)$ is written in the canonical form*

$$N = \{i_0, i_1, \dots, i_{m-1}\} \quad \textit{so that} \quad i_0 < i_1 < \cdots < i_{m-1}, \tag{2}$$

and $s : N \to N$ is defined by $s(i) = i'$ if (i, i') is an arc of D.

Then, if $\underset{m}{\oplus}$ indicates addition modulo m,

$$s(i_j) = i_{j \underset{m}{\oplus} n_1} \quad \textit{for } j = 0, \dots, m-1, \tag{3}$$

or, equivalently, $|\{i, i \oplus 1, \dots, s(i) \ominus 1\} \cap N| = n_1$.

The following is a generalization to the case of several cycles:

Theorem 1.4 *Suppose $1 \leq k \leq n-1$, $1 \leq m \leq n-1$.*

If $D_1, \dots, D_d$ are disjoint simple directed cycles in $G(\mathscr{C}_n^k)$, all having length m/d, $N = \cup_u V(D_u)$ is written in the canonical form (2), and $m = |N|$, then:

(a) $\lceil km/n \rceil = \lfloor (k+1)\, m/n \rfloor$,

(b) if $n_1 = \lceil km/n \rceil$, then $\gcd(m, n_1) = d$,

(c) if $s : N \to N$ is defined by $s(i) = i'$ if (i, i') is an arc of D_u for some $u = 1, \dots, d$, then equation (3) holds, i.e., $s(i_j) = i_{j \underset{m}{\oplus} n_1}$ for $j = 0, \dots, m-1$.

Conversely,

Theorem 1.5 *Suppose the conditions (a) and (b) above are satisfied, $N \subset \mathbb{Z}_n$ is written in the canonical form (2), and $s : N \to N$ defined by the condition (3) satisfies*

$$(i, s(i)) \quad \textit{is an arc of } G(\mathscr{C}_n^k) \textit{ for all } i \in N. \tag{4}$$

Then there exist (uniquely determined) d disjoint simple directed cycles in $G(\mathscr{C}_n^k)$, $D_1, \dots, D_d$, all having the same length m/d, such that $N = \cup_u V(D_u)$.

If the previous conditions hold, then we have the following interlacing property of the cycles:

Lemma 1.6 *Suppose the assumptions of theorem 1.4 hold. Then in each cyclic interval of N (expressed as in (2)) of length d, there is exactly one point of each cycle.*

We now generalize theorem 1.2 to the case of several disjoint cycles.

Theorem 1.7 *Let n, k, m be given, with $1 \leq k \leq n-1$ and $0 \leq m \leq n-1$.*

There exist d disjoint cycles in $G(\mathscr{C}_n^k)$ each of length m/d if and only if equation (1) *holds and* $\gcd(m, n_1) = d$*, where n_1 is the common value.*

Lemma 1.1 relates the existence of a simple cycle D to the condition $\gcd(n_1, m) = 1$ and contractions $\mathscr{C}_n^k/V(D)$, and we have seen that existence of several disjoint cycles is related to some algebraic conditions. The following result shows that contractions are also related.

Theorem 1.8 *Suppose $n, k, m, n', k' \in \mathbb{N}$ and $N \subset \mathbb{Z}_n$ are given, such that $2 \leq k \leq n - 2$, $m = |N|$, $1 \leq m \leq n - 2$, and $1 \leq k' < n'$. Then the following are equivalent:*

(a) $\mathscr{C}_n^k/N \sim \mathscr{C}_{n'}^{k'}$.

(b) $|E(\mathscr{C}_n^k/N)| = |V(\mathscr{C}_n^k/N)|$.

(c) There exist d disjoint simple directed cycles of $G(\mathscr{C}_n^k)$, $D_1, D_2, \ldots, D_d$, having the same length, such that $N = \cup_u V(D_u)$.

From a geometric point of view, given a clutter $\mathscr{C}$ and a subset $N \subset V(\mathscr{C})$, we may interpret $P(\mathscr{C}/N)$ as the intersection of $P(\mathscr{C})$ with the subspace $\{x \in \mathbb{R}^n : x_i = 0 \ \forall\, i \in N\}$. Since the conditions $x_i \geq 0$ for all $i \in N$, are some of the inequalities defining $P(\mathscr{C}/N)$, vertices in $P(\mathscr{C}/N)$ (considered as subset of $\mathbb{R}^n$) are already vertices of $P(\mathscr{C})$. Thus, if for $x \in \mathbb{R}^n$ we let $N(x) = \{i \in \mathbb{Z}_n : x_i = 0\}$ and $m(x) = |N(x)|$, we see that a vertex x of $P(\mathscr{C}_n^k)$ with $N(x) \neq \emptyset$ will have a corresponding vertex in $P(\mathscr{C}_n^k/N(x))$ (regarded now as a subset of $\mathbb{R}^{n-m(x)}$), all of whose coordinates are positive. The remarkable fact, as shown by Argiroffo and Bianchi [3], is that for all vertices $x \in P(\mathscr{C}_n^k)$, $N(x)$ is such that $\mathscr{C}_n^k/N(x)$ is a circulant clutter. The following is a variant of their result:

Theorem 1.9 *The point x is a vertex of $P(\mathscr{C}_n^k)$ if and only if there exist n' and k', such that $1 \leq k' < n'$, $\mathscr{C}_n^k/N(x) \sim \mathscr{C}_{n'}^{k'}$,* $\gcd(n', k') = 1$*, and $x_i = 1/k'$ for all $i \notin N(x)$. (Here we allow $N(x) = \emptyset$.)*

Using our previous results, we obtain alternative characterizations of the vertices of $P(\mathscr{C}_n^k)$:

Theorem 1.10 *Suppose n and k are given, with $1 \leq k < n$. For $x \in \mathbb{R}^n$, let $N(x)$ be written in the canonical form* (2) *(if $N(x) \neq \emptyset$), and let $m = |N(x)|$.*

Then, x is a vertex of $P(\mathscr{C}_n^k)$ if and only if the following conditions hold:

(i) $m \leq n - 2$, and the equality (1) *is satisfied,*

(ii) if n_1 is the common value in the equality (1)*, then $n_1 < k$ and* $\gcd(n - m, k - n_1) = 1$,

(iii) $x_i = 1/(k - n_1)$ *for all* $i \notin N(x)$, *and*

(iv) *if* $m > 0$ *and* $d = \gcd(m, n_1)$, *then there exist* d *disjoint simple directed cycles in* $G(\mathscr{C}_n^k)$, $D_1, \ldots, D_d$, *all of length* m/d, *and such that* $N(x) = \cup_u V(D_u)$.

Alternatively, we could change the condition (iv) to:

(iv') *if* $m > 0$, *and* s *is defined by equation* (3), *then equation* (4) *holds.*

Corollary 1.11 *Let* n, k *and* m *be given non negative integers, such that* $1 \leq k < n$ *and* $1 \leq m < n - 1$. *Then,* $P(\mathscr{C}_n^k)$ *has a vertex with exactly* m *zero coordinates and the remaining coordinates taking the value* $1/k'$ *if and only if (i) the equation* (1) *holds, (ii) if* n_1 *is its common value, then* $k' = k - n_1$ *is positive, and (iii)* $\gcd(n - m, k') = 1$.

Our results can be used to study many families of circulant clutters. For example, the following is a characterization of ideal and mni circulant clutters in purely arithmetical terms:

Proposition 1.12 *If* $n \geq 3$ *and* $1 \leq k \leq n-1$, *then* $\mathscr{C}_n^k$ *is ideal or mni if and only if for every* m, $1 \leq m \leq n-2$, *for which (i)* $\lceil km/n \rceil = \lfloor (k+1)\, m/n \rfloor$, *and (ii) if* $n_1 = \lceil km/n \rceil$ *and* $\gcd(n - m, k - n_1) = 1$, *then necessarily* $n_1 = k - 1$.

If these conditions are satisfied, then $\mathscr{C}_n^k$ *is mni if* $\gcd(n, k) = 1$, *and otherwise is ideal.*

As already mentioned, Cornuéjols and Novick [5] gave a complete description of all the ideal and mni circulant clutters. Using properties of the Farey series, we may obtain the same results, without using Lehman's theorems.

There have been many efforts to introduce and study more general classes of clutters encompassing ideal and mni clutters. Of interest to us are near-ideal clutters, introduced by Argiroffo in her Ph.D. thesis [2] (see also [4]). Near-ideal circulant clutters may be defined as those for which $P(\mathscr{C}_n^k) \cap \{x \in \mathbb{R}^n : \mathbf{1} \cdot x \geq \lceil n/k \rceil\}$ is the convex hull of the 0-1 vertices of $P(\mathscr{C}_n^k)$.

We have:

Proposition 1.13 *Suppose* n *and* k *are given,* $n > k \geq 3$. *Then,*

(i) $\mathscr{C}_n^k$ *is not near-ideal if and only if there exist* n' *and* k' *such that*

$$k > k' > 1, \quad \gcd(k', n') = 1, \quad \frac{n'}{k'} > \left\lceil \frac{n}{k} \right\rceil, \quad \frac{n}{k+1} \geq \frac{n'}{k'+1}.$$

(ii) *For any* $\nu \geq 2$, $\mathscr{C}_{\nu k}^k$ *is not near-ideal except for* $\mathscr{C}_6^3$, $\mathscr{C}_9^3$ *and* $\mathscr{C}_8^4$, *which are ideal.*

(iii) *If $k \geq \frac{2}{3}n - 1$, then $\mathscr{C}_n^k$ is near-ideal.*

(iv) *If $k \geq 3$ and $n \geq 13k$, then $\mathscr{C}_n^k$ is not near-ideal.*

(ii) was observed in [3]. A result very similar to (iii), with the bound $k \geq \lfloor 2n/3 \rfloor$, was obtained by Argiroffo [2], using techniques involving blockers.

It is rather simple to construct a table to show the values of n and k for which $\mathscr{C}_n^k$ is near-ideal. Examining such a table reveals that we cannot hope for an exhaustive classification of near-ideal circulant clutters, similar to that given by Cornuéjols and Novick for ideal and mni circulant clutters.

As a final application, we show that the 0-1 vertices of $P(\mathscr{C}_n^k)$ always have fewer non zero coordinates than fractional vertices, and the number of non zero coordinates of 0-1 vertices are consecutive:

Proposition 1.14 *Suppose x is a 0-1 vertex of $P(\mathscr{C}_n^k)$, with $|N(x)| = m$, and x' is another vertex, not necessarily 0-1, with $|N(x')| = m'$. Then,*

(i) If x' is a fractional vertex, then $m' < m$.

(ii) If x' is 0-1 and $m < m'$, then, for any $m'' \in \mathbb{N}$ with $m < m'' < m'$, there exists a 0-1 vertex, x'', of $P(\mathscr{C}_n^k)$ with $|N(x'')| = m''$.

References

[1] N. Aguilera: Notes on “Ideal 0, 1 Matrices” by Cornuéjols and Novick. Preprint (2007). http://math.unl.edu.ar/~aguilera/papers/onideal01.pdf

[2] G. Argiroffo: Clasificación de clutters no ideales, Ph.D. Thesis, Universidad Nacional de Rosario, Argentina (2005).

[3] G. Argiroffo, S. Bianchi: On the set covering polyhedron of circulant matrices: characterization of all fractional extreme points. Preprint (2006).

[4] G. Argiroffo, S. Bianchi, G. Nasini: On a certain class of nonideal clutters. *Discrete Applied Mathematics* **254** (2006) 1854–1864.

[5] G. Cornuéjols, B. Novick: Ideal 0, 1 Matrices. *Journal of Combinatorial Theory, Series B* **60** (1994) 145–157.

[6] A. Lehman: On the Width-Length Inequality. *Mathematical Programming* **17** (1979) 403–417.

[7] A. Lehman: On the Width-Length Inequality and Degenerate Projective Planes. In *Polyhedral Combinatorics* (W. Cook and P. D. Seymour, eds.), *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **1**, American Mathematical Society (1990) 101–105.

Even-hole-free planar graphs have bounded treewidth

Aline Alves da Silva[1]

Departamento de Computação
Universidade Federal do Ceará
Fortaleza, Brazil

Ana Silva[1]

Departamento de Computação
Universidade Federal do Ceará
Fortaleza, Brazil

Cláudia Linhares Sales[1]

Departamento de Computação
Universidade Federal do Ceará, Brazil
MASCOTTE Project/INRIA, France

Abstract

The class of planar graphs has unbounded treewidth, since the $k \times k$ grid, $\forall k \in \mathbb{N}$, is planar and has treewidth k. So, it is of interest to determine subclasses of planar graphs which have bounded treewidth. In this paper, we show that if G is an even-hole-free planar graph, then it does not contain a 9×9 grid minor. As a result, we have that even-hole-free planar graphs have treewidth at most 44.

Keywords: Planar graphs, even-hole-free graphs, treewidth.

1 Introduction

The definitions of tree decomposition and treewidth were introduced by Robertson and Seymour in their series of papers on graph minors, published during the nineties. It is known that many $\mathcal{NP}$-hard problems can be polynomially solved if a tree decomposition of bounded treewidth is given. So, it is of interest to bound the treewidth of certain classes of graphs. In this context, the planar graphs seem to be specially challenging because, in despite of having many known bounded metrics (for example, maximum clique and chromatic number at most four [4],[5]), they have unbounded treewidth (one only need to notice that the $k \times k$ grid is a planar graph and has treewidth k, for all $k \in \mathbb{N}$ [2]). So, an alternative approach is to restrict ourselves to a subclass of planar graphs. In this paper, we investigate the class of even-hole-free planar graphs (from now on, we will refer to this class as Γ). In [9], we show that a graph in Γ does not contain a 10×10 grid subdivision. In this paper, we prove a stronger fact: a graph in Γ does not contain a 9×9 grid minor. In [6], Robertson et al. prove that if a planar graph G does not have a $k \times k$ grid minor, then G has treewidth at most $6k - 5$. The same authors improve their result by proving that if G, planar, does not have a $k \times k$ grid minor, then G has treewidth at most $5k - 1$[7]. This last result, together with ours, imply that a graph of Γ has treewidth at most 44.

2 Preliminaries

Let $G = (V, E)$ be a graph. We say that G *contains* H if H is a subgraph of G, and that G is H*-free* if H is not an induced subgraph of G. A *hole* in G is an induced cycle of size at least four. If a hole has even size, we say that it is an *even-hole*. Let $A, B \subseteq V$. We denote by $N^A(B)$ the set $\{u \in A \setminus B : u \in N(v),\ for\ some\ v \in B\}$, and call it the *neighborhood of* B *in* A. The $k \times l$ *grid* is the graph $G_{k \times l} = (V, E)$, where $V = \{v_{i,j} : 1 \leq i \leq k, 1 \leq j \leq l, i, j \in \mathbb{N}\}$ and $E = \{(v_{i,j}, v_{i',j'}) : |i - i'| + |j - j'| = 1\}$. Let (x, y) be an edge in G. The graph G_{xy} obtained from G by the *contraction* of (x, y) is such that $V(G_{xy}) = V(G) \setminus \{x, y\} \cup \{x * y\}$, and $E(G_{xy}) = E(G) \setminus \{(x, y)\} \cup \{(x * y, z) : z \in N^G(x) \cup N^G(y)\}$, where $x * y$ is the vertex obtained by the identification of x and y. We say that a graph H is a *minor* of G if it can be obtained by a sequence of vertex or edge deletions, or edge contractions.

A *tree decomposition* of G is a pair $\langle \{X_i | i \in I\}, T \rangle$, where each X_i is a

[1] The authors are partially supported by CNPq/Brazil. Email: {aline, shirley, linhares}@lia.ufc.br

subset of V and T is a tree whose nodes are the elements of I. Furthermore, the following three properties must hold: $\bigcup_{i \in I} X_i = V$; for every edge $(u, v) \in E$, there exists $i \in I$ such that $\{u, v\} \subseteq X_i$; and for all $i, j, k \in I$, if j lies on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$. The width of $\langle \{X_i | i \in I\}, T \rangle$ is equal to $max\{|X_i| \mid i \in I\} - 1$. The *treewidth* of G, $tw(G)$, is the minimum k such that G admits a tree decomposition of width k. The following theorem is the main result presented in this paper:

Theorem 2.1 *If $G \in \Gamma$, then G has no 9×9 grid minor.*

Theorem 2.2, together with Theorem 2.1, leads to an upper bound for the treewidth of even-hole-free planar graphs, as stated by Corollary 2.3.

Theorem 2.2 ([6],[7]) *If a planar graph G contains no $G_{k \times k}$ minor, then $tw(G) \leq 5k - 1$.*

Corollary 2.3 *If $G \in \Gamma$, then $tw(G) \leq 44$.*

The proof of Theorem 2.1 will be sketched in the next section. We finish this section by introducing some notation and definitions that will be used there. Let G be any graph and let $H \subseteq G$ be a minimal induced subgraph of G that contains a $G_{k \times l}$ minor, $k, l \in \mathbb{N}$. We say that H is a *model* of $G_{k \times l}$ in G. Note that, since H is minimal, $G_{k \times l}$ can be obtained from H only by edge contractions or deletions. Moreover, observe that each vertex $v_{i,j}$ of $G_{k \times l}$ is originated by a set of vertices, denoted by $V_{i,j}$, that induces a connected subgraph of H. We say such a set is a *node* and that the nodes $V_{i,j}$ and $V_{p,q}$ are *adjacent* if $v_{i,j}$ and $v_{p,q}$ are adjacent vertices in $G_{k \times l}$. Note that if $V_{i,j}$ and $V_{p,q}$ are adjacent, then there must be at least one edge between them in H; however, if there is such an edge, they are not necessarily adjacent. Finally, let $V_{i,j}$ be a node. If $1 < i < k$ and $1 < j < l$, we say that $V_{i,j}$ is an *internal node*.

Given three induced paths, P_1, P_2 and P_3, we say that they are *fittable* if there are no chords between them, except if they intersect in their extremities. A *dot-dot* structure consists of two vertices and three fittable paths connecting them of length at least two. A *triangle-triangle* structure consists of two K_3's, $\langle x, y, z \rangle$ and $\langle x', y', z' \rangle$, and three fittable paths, P_x, P_y and P_z, connecting x to x', y to y', and z to z', respectively. In addition, at most one of P_x, P_y and P_z has length one. Since at least two of the paths have the same parity, we have that if G is even-hole-free, then it does not contain a dot-dot structure neither a triangle-triangle structure.

3 Forbidden $G_{9\times 9}$ minor

In the following, a *forbidden structure* is a dot-dot structure or triangle-triangle structure. Let G be any graph of Γ and let H be a model of $G_{k\times l}$, $k, l \in \mathbb{N}$. Next, we present lemmas that analyze the internal structure of a model.

Lemma 3.1 *For every node $V_{i,j}$ of H, $G[V_{i,j}]$ is either a tree or its induced cycles are of length at most three.*

Actually, the structure of a node $V_{i,j}$ of H is more accurately represented in Figure 1. The "upper" (or "lower") path represented in the figure may not exist. Also, the paths might be connected by an edge, instead of triangles.

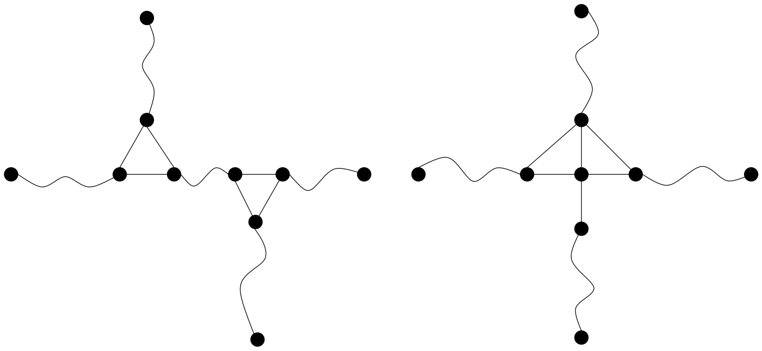


Fig. 1. Structure of a node.

Note that H may admit more than one partition whose contraction of the $V_{i,j}$'s lead to a $G_{k\times l}$ grid. The next lemma guarantees that always exists a partition satisfying some requirements.

Lemma 3.2 *Let $V_{i,j}$ be an internal node. If G has a model of a $G_{k\times l}$ grid, then G has a model of $G_{k\times l}$ such that there is a unique edge connecting $V_{i,j}$ to $V_{i,j+1}$, $V_{i-1,j}$ and $V_{i+1,j}$.*

We say that $H' \subset H$ is a (p, q)*-internal submodel* of H if it is a $G_{p\times q}$ model that uses consecutive "rows" and "columns" of H and contains only internal nodes of H. We denote by $r(H')$ and $c(H')$ the indices of the row and column of the left uppermost node of H', respectively. As we have said before, we will prove that a $G_{9\times 9}$ model has a forbidden structure. The following lemma gives a hint on how to pick the vertices or triangles of the forbidden structure.

Lemma 3.3 *Let H' be a $(3, 2)$-internal submodel of H. Set $i = r(H')$ and $j = c(H')$. Then H' contains a vertex (or a triangle) and three fittable paths con-*

necting this vertex (or triangle) to vertices $u \in N^{V_{i,j}}(V_{i-1,j})$, $b \in N^{V_{i+2,j}}(V_{i+3,j})$ and $r \in N^{V_{i+1,j+1}}(V_{i+1,j+2})$ of H'.

The following theorem implies Theorem 2.1.

Theorem 3.4 *Let H be a $G_{9\times 9}$ model. Then, H contains a forbidden structure.*

The proof of Theorem 3.4 basically consists of taking three $(3,2)$-internal submodels in the $G_{9\times 9}$ model, away from each other, and a substructure as described in Lemma 3.3. As at least two of them are of the same "type" and, if there is no chords between them, we can easily pick fittable paths in the model in order to construct a dot-dot or a triangle-triangle structure. Figure 2 shows a choice of $(3,2)$-internal submodels and paths. The vertices represent the nodes of H and the edges represent the edges between adjacent nodes. The $(3,2)$-internal submodels are in grey and the paths are in black. Note that the paths in Figure 2 do not have chords to the substructures in the $(3,2)$-internal submodels, except in their extremities. In addition, the paths are not necessarily induced. However, by using planarity arguments and the removal of vertices when a chord is detected, we can easily obtain induced paths whose union produces the desired forbidden structure.

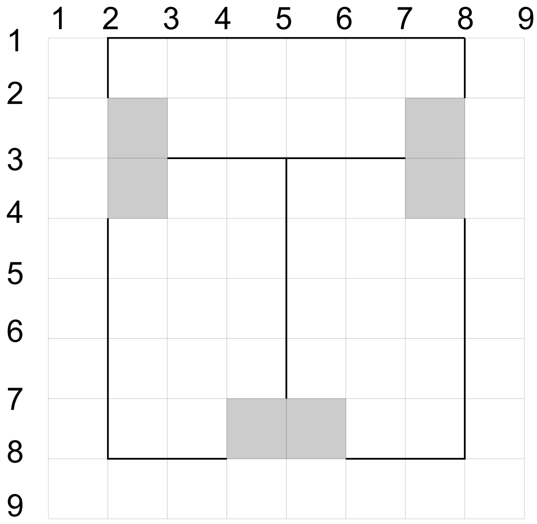


Fig. 2. $G_{9\times 9}$ model.

4 Conclusion

In this paper, we have found an upper bound for the treewidth of the graphs of Γ. With respect to lower bounds, a graph of Γ having a 3 × 3-grid minor exists [8]. The manual attempt to design graphs of Γ with bigger grid minors revealed the hardness of this task. Perhaps, some computation effort should be employed on this. In [8], two polynomial, non-exact algorithms to compute a tree decomposition of a graph $G \in \Gamma$ are given, both based on known characterizations of even-hole-free graphs ([1],[3]). In the first one, a tree decomposition is built from basic graphs by concatenating the tree decomposition of small pieces via the clique, k-stars ($k = 1, 2, 3$) and 2-join cutsets. In the second one, a tree decomposition is built by including one by one the vertices of G, following their bi-simplicial order. In the first case, if G has no 2-join cutset, the algorithm returns a tree decomposition of width at most $4 \cdot (h(\mathcal{T}) + 1)$ where $\mathcal{T}$ is the decomposition tree given by the Decomposition Theorem of [3] and $h(\mathcal{T})$ is the height of $\mathcal{T}$. In the second one, if G admits a bi-simplicial order $\langle v_n, \ldots, v_1 \rangle$ such that for every v_i, the two cliques of $N(v_i)$, if they exist, belong to distinct components of $G[v_{i-1}, \ldots, v_1]$, then using this order, the algorithm returns an optimal tree decomposition of G.

References

[1] Addario-Berry, L., M. Chudnovsky, F. Havet, B. Reed and P. Seymour, *Bisimplicial vertices in even-hole-free graphs* (2006), submitted to J. of Comb. Theory, Series B.

[2] Biró, C., *Treewidth and grids*, Technical report, School of Mathematics, Georgia Institute of Technology (2005).

[3] Conforti, M., G. Cornuéjols, A. Kapoor and K. Vuskovic, *Even-hole-free graphs part i: Decomposition theorem*, J. Graph Theory **39** (2002), pp. 6–49.

[4] Kuratowski, C., *Sur le problème des courbes gauches en topologie*, Fund. Math. **15** (1930), pp. 271–283.

[5] Robertson, N., D. Sanders, P. Seymour and R. Thomas, *The four-colour theorem*, J. of Comb. Theory, Series B. **70** (1997), pp. 2–44.

[6] Robertson, N., P. Seymour and R. Thomas, *Quickly excluding a planar graph*, J. of Comb. Theory, Series B **62** (1994), pp. 323–348.

[7] Robertson, N., P. Seymour and R. Thomas, *Tree decompositions of graphs*, www.math.gatech.edu/∼thomas/SLIDE/CBMS/trdec.pdf (2000).

[8] Silva, A., “Decomposição e Largura em Árvore de Grafos Planares Livres de Ciclos Pares Induzidos,” Master’s thesis, Universidade Federal do Ceará (2007).

[9] Silva, A., A. Silva and C. Linhares Sales, *Largura em Árvore de grafos planares livres de ciclos pares induzidos*, Anais do 39o. SBPO (2007), to appear.

List Colouring Constants of Triangle Free Graphs

Omid Amini [1,2]

École Polytechnique &
Projet Mascotte I3S-CNRS/UNSA/INRIA
Sophia-Antipolis France

Bruce Reed [1,2]

Projet Mascotte I3S-CNRS/UNSA/INRIA
Sophia-Antipolis France &
Canada Research Chair in Graph Theory
Department of Computer Science McGill University

Abstract

In this paper we prove a result about vertex list colourings which in particular shows that a conjecture of the second author (1999, Journal of Graph Theory **31**, 149-153) is true for triangle free graphs of large maximum degree. There exists a constant K such that the following holds: Given a graph G and a list assignment L to vertices of G, assigning a list of available colours $L(v)$ to each vertex $v \in V(G)$, such that $|L(v)| = \frac{K\Delta}{\log(\Delta)}$, then there exists a proper list colouring of vertices of G provided that for each colour c, the graph induced by all vertices v with $c \in L(v)$ is triangle free and has maximum degree at most Δ.

Keywords: Graph colouring, List colouring, Probabilistic method, Randomised algorithms.

1 Introduction

Let $G = (V, E)$ be a graph. A *list assignment* to vertices of G associates a list $L(v)$ of available colours to each vertex $v \in V$. A *proper L-colouring* of G is a proper colouring such that the colour of each vertex v belongs to $L(v)$. Given a colour c in $L = \bigcup_{v \in V} L(v)$, we denote by G_c the graph induced on all vertices v with $c \in L(v)$, i.e. $G_c = G[\{v \in V \mid c \in L(v)\}]$.

We denote by Δ the integer $\max_{c \in L} \Delta(G_c)$, and we write $\ell(v) = |L(v)|$. It was conjectured by Reed in [5] that if $\ell(v) \geq \Delta + 1$, then G admits a proper L-colouring. Bohman and Holzman [1] provided a counter-example to this conjecture. Reed [5] and Haxell [2] proved sufficient linear bounds of type, respectively, $\ell(v) \geq 2e\Delta$ and $|L(v)| \geq 2\Delta$ for the existence of a proper L-colouring. Furthermore Reed and Sudakov [6] proved that the conjecture is asymptotically true. An intriguing open question is the following

Question 1 Is there any constant C such that for every G and L as above with the extra condition that $|L(v)| \geq \Delta + C$, G admits a proper L-colouring?

The work of this paper is motivated by the above question restricted to triangle free graphs. We prove the following theorem which in particular provides an answer to the above question for triangle free graphs.

Theorem 1 Let G be a graph and L be a list assignment as above. Suppose that each subgraph G_c is triangle free. There exists an absolute constant K such that if $|L(v)| \geq \frac{K\Delta}{\log \Delta}$ then G admits a proper L-colouring.

The general idea of the proof comes from Johansson's proof [3] that triangle free graphs have list chromatic number at most $\mathcal{O}(\frac{\Delta}{\log(\Delta)})$. It uses a randomised greedy algorithm [3], known as *semi-random method*, which at each step consists of, a) colouring a small set of vertices, b) removing the colour of these vertices from the lists of all neighbours, and c) removing all the coloured vertices. The main point of this approach is the key observation that with a good choice of parameters the size of lists shrinks more slowly than the maximum degree of the colour classes G_c and so after a sufficient number of iterations, each list should contain a number of colours at least twice the maximum degree of colour classes. At this stage, one can finish the colouring using Haxell's deterministic theorem [2]. The analysis of this algorithm uses classical concentration tools but also needs the polynomial method of [4] and is similar to Vu's approach in [7].

[1] Email: `omid.amini@sophia.fr,breed@cs.mcgill.ca`

[2] This research has been supported by the European project AEOLUS.

2 Proof of Theorem 1: Main ideas

An *L-distribution* is the following data: for each vertex v and each $c \in L(v)$, we are given a non-negative weight $p(v,c) < 1$. Given an L-distribution, by $C(v)$ we denote the total weight of colours at v, i.e. $C(v) = \sum_{c \in L(v)} p(v,c)$. For a colour $c \notin L(v)$, we understand implicitly that $p(v,c) = 0$. An L-distribution is called *normalised* if in addition we have $C(v) = 1$ for all $v \in V(G)$. An L-distribution proposes a natural way for generating a set of colours at a given vertex v: *For each colour $c \in L(c)$, we conduct a coin flip with probability of win equal to $p(v,c)$ and generate c if we win. So with probability $p(v,c)$, c is among the generated colours for v.* By $x_{v,u}$ we will denote the expected number of common generated colours for v and u. We have

$$x_{v,u} = \sum_{c \in L(v) \cap L(u)} p(v,c)p(u,c)$$

On the other hand, applying a simple Chebyshev inequality, we have

$$Prob(\text{ there exists at least one colour generated for both } v \textit{ and } u) \ \leq x_{v,u}.$$

To prove Theorem 1 we use the *Generalised Wastefull Colouring Procedure (GWCP)*. The greedy algorithm then consists of several iterations, let us say T, of the proposed procedure *GWCP*. We suppose that two global fixed parameters α and α_* are also given, see later for a possible choice. At each iteration, *GWCP* takes as an input a subset $V_i \subset V(G)$ (the set of uncoloured vertices), a list assignment L_i, an L_i-distribution $\{p_i(v,c)\}$ and a subset $F_i(v) \subset L_i(v)$ for each vertex v (the set of *forbidden* colours for v). It then provides as output a subset $V_{i+1} \subset V_i$, a new list assignment $L_{i+1} \subset L_i$, an updated L_{i+1}-distribution $\{p_{i+1}(v,c)\}$, and a new subset $F_{i+1}(v)$ of forbidden colours ($F_i(v) \subset F_{i+1}(v)$). The general scheme of the procedure at the beginning of the ith iteration is described below:

Generalised Wastefull Colouring Procedure:

(i) **Generating colours** For each vertex $v \in V_i$ and each colour c, choose the colour c for v with probability $\alpha.p_i(v,c)$. Let $\mathcal{C}_v$ be the set of generated colours for v.

(ii) **Updating the lists**: For each vertex v and each generated colour $c \in \mathcal{C}_v$, remove c from the lists of all neighbours u of v for which c is not forbidden (for $c \notin F_{i-1}(u)$). The new lists will be called L_{i+1}.

(iii) **Colouring some vertices:** If $\mathcal{C}_v \cap (L_{i+1}(v) \setminus F_i(v)) \neq \emptyset$, give a colour to v from this subset. Define V_{i+1} to be the set of still uncoloured vertices after this step.

(iv) **Updating the probabilities and the new sets of forbidden colours**: The way we update the weights of colours, i.e. the definition of $p_{i+1}(v,c)$ and so $F_{i+1}(v)$, is described below.

Let us first define some more notations: $L'_i := L_i \setminus F_i$ is the set of *available* colours at v at step i. $\ell_i(v)$ will denote the cardinality of $L'_i(v)$. $G_{i,c}$ is the graph induced on vertices v with $c \in L'_i(v)$. For a vertex v with a colour $c \in L'_i(v)$, by $d_{i,c}(v)$ we will represent the degree of v in $G_{i,c}$. We initialise $d_{1,c}(v) = \Delta$ and $\ell_1(v) = \frac{K\Delta}{\log\Delta}$ for some large constant K.

We now describe the way we update the L_i-distributions and the sets F_i: At the beginning, we define the distribution $\{p_1(v,c)\}$ to be the uniform normalised distribution on $L(v)$, i.e.: $p_1(v,c) = \frac{1}{\ell_1}$ if $c \in L_1(v)$ and is zero otherwise. Let us define $Keep_i(v,c)$ to be the probability that v keeps the colour c after the step **updating the lists** of the ith iteration, i.e. $Keep_i(v,c) = Prob(\ c \in L_{i+1}(v))$.

For a given vertex v and given colour $c \in L_{i+1}(v)$, we will define $p_{i+1}(v,c)$ essentially as follows:

- If $\frac{p_i(v,c)}{Keep_i(v,c)} \leq \alpha_*$, define $p_{i+1}(v,c) = \frac{p_i(v,c)}{Keep_i(v,c)}$;
- If $c \in F_i(v,c)$, then we set $p_{i+1}(v,c) = \alpha_*$;
- Otherwise, $\frac{p_i(v,c)}{Keep_i(v,c)} > \alpha_* > p_i(v,c)$. If c is not assigned to any neighbour of v during the ith iteration, we set $p_{i+1}(v,c) = \alpha_*$. If c is assigned to a neighbour of v, then conduct a coin flip with probability of win equal to $\frac{\frac{p_i(v,c)}{\alpha_*} - Keep_i(v,c)}{1 - Keep_i(v,c)}$ and define $p_{i+1}(v,c) = \alpha_*$ if we win and $p_{i+1}(v,c) = 0$ otherwise.

It is not difficult to see that with this definition we have $\mathbb{E}(p_{i+1}(v,c)) = p_i(v,c)$. Now define $F_{i+1} = \{c \mid p_{i+1}(v,c) = \alpha_*\}$. We have $F_i(v) \subset F_{i+1}(v)$. Note that $\{p_i(v,c)\}$ does not remain necessary a normalised distribution on $L_i(v)$. Remark that by the linearity of expectations we also have $\mathbb{E}(C_{i+1}(v)) = C_i(v)$.

The first remark is that the variable C_i is highly concentrated around its expected value, which implies that $C_i(v) \sim 1$. How about the random variables $x_{u,v}$ at the $(i+1)$th iteration, which we denote by $x_{i+1,u,v}$? How do their value evolve[3]? Remember that $x_{i+1,u,v} = \sum_c p_{i+1}(u,c)p_{i+1}(v,c)$.

As all the graphs G_c are triangle free, it follows that the variables $p_{i+1}(u,c)$ and $p_{i+1}(v,c)$ are independent and so we have $\mathbb{E}(x_{i+1,u,v}) = \sum_c p_i(u,c)p_i(v,c) =$

[3] Remark that in the ith iteration, $x_{i+1,u,v}$ is a random variable which depends on the random choices we make at this step, while we have already from previous iterations the values of $p_i(v,c)$ and so $x_{i,u,v}$.

$x_{i,u,v}$. It turns out that these variables are also highly concentrated around their expected value, which roughly permits us to conclude that with positive probability we can ensure $x_{i,u,v} \sim x_{1,u,v} = \sum_c p_1(u,c)p_1(v,c) = \frac{|L_u \cap L_v|}{\ell_1^2}$. To describe the behaviour of lists sizes and degrees through the time, it is more comfortable to introduce some new variables: for each i and each $v \in V_{i-1}$, we define a new random variable $t_{i,v}$ as follows:

$$t_{i,v} = \begin{cases} 1 \text{ if } v \text{ remains uncoloured after the } (i-1)\text{th iteration, i.e. } v \in V_i; \\ 0 \text{ if } v \text{ gets a colour, i.e. } v \notin V_i. \end{cases}$$

It is clear[4] that

$$(1) \qquad d_{i,c}(v) \leq \sum_{u \in N_{G_{i-1,c}(v)}} t_{i,u}.$$

For a given vertex v, let us define the random variable $x_{i,v}$ as follows:

$$(2) \qquad x_{i,v} = \sum_{u \in N_{G_{i-1}}(v)} x_{i,v,u} t_{i,u}.$$

The random variable $x_{i,v}$ has a simple interpretation: $x_{i,v}$ counts the expected number of common colours between v and one of its neighbours in G_i with respect to the distribution L_i. The two above equations are very similar: Equation 2 can be seen as a weighted version of Equation 1. We can see that it is possible to apply the polynomial method of [4] to prove that both the variables are some how concentrated [5]. Remember $x_{i,u,v}$ is also highly concentrated and so $x_{i,u,v} \sim \frac{|L(u) \cap L(v)|}{\ell_1^2} \leq \frac{1}{\ell_1}$. Intuitively $\mathbb{E}(t_{i,u}) \sim 1 - \alpha$ which implies that $\mathbb{E}(x_{i,v}) \sim \sum_{u \in N_{G_{i-1}}(v)} x_{i-1,u,v}\mathbb{E}(t_{i,u}) \sim (1-\alpha)x_{i-1,v}$, and $\mathbb{E}(d_{i,c}(v)) \leq (1-\alpha)d_{i-1,c}(v)$.

To finish the outline of the proof, let us show how one can use the *entropy* function $H_i(v)$ to bound the size of lists. We remember that the entropy $H_i(v)$ is defined as

$$H_i(v) := -\sum_c p_i(v,c) \log(p_i(v,c)).$$

Remark that $e^{H_i(v)} = \prod_c \ p_i(v,c)^{-p_i(v,c)}$. It turns out $e^{H_i(v)}$ provides a lower bound for the number of colours in $L_i(v)$[6]. Writing the changes in entropy

[4] We have an inequality here because it is possible that u loses the colour c.

[5] We note that the classical concentration tools can not be applied directly here.

[6] An intuitive idea: because of the concentration phenomena we should have $C_i = \sum_c p_i(v,c) \sim 1$. With the extra hypothesis that the distribution $\{p_i(v,c)\}$ is almost uniform, i.e. $p_i(v,c) \sim \frac{1}{\ell_i}$, we infer that $e^{H_i(v)} = (\frac{1}{\ell_i})^{-\sum p_i(v,c)} \sim \ell_i$. One can see that the size of $F_i(v)$ is small enough and we can ignore it.

step by step, we can see that the random variables $x_{i,v}$ enter to the picture very naturally: indeed we have

$$H_i(v) - H_{i-1}(v) = -\alpha x_{i,v} - (p_i(v,c) - p_{i-1}(v,c)) \log(p_i(v,c)).$$

and so $H_i(v) - H_{i-1}(v) \sim -\alpha x_{i,v}$ which shows that

$$H_i(v) - H_{i-1}(v) \sim -\alpha(1-\alpha)^{i-1}\frac{\Delta}{\ell_1}.$$

Summing up over all i and using $\ell_1 = \frac{K\Delta}{\log(\Delta)}$ and $H_1(v) \sim \log(\Delta)$, we infer that $H_i(v) \geq \log(\Delta^{1-\frac{1}{K}})$. This proves that after T steps we (intuitively) have $\geq \Delta^{1-\frac{1}{K}}$ colours in each list $L_T(v)$. The size of F_T will be also small enough to ensure that $\ell_T(v) \geq \Delta^{1-\frac{1}{K}}/2$. On the other hand, the degree sequence will decrease by at least a multiplicative factor $(1-\alpha)$ and so after T steps, $d_{T,c}(v) \leq (1-\alpha)^T\Delta \sim e^{-\alpha T}\Delta$. If we choose T and α in such a way that $\alpha T \geq \frac{\log(\Delta)}{500}$, the above arguments implie $d_{T,c}(v) \leq e^{-\frac{\log(\Delta)}{500}}.\Delta = \Delta^{1-\frac{1}{500}}$. Now if $K = 1000$, we will have $l_T(v) \geq \Delta^{1-\frac{1}{K}}/2 \geq 2\Delta^{1-\frac{1}{500}} \geq 2.d_{T,c}(v)$. This finishes the outline of the proof. For the choice of α and α_* we can for example suppose $\alpha_* = \Delta^{-\frac{49}{50}}$, $\alpha = \Delta^{-\frac{1}{20}}$. The omitting details are postponed to the full version of this paper.

References

[1] T. Bohman and R. Holtzman, *On a List Colouring of Conjecture of Reed*, Journal of Graph Theory, **41**, No. 2, 106–109, 2002

[2] P. Haxell, *A Note on Vertex List Colouring*, Combinatorics, Probability and Computing, **10**, No. 4, 345–347, 2001

[3] A. Johansson, *Asymptotic Choice Number for Triangle Free Graphs*, DIMACS Technical Report 91-95, 1996.

[4] J. H. Kim and V. H. Vu, *Concentrations of Multivariate Polynomials and its Applications*, Combinatorica, **20**, No.3, 417–434, 2000.

[5] B. Reed, *The List Colouring Constants*, Journal of Graph Theory, **31**, 149–153, 1999

[6] B. Reed, and B. Sudakov, *Asymptotically the List Colouring Constants are 1*, Journal of Combinatorial Theory, Series B, **86**, 27–37, 2002.

[7] V. H. Vu, *A General Upper Bound on the List Chromatic Number of Locally Sparse Graphs*, Combinatorics, Probability and Computing, **11**, No. 1, 103-111, 2002.

The Proportional Colouring Problem: Optimizing Buffers in Wireless Mesh Networks

Florian Huc [1]

I3S (CNRS-UNSA), INRIA
Sophia Antipolis, France

Cláudia Linhares-Sales [1]

Universidade Federal do Ceará, Brazil
MASCOTTE Project/INRIA, France

Hervé Rivano [1]

I3S (CNRS-UNSA), INRIA
Sophia Antipolis, France

Abstract

In this paper, we consider a new edge colouring problem motivated by wireless mesh networks optimization: the proportional edge colouring problem. Given a graph G with positive weights associated to its edges, we want to find a proper edge colouring which assigns to each edge at least a proportion (given by its weight) of all the colours. If such colouring exists, we want to find one using the minimum number of colours. We proved that deciding if a weighted graph admits a proportional edge colouring is polynomial while determining its proportional edge chromatic number is NP-hard. We also give a lower and an upper bound that can be polynomially computed. We finally characterize some graphs and weighted graphs for which we can determine the proportional edge chromatic number.

Keywords: edge colouring, proportional colouring, wireless mesh network.

1 Introduction

Given a weighted graph (G, w) where w is a weight function from $E(G)$ to $\mathbb{R}^+$, several distinct colouring problems of G have been defined. In [2], one wants to colour the vertices of G while minimizing the sum of the weights of the edges whose extremities receive the same colour. In [5], one wants to colour the vertices of G so that for each edge uv, $|c(v) - c(u)| \geq w(uv)$, where $c(u)$ and $c(v)$ are the colours assigned to u and v. In this paper, we consider a proper edge colouring which assigns to each edge e at least a fraction (given by the weight of e) of all the colours used. By "proper", we mean that, for any two adjacent edges, the sets of assigned colours have an empty intersection. If such colouring exists, we want to find one using the minimum number of colours, number which we call proportional edge chromatic number.

In next section, we introduce the notations we will use throughout this paper and recall some results about edge colouring. Then we will present the telecommunication problem which is modelled by the proportional edge colouring problem, together with some complexity results and bounds for the proportional edge chromatic number. Finally, we characterize a class of graphs and a class of weighted graphs whose proportional edge chromatic number can be computed in polynomial time.

2 Preliminaries

Throughout the paper, (G, w) denotes a weighted simple graph where w is a positive function called *weight function* defined on the edges of G, $w : E(G) \rightarrow [0, 1]$. We denote by $\Delta(G)$ the maximum degree of G.

The classical edge colouring problem is to determine the edge chromatic number of a simple graph G, $\chi'(G)$, that is, the minimum integer k such that G admits a proper edge colouring using k colours. In 1964, Vizing proved that $\chi'(G)$ is at most $\Delta + 1$ [7]. Since it is at least Δ, we can classify every graph: a graph is in *Class 1* if its edge chromatic number is Δ and in *Class 2* otherwise. Surprisingly, deciding if a graph is Class 1 or 2 is hard [3], even for cubic graphs [4]. For the upper bound presented in this work, we recall a possible definition of the fractional edge chromatic number: $\chi'^* = \min_{k \geq 1} \frac{\chi'(G^k)}{k}$, where G^k is the graph obtained from G by replacing each edge by k parallel edges. Also recall that this parameter can be determined in polynomial time [6].

[1] C. Linhares-Sales is partially supported by CNPq/Brazil. F. Huc is supported by CNRS and région PACA. This work has been partially funded by European project IST/FET AEOLUS and ANR-JC OSERA. Email: Florian.Huc, Claudia.Linhares-Sales, Herve.Rivano@sophia.inria.fr

Since we only consider proportional edge colouring, *we will omit to precise edge* in the remainder of the paper.

3 Proportional colouring

The proportional colouring problem is motivated by the following telecommunication problem. We consider a slotted time division multiplexing wireless mesh network connecting routers through *directional* antennas. We denote by *call* two antennas communicating together. The network topology defines a graph $G = (V, E)$ whose vertices are the routers and the edges are the achievable calls. For the sake of radio interferences and near-far effect, each router can be involved in at most one call at a time. Therefore, a set of simultaneously achievable calls is a matching of G. More precisely, given a set of calls, the classical *timeslot assignment* problem consists in decomposing this set into a minimum number of subsets of simultaneously achievable calls, which is equivalent to the proper edge colouring problem [1]. Remark that without the assumption of directional antennas, this problem would be modelled by the induced matching problem.

The proportional colouring problem arises when we consider *Constant Bit Rates* (CBR) requests. We are given a set of communication requests, each request being a source-destination path in the network, and a bit rate. Sending this amount of data on the paths induces that each call has to be periodically activated in a given proportion of the time. This is modelled by a weight function $w : E(G) \rightarrow [0, 1]$.

The problem is now, if possible, to find a *periodical schedule* of the calls satisfying the CBR requests, that is, such that to each call a number of timeslots proportional to its weight is assigned to. Besides, the length of the period is proportional to the size of the buffer needed at each router, which is an important parameter of the cost of the network.

We therefore want a proper edge colouring of G that satisfies the proportions given by the weights of the edges, with the minimum number of colours. This is *the proportional colouring problem* formally defined as follows.

Definition 3.1 [proportional colouring] Given a weighted graph (G, w), a *proportional colouring* of (G, w) is a function $\mathcal{C} : E \rightarrow \mathcal{P}(\{1, \ldots, c\})$ such that for all $e \in E$, we have: $|\mathcal{C}(e)| \geq c \cdot w(e)$ and for all adjacent edges e, f, $\mathcal{C}(e) \cap \mathcal{C}(f) = \emptyset$. We call the proportional chromatic number of G, $\chi'_\pi(G, w)$, the minimum number of colours c for which a proportional colouring of G exists. If it does not exist then $\chi'_\pi(G, w) = \infty$.

Definition 3.2 [m-graph] Given a weighted graph (G, w) and an integer m,

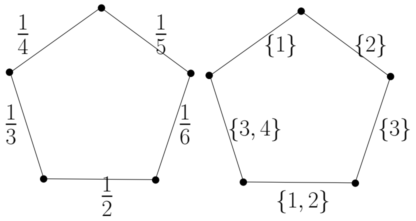


Fig. 1. A weighted C_5 and a proportional colouring using 4 colours

the m-graph G_m is constructed on vertex set $V(G)$ as follows: given an edge $e = (uv) \in E(G)$, we put $\lceil mw(e) \rceil$ multiple edges uv in $E(G_m)$.

Remark 3.3 Notice that given a weighted graph (G, w) and an integer m, if a colouring using km colours (where k is a constant) which gives at least k colours to all the edges of G_m exists, then this colouring can be easily transformed into a proportional colouring of (G, w).

Definition 3.4 [mcd(w)] Given a weighted graph (G, w) with w taking value in $\mathbb{Q}$, we set $mcd(w)$ as the minimum common denominator of all the values taken by w.

Figure 1 is an example of a weighted graph and its proportional colouring. The proportional colouring problem is divided into two subproblems. The first one consists in proving that there exists an integer c such that a c-proportional colouring of (G, w) exists. The second one is to determine the proportional chromatic number of (G, w). We start by giving simple facts which we will use to prove Theorem 3.6.

Fact 3.5 *Let (G, w) be a weighted graph.*

- *If there is a vertex u of G with $\sum_{uv \in E(G)} w(uv) > 1$ then $\chi'_\pi(G, w) = \infty$.*
- *If for all $uv \in E(G)$, $w(uv) \leq 1/(\Delta + 1)$ then $\chi'_\pi(G, w) \leq \Delta + 1$.*
- *Similarly, if for all $uv \in E(G)$, $w(uv) \leq 1/\chi'^*(G_{mcd(w)})$ then $\chi'_\pi(G, w) \leq q$ where q is the numerator of $\chi'^*(G_{mcd(w)})$.*

Theorem 3.6 *Let (G, w) be a weighted graph with w taking value in $\mathbb{Q}$.*

i) Determining if there exists a proportional colouring for (G, w) is polynomial.

ii) Determining the proportional chromatic number of (G, w) is NP-hard.

Proof.

i) Let (G, w) be any weighted graph with $w : E(G) \to [0, 1] \cap \mathbb{Q}$. Let $mcd := mcd(w)$. In the mcd-graph, we have $\chi'^*(G_{mcd}) \leq mcd \Leftrightarrow \chi'_\pi(G, w) < \infty$.

Indeed, suppose $\chi'_\pi(G,w) = k < \infty$. Then, there exists a proportional colouring using k colours. If we repeat this colouring *mcd* times, we still have a proportional colouring, even if it is not minimal. This colouring gives a proper edge colouring of G_{mcd} using $k \cdot mcd$ colours, in which each edge receives at least k colours. Therefore, $\chi'^*(G_{mcd}) \leq \frac{k \cdot mcd}{k}$. On the other hand, suppose that $\chi'^*(G_{mcd}) = \frac{q}{p}$ and consider this optimal fractional edge colouring of G_{mcd}. This colouring can be extended to an edge colouring of G that uses q colours and assigns $p \cdot mcd \cdot w(e)$ to each edge e. This colouring is proportional, since, by assumption, $mcd \cdot w(e) \cdot p \geq q \cdot w(e)$. Hence $\chi'_\pi(G,w) = k \leq q < \infty$.

ii) Given a graph G of maximum degree Δ, we set the weights of its edges to $\frac{1}{\Delta+1}$. Since G is $\Delta+1$ colourable, G is proportionally $\Delta+1$ colourable. Now, observe that if (G,w) admits a proper edge colouring with Δ colours, then it admits a proportional colouring with Δ colours. Since determining if G is Δ or $\Delta+1$ colourable is NP-hard [3], it is NP-hard to determine the proportional chromatic number of an instance (G,w). □

In despite of the difficulty of computing the proportional chromatic number of a weighted graph (G,w), we can deduce polynomially computable lower and upper bounds. Clearly, for a weighted graph (G,w), $\chi'_\pi(G,w) \geq \Delta(G)$. Theorem 3.7 improves this lower bound.

Theorem 3.7 (Lower bound) *Let m be the minimum integer satisfying for all $u \in V$, $\sum_{v \text{ } st \text{ } uv\in E} \lceil mw(uv) \rceil \leq m$. If no m satisfies all the above equations, then no proportional colouring exists. Otherwise, if there is a solution m and that (G,w) admits a proportional colouring, $\chi'_\pi(G,w)$ is at least m.*

Theorem 3.8 (Upper bound) *Let (G,w) be a weighted graph such that w takes value in $\mathbb{Q}$. If $G,w)$ admits a proportional colouring, then there is a proportional colouring of (G,w) using q colours, where q is the numerator of $\chi'^*(G_{mcd(w)})$.*

The proof of Theorem 3.8 follows from item (i) of Theorem 3.6. In general, given a weighted graph (G,w), the *mcd* of the values taken by w is not an upper bound. Indeed, consider $(P, \mathbf{\frac{1}{3}})$: the Petersen graph P with the constant weight function equal to $\frac{1}{3}$. Since $\chi'^*(P) = 3$, we have $\chi'_\pi(P, \mathbf{\frac{1}{3}}) < \infty$. However $\chi'_\pi(P, \mathbf{\frac{1}{3}}) > 3 = mcd(\mathbf{\frac{1}{3}})$, since P is not 3-colourable.

The proportional chromatic number may also be different from the upper bound given by Theorem 3.8, because an edge can receive proportionally more colours than it required. This is illustrated by Figure 1. Indeed, in this example, we have $\chi'^*(C_{5\ mcd(w)}) \geq 50 > 4 = \chi'_\pi(C_5, w)$.

One can also ask for which classes of graphs the proportional colouring problem can be polynomially solved. The next theorems announce two positive results, the first one giving a condition to the lower bound be reached.

Theorem 3.9 *Let (G, w) be a weighted bipartite graph. If there is a solution m to the set of equations given by Theorem 3.7, then (G, w) admits a proportional colouring using m colours. In fact, for any weighted graph (G, w), $\chi'_\pi(G, w) = m \Leftrightarrow \chi'(G_m) = m$.*

Theorem 3.10 *Let (G, w) be a weighted graph with w taking values in $\mathbb{Q}$. Let e be an edge with an end vertex v such that $\sum_{uv \in E} w(uv) = 1$. Then, the denominator of $w(e)$ divides $\chi'_\pi(G, w)$. In particular, if the proportional chromatic number of (G, w) is finite and every edge e has an end vertex v with $\sum_{uv \in E} w(uv) = 1$, then $\chi'_\pi(G, w) = mcd(w)$.*

Motivated by the applications modelled by the proportional colouring problem and its hardness, we pose the following general open questions: find approximation algorithms for the classes of graphs which usually occur in telecommunication, as circular-arc graphs and triangular lattices, and determine other classes of graphs (and weighted classes of graphs) for which the proportional chromatic number can be calculated in polynomial time.

References

[1] J. Bibb Cain, T. Billhartz, L. Foore *A link scheduling and ad hoc networking approach using directional antennas*, Military comm. conf., **1** Oct 2003, 643–648.

[2] Carlson, R.C. and Nemhauser, G.L., *Scheduling to minimize interaction cost*, Oper. Res. **1**4 (1966), 52–58.

[3] Holyer, I. *The NP-completeness of edge colourings*, SIAM J. Computing **1**0 (1981), 718–720.

[4] Koreas, D.P., *The NP-completeness of edge chromatic number in triangle free graphs with maximum vertex of degree 3*, App. Math. and Comp. **8**3 (1) (1997), 13–17.

[5] McDiarmid, C., *Discrete mathematics and radio channel assignment*, In: Recent advances in algorithms and combinatorics, eds. B. Reed and C. Linhares-Sales, Springer, New York, 2003, 27–63.

[6] G. L. Nemhauser, S. Park, *A polyhedral approach to edge colouring*, Operations Research Letters **10**, 315-322, 1991.

[7] Vizing, V.G., *On an estimate of a chromatic class of a p-graph* (In Russian), Diskret. Analiz. **3** (1964), 25–30.

On maximizing clique, clique-Helly and hereditary clique-Helly induced subgraphs

L. Alcón [a] L. Faria [b] C. M. H. de Figueiredo [b] M. Gutierrez [a]

[a] *U.N.L.P., CC 172, (1900) La Plata, Argentina*

[b] *U.F.R.J., Caixa Postal 68511, 21941-972, Rio de Janeiro, Brazil*

Abstract

Clique-Helly and hereditary clique-Helly graphs are polynomial-time recognizable. Recently, we presented a proof that the clique graph recognition problem is NP-complete [1]. In this work, we consider the decision problems: given a graph $G = (V, E)$ and an integer $k \geq 0$, we ask whether there exists a subset $V' \subseteq V$ with $|V'| \geq k$, such that the induced subgraph $G[V']$ of G is, respectively, a clique, clique-Helly or hereditary clique-Helly graph. The first problem is clearly NP-complete from [1]; we prove that the other two mentioned decision problems are NP-complete, even for maximum degree 6 planar graphs. We consider the corresponding maximization problems of finding a maximum induced subgraph that is, respectively, clique, clique-Helly or hereditary clique-Helly. We show that these problems are Max SNP-hard, even for maximum degree 6 graphs. We generalize these results for other graph classes. We exhibit a polynomial 6-approximation algorithm to minimize the number of vertices to be removed in order to obtain a hereditary clique-Helly subgraph.

1 Introduction

A *complete set* of a graph $G = (V, E)$ is a subset of V inducing a complete subgraph. A *clique* is a maximal complete set. Denote by $\mathcal{C}(G)$ the *clique family* of G. The *clique graph* $K(G)$ of G is the intersection graph of $\mathcal{C}(G)$. Say that G is a *clique graph* if there exists a graph H such that $G = K(H)$. A *clique-Helly* graph is a graph where $\mathcal{C}(G)$ satisfies the *Helly property*: any pairwise intersecting subfamily of $\mathcal{C}(G)$ has non empty total intersection [5]. A *hereditary clique-Helly* graph is a graph where every induced subgraph is clique-Helly. The class of hereditary clique-Helly graphs (H$\mathcal{K}$H) is contained in the class of clique-Helly graphs ($\mathcal{K}$H), which in turn is contained in the class of clique graphs ($\mathcal{K}$).

Clique graphs and subclasses have been much studied as intersection graphs, in the context of graph operators, and are included in several books [4,9,12,14].

Let $\mathcal{A}$ represent a class of graphs; consider the following problems:

$\mathcal{A}$-RECOGNITION ($\mathcal{A}$-REC)
Instance: Graph $G = (V, E)$.
Question: Does G belong to $\mathcal{A}$?

$\mathcal{A}$-SUBGRAPH ($\mathcal{A}$-SUB)
Instance: Graph $G = (V, E)$ and a positive integer k.
Question: Is there a subset $V' \subseteq V$ with $|V'| \geq k$, such that the subgraph $G[V']$ induced by the set V' belongs to $\mathcal{A}$?

MAXIMUM $\mathcal{A}$-SUBGRAPH (MAX-$\mathcal{A}$)
Instance: Graph $G = (V, E)$.
Goal: Find a subset $V' \subseteq V$ such that $|V'|$ is maximum and the subgraph $G[V']$ induced by the set V' belongs to $\mathcal{A}$.

Clearly, if $\mathcal{A}$-REC is an NP-complete problem, then $\mathcal{A}$-SUB is also an NP-complete problem. Moreover, $\mathcal{A}$-SUB is NP-complete for any polynomial-time recognizable class of graphs $\mathcal{A}$ which is hereditary, contains arbitrarily large graphs and is not the class of all graphs [7]. Examples of such classes are hereditary clique-Helly, comparability, permutation, perfect, circular-arc, circle, line, planar, bipartite, chordal and interval graphs. However, there are some important - of course non-hereditary - classes of graphs, for which both $\mathcal{A}$-REC and $\mathcal{A}$-SUB are polynomial. Examples are the class of connected graphs and the class of graphs with a perfect matching. Consequently, studying the complexity of $\mathcal{A}$-SUB is not trivial. In addition, when $\mathcal{A}$-SUB is NP-complete, there is no polynomial-time algorithm for solving MAX-$\mathcal{A}$ (providing that $P \neq NP$); so it is natural to ask for approximation algorithms for MAX-$\mathcal{A}$.

In this paper we study those problems for the classes H$\mathcal{K}$H, $\mathcal{K}$H and $\mathcal{K}$. The nomenclature used for them is summarized in a table at the end of Section 2.

In [11,13], it was shown that $\mathcal{K}$H-REC and H$\mathcal{K}$H-REC are polynomial-time solvable problems. In [1], we presented a proof that $\mathcal{K}$-REC is NP-complete, $\mathcal{K}$-SUB is NP-complete. In Section 2, we prove that both $\mathcal{K}$H-SUB and H$\mathcal{K}$H-SUB are NP-complete for maximum degree 6 planar graphs. We also prove that MAX-$\mathcal{K}$, MAX-$\mathcal{K}$H and MAX-H$\mathcal{K}$H are Max SNP-hard for maximum degree 6 graphs, meaning that [10] they are approximable with a fixed ratio in polynomial time

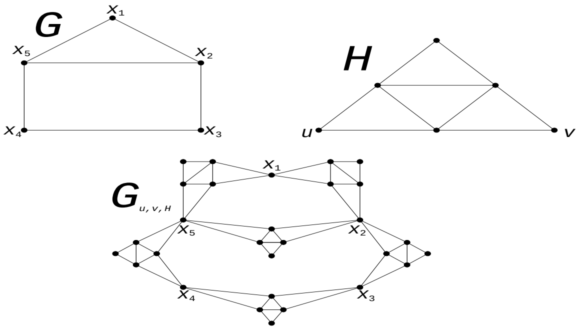


Fig. 1. Graph $G_{u,v,H}$ obtained by replacing each edge of G by a Hajós graph.

but [3] there is a constant $\varepsilon > 0$, such that the existence of a polynomial-time approximation algorithm for MAX-$\mathcal{K}$, for MAX-$\mathcal{K}$H, or for MAX-H$\mathcal{K}$H, restricted to maximum degree 6 graphs, with performance ratio at most $1 + \varepsilon$, implies that P = NP.

In Section 3, we present a general result giving necessary conditions for a class of graphs $\mathcal{A}$, characterized by a collection of forbidden induced subgraphs, to satisfy the property that MAX-$\mathcal{A}$ is Max SNP-hard.

Besides the negative result of Section 2, we show in Section 4 a naive polynomial-time 6-approximation algorithm to minimize the number of vertices to be removed in order to obtain a hereditary clique-Helly subgraph.

Our results are supported by reductions from the NP-complete [6] problem VERTEX COVER (VC) for cubic planar graphs, and the Max SNP-complete [2,3,10] problem MINIMUM VERTEX COVER (MIN-VC) for cubic graphs. For establishing that a problem is Max SNP-hard, we use the approximation preserving reduction, called *L-reduction*, of Papadimitriou and Yannakakis [10].

Due to space limitations some definitions and proofs are omitted.

2 Clique, clique-Helly and hereditary clique-Helly classes

A polynomial-time recognition algorithm for the class of hereditary clique-Helly graphs was presented in [11]. This algorithm uses a finite family of forbidden induced subgraphs, the so-called *ocular graphs*. In [11], ocular graphs were shown to be the minimal forbidden configurations for hereditary clique-Helly graphs. The graph H in Figure 1, called the Hajós graph, is one of the ocular graphs.

For $K_4 - free$ graphs, the classes of clique graphs, clique-Helly graphs and hereditary clique-Helly graphs are the same [14].

Let $G = (V, E)$ be a graph. Let H be the Hajós graph and u, v be the two

non-adjacent vertices H depicted in Figure 1. We obtain a new graph $G_{u,v,H}$ by replacing each edge of G by a Hajós graph. This means that $G_{u,v,H}$ is the graph where $V(G) \subseteq V(G_{u,v,H})$ and, for each edge xy of G, there is a copy S_{xy} of $H - \{u, v\}$ in $G_{u,v,H}$, and the additional set of edges
$E_{xy} = \{xs : s \in V(S_{xy}) \text{ and } us \in E(H)\} \cup \{yt : t \in V(S_{xy}) \text{ and } vt \in E(H)\}$.
We give an example for this operation in Figure 1.

Lemma 2.1 *Let $G = (V, E)$ be any graph and $G_{u,v,H}$ be the graph obtained from G by the process described above. Then,*
1. $G_{u,v,H}$ is K_4-free.
2. If G is cubic and planar, then $G_{u,v,H}$ is a maximum degree 6 planar graph.
3. For any edge xy of G, the subgraphs of $G_{u,v,H}$ induced by $V(S_{xy}) \cup \{x, y\}$, is isomorphic to the Hajós graph H. These are the only induced subgraphs of $G_{u,v,H}$ that are isomorphic to an ocular graph.
4 . $|V(G_{u,v,H})| = |V| + 4\,|E|$.
5. If $V' \subseteq V$, then V' is a vertex cover of G if and only if $G_{u,v,H} - V'$ is an ocular-free induced subgraph of $G_{u,v,H}$. Moreover V' is minimum if and only if $G_{u,v,H} - V'$ has the largest number of vertices.

Theorem 2.2 $\mathcal{K}$H-sub *and* h$\mathcal{K}$H-sub *are NP-complete problems for maximum degree 6 planar graphs.*

Theorem 2.3 max-$\mathcal{K}$*,* max-$\mathcal{K}$H *and* max-h$\mathcal{K}$H *are Max SNP-hard for maximum degree 6 graphs.*

We summarize in the following table our results and the ones in the literature. (*: in the present paper for maximum degree 6 graphs).

Class	Recognition	Induced subgraph	Maximum induced subgraph
clique graph	$\mathcal{K}$-rec **NPC** [1]	$\mathcal{K}$-sub **NPC**	max-$\mathcal{K}$ **Max SNP-hard**
clique-Helly	$\mathcal{K}$H-rec **P** [13]	$\mathcal{K}$H-sub **NPC**	max-$\mathcal{K}$H **Max SNP-hard**
hereditary clique-Helly	h$\mathcal{K}$H-rec **P** [11]	h$\mathcal{K}$H-sub **NPC**[7] *	max-h$\mathcal{K}$H **Max SNP-hard**[8]*

3 A general theorem

In order to generalize the results of Section 2, we consider in the present section, general graphs G and H, and u, v two non-adjacent vertices of H.

The (u, v, H)-*edge-replacing* operation of G is the one which obtains the graph $G_{u,v,H}$ where $V(G) \subseteq V(G_{u,v,H})$, and for each edge xy of G there is a copy S_{xy} of $H - \{u, v\}$ in $G_{u,v,H}$, and the additional set of edges

$$E_{xy} = \{xs, yt : s, t \in V(S_{xy}) \text{ and } us, vt \in E(H)\}.$$

For each edge xy of G, denote by H_{xy} the subgraph of $G_{u,v,H}$ induced by $V(S_{xy}) \cup \{x, y\}$. Notice that H and H_{xy} are isomorphic, i.e. $G_{u,v,H}$ contains H as an induced subgraph.

Observe that the obtained graph $G_{u,v,H}$ may depend on the order the end vertices of each edge xy of G are considered. However, the properties of $G_{u,v,H}$ used in proof of the following results hold regardless we consider for an edge xy of G the order x, y or the order y, x.

Theorem 3.1 *Let $\mathcal{A}$ be a class of graphs, $\mathcal{F}$ be a collection of graphs which characterize $\mathcal{A}$ by forbidden induced subgraphs, and H be an element of $\mathcal{F}$ with a pair of non adjacent vertices u and v, such that, for every graph G, the only induced subgraphs of $G_{u,v,H}$ isomorphic to any element of $\mathcal{F}$ are the subgraphs H_{xy}, where xy is any edge of G. If $n = |V(G)|$ and $m = |E(G)|$ then $Opt_{\text{MAX-}\mathcal{A}}(G_{u,v,H}) = n + m(|V(H)| - 2) - Opt_{\text{MIN-VC}}(G)$.*

Theorem 3.2 *If $\mathcal{A}$ is a class of graphs satisfying the conditions of Theorem 3.1, then* MAX-$\mathcal{A}$ *is Max SNP-hard.*

We mention diamond-free, gem-free, and $K_{3,3}$-free graphs as other classes of graphs for which these results are applicable.

4 A 6-approximation algorithm

In this section we consider the following minimization problem: given a graph $G = (V, E)$, find a minimum subset $V' \subset V$ such that $V - V'$ induces a hereditary clique-Helly subgraph of G. Since the class H$\mathcal{K}$H is characterized by forbidding an ocular graph as induced subgraph, we can design a 6-approximation algorithm Π by recursively looking for an ocular induced subgraph in the current graph; in case we find, its 6 vertices are removed from the current graph.

Each found ocular induced subgraphs requires at least one vertex in the optimum solution. As all of them are vertex disjoint, we have taken at most 6 times the size of the optimum solution.

Future work: We have proved that if $\mathcal{A}$ is H$\mathcal{K}$H, $\mathcal{K}$H or $\mathcal{K}$ then $\mathcal{A}$-SUB is NP-complete and MAX-$\mathcal{A}$ is Max SNP-hard for maximum degree 6 graphs. However, since any ocular graph has a 4-degree vertex, all these problems are polynomial for maximum degree 3 graphs. Hence, it is left as an open problem to determine the maximum k, $3 \leq k \leq 5$, such that $\mathcal{A}$-SUB or MAX-$\mathcal{A}$ are polynomial-time solvable problems for maximum degree k graphs. Besides, we are currently working on the design of approximation algorithms to the problems MAX-$\mathcal{K}$, MAX-$\mathcal{K}$H, and MAX-H$\mathcal{K}$H.

References

[1] L. Alcón, L. Faria, C. M. H. de Figueiredo and M. Gutierrez. Clique graph recognition is NP-complete. In *Proc. WG 2006, Lecture Notes in Comput. Sci.*, vol. 4271, Springer, 2006, pp. 269–277.

[2] P. Alimonti, V. Kann. Hardness of Approximating Problems on Cubic Graphs, Workshop ASHCOMP'96, September 10-19. pp. 288-298, Udine, Italy.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. Proof verification and hardness of approximation problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, FOCS'92, 14–23, 1992.

[4] A. Brandstädt,V.B. Le, J.P. Spinrad. *Graph Classes: A survey*, SIAM Monographs on Discrete Mathematics and Applications, 1999.

[5] M. C. Dourado, F. Protti and J. L. Szwarcfiter. Computational Aspects of the Helly Property: a Survey. *J. of the Brazilian Computer Society*, 12:7–33, 2006.

[6] R. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher (eds.), *Complexity of Computer Computations*, Plenum Press, New York, 85-103, 1972.

[7] J. Lewis, M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980.

[8] C. Lund, M. Yannakakis. The approximation of maximum subgraph problems. in *Proc. 20th Int. Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci.* **700**, Springer-Verlag, 40-51.

[9] T. A. McKee, F. R. McMorris. *Topics in Intersection Graph Theory*, SIAM Monographs on Discrete Mathematics and Applications, 1999.

[10] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. System Sci.*, 43:425–440, 1991.

[11] E. Prisner. Hereditary clique-Helly graphs. *J. Combin. Math. Combin. Comput.*, 14:216–220, 1993.

[12] E. Prisner. *Graph Dynamics*, Pitman Research Notes in Mathematics 338, Longman, 1995.

[13] J. L. Szwarcfiter. Recognizing clique-Helly graphs. *Ars Combinatoria*, 45:29–32, 1997.

[14] J. L. Szwarcfiter. A survey on Clique Graphs. In *Recent Advances in Algorithmic Combinatorics*, C. Linhares-Sales and B. Reed, eds.,Springer-Verlag, 2002.

G-Graphs and Algebraic Hypergraphs

Alain Bretto [a,1] Cerasela Jaulin [a,2] Kevin G. Kirby [b,4]
and Bernard Laget [c,3]

[a] *Department of Computer Science CNRS UMR-6072 Université de Caen, Campus II, Bd. Marechal Juin BP 5186, 14032 Caen cedex, France.*

[b] *Department of Computer Science, Northern Kentucky University, Highland Heights, Kentucky 41099 USA.*

[c] *ENISE 58, rue Jean Parot, 42023 Saint Etienne Cedex 02, France.*

Abstract

$\mathbb{G}$-graphs have been introduced in [4], and have some applications in both symmetric and semi-symmetric graphs, cage graphs, and expander graphs [1], [2], [3]. In this paper we present a characterization of $\mathbb{G}$-graphs, introduce the notion of principal clique hypergraphs, and present some of their basic properties.

Keywords: Algebraic Graph Theory, Cayley Graphs, G-Graphs, Hypergraphs.

1 Introduction

The graphs called $\mathbb{G}$-*graphs* are constructed from a group [1], [4]. These graphs, like Cayley graphs, have nice and highly regular properties; they may

[1] Email: alain.bretto@info.unicaen.fr

[2] Email: cerasela.jaulin@info.unicaen.fr

[3] Email: laget@enise.fr

[4] Email: kirby@nku.edu

or may not be regular. Most well-known graphs are in fact $\mathbb{G}$-graphs: Hamming graphs, meshes of d-ary trees $MT(d,1)$, and some star graphs, to name a few. Moreover the algorithm to construct $\mathbb{G}$-graphs is simple. These graphs can be used in many areas of science where Cayley graphs occur and have applications in coding theory [3] as well as in the construction of symmetric and semi-symmetric graphs [2]. One can also use $\mathbb{G}$-graphs to construct expander graphs.

In this article we first characterize $\mathbb{G}$-graphs. We shall then introduce the notion of *principal-clique hypergraphs* and investigate their basic properties.

2 Preliminaries

In this article all graphs are finite, and we allow multiple edges between vertices; such graphs are sometimes termed *multigraphs*. We write a graph as $\Gamma = (V; E; \epsilon)$, with vertices V, edges E, and a function ϵ assigning each edge to a set of one or two endpoints in V. For convenience, if $a \in E$, we denote $\epsilon(a) = [x; y]$ with the meaning that the extremities x, y of a may be equal (*loop*) or not. For $x, y \in V$, the set $M = \{a \in E, \epsilon(a) = [x; y]\}$ is called *multiedge* or *p-edge* if the cardinality of M is p. For complete definitions, see [1].

A graph is *k-partite* if there is a partition of V in k *parts* such that each part consists contains no edges other than loops; we will write such a graph as $\Gamma = (\sqcup_{i\in I} V_i; E; \epsilon)$, taking $|I| = k$. When k is minimal Γ is called a *k-graph.* $Aut_\pi \Gamma$ is the subgroup of automorphisms of Γ that take each part of the partition to itself.

Recall that an *action* of a group G (with identity e) on a set X is a map $G \times X \to X$, $(g, x) \mapsto g.x$ satisfying $e.x = x$ and $g.(g'.x) = (gg').x$, for every $x \in X, g, g' \in G$. The action is *transitive* if $\forall x, y \in X \quad \exists g \in G$ such that $g.x = y$. The *stabilizer* of $x \in X$ is $Stab_G x = \{g | g.x = x\}$.

A *hypergraph* $\mathcal{H}$ on a finite set $\mathbf{S}$ is a family $(E_i)_{i\in I}$, $I = \{1, 2, \ldots, n\}$ of non-empty subsets of $\mathbf{S}$, called *hyperedges*, with $\bigcup_{i\in I} E_i = \mathbf{S}$. We write $\mathcal{H} = (\mathbf{S}; (E_i)_{i\in I})$. An *intersecting family* in a hypergraph $\mathcal{H}$ is a set of hyperedges having non-empty pairwise intersection. For $x \in \mathbf{S}$, a *star* of $\mathcal{H}$ with x as a center is the set of all hyperedges that contain x, and is denoted $\mathcal{H}(x)$. The *degree of* x is the cardinality of the star $\mathcal{H}(x)$. We will denote it by $deg(x)$. A *triangle* is an intersecting family with three hyperedges which is not contained in a star. The *dual* of a hypergraph $\mathcal{H}$ on $\mathbf{S}$ with vertices x_i and hyperedges E_j is a hypergraph $\mathcal{H}^*$ whose vertices e_j are identified with the hyperedges E_j of $\mathcal{H}$, and whose hyperedges are $X_i = \{e_j | x_i \in E_j\}$. Equiv-

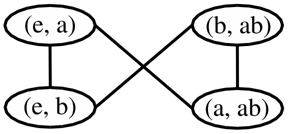
alently, $\mathcal{H}^* = (E, (\mathcal{H}(x))_{x \in \mathbf{S}})$. A hypergraph has the *Helly property* if each intersecting family has a non-empty intersection, thus belonging to a star.
The *2-section* of a hypergraph $\mathcal{H}$ is the graph denoted by $[\mathcal{H}]_2$ whose vertices are vertices of $\mathcal{H}$ and such that two vertices form an edge if and only if they are in the same hyperedge of $\mathcal{H}$. A hypergraph H is *conformal* if all the cliques of $[H]_2$ are hyperedges of H. It can be shown that a hypergraph $\mathcal{H}$ has the Helly property iff its dual is conformal [6]. For a hypergraph $\mathcal{H} = (\mathbf{S}; (E_i)_{i \in I})$ a family $E_0 \subset E$ is defined to be a *matching* if the hyperedges of E_0 are pairwise disjoint. $\nu(\mathcal{H})$ denotes the maximum cardinality of a matching in $\mathcal{H}$. A *transversal* of a hypergraph $\mathcal{H} = (\mathbf{S}; E_1, E_2, \cdots, E_m)$ is defined to be a set of $T \in \mathbf{S}$ such that $T \cap E_i \neq \emptyset$, $i = 1, 2, \cdots, m$. The *transversal number* $\tau(\mathcal{H})$ is defined as the minimum number of vertices in any transversal. It is easy to see that $\nu(\mathcal{H}) \leq \tau(\mathcal{H})$. If $\nu(\mathcal{H}) = \tau(H)$ we will say that $\mathcal{H}$ has the *Koenig property*.

3 G-Graphs of groups

In what follows, G will be a finite group and S a set of elements in G. We consider for any $s \in S$ the (left) action of $\langle s \rangle$ (the subgroup generated by s) on G; this gives a partition $G = \sqcup_{x \in T_s} \langle s \rangle x$ when T_s is a so-called *right transversal* of $\langle s \rangle$ [1], [4]. If $o(s) = |\langle s \rangle|$ is the *order* of s, we have the cycles $(s)x = (x, sx, s^2x, \ldots s^{o(s)-1}x)$ of the permutation $g_s : x \mapsto s\,x \quad (x \in T_s)$.
We are going to build a graph, called the $\mathbb{G}$-*graph* for (G, S), denoted by $\Phi(G; S) = (V = \sqcup_{s \in S} V_s; E; \epsilon)$ in the following way:

- The vertices of $\Phi(G; S)$ are the cycles of g_s, $s \in S$. So $V = \sqcup_{s \in S} V_s$ with $V_s = \{(s)x, x \in T_s\}$.
- For $(s)x, (t)y \in V$, if $|\langle s \rangle x \ \cap \langle t \rangle y| = p$, $p \geq 1$ then $[\langle s \rangle x; \langle t \rangle y]$ is a p-edge.

We may represent the set of edges as $E = \{([\langle s \rangle x; \langle t \rangle y]; u) | u \in \langle s \rangle x \ \cap \langle t \rangle y\}$ so that for an edge $a = ([\langle s \rangle x; \langle t \rangle y]; u)$ one has $\epsilon(a) = [\langle s \rangle x; \langle t \rangle y]$. $\Phi(G; S)$ is a k-partite graph with $|S| = k$. Any vertex has a $o(s)$-loop. We denote $\widetilde{\Phi}(G; S)$ the graph $\Phi(G; S)$ with loops removed.
Example: Let G be the Klein group , $G = \{e, a, b, ab\}$, and $S = \{a, b\}$ with $a^2 = b^2 = e$ and $ab = ba$. The cycles of g_a are: $(a)e = (e, ae) = (e, a)$ and $(a)b = (b, ab)$ The cycles of g_b are: $(b)e = (e, be) = (e, b)$ and $(b)a = (a, ba) = (a, ab)$ The graph $\tilde{\Phi}(G; S)$ is:



Many well known graph families such as Hamming graphs, 1−dimensional meshes of d-ary trees , and certain star graphs, are $\mathbb{G}$-graphs. Moreover many classical graphs such as Pappus graphs, the Heawood graph, and the Cuboctahedral graph are $\mathbb{G}$-graphs as well [5].

4 Properties and characterization

Proposition 4.1 *Let $\Phi(G;S) = (V;E;\epsilon)$ be a $\mathbb{G}$-graph. Then the following properties are equivalent:*

- *$\Phi(G;S)$ has no multiedges except for loops.*
- *For all $s,t \in S$, $\langle s\rangle \cap \langle t\rangle = \{e\}$.*

In particular these properties are obtained when for all $s,t \in S$, $\gcd(o(s), o(t)) = 1$.

As with the Cayley graph $\mathrm{Cay}(G;S)$, it is easy to prove that $\Phi(G;S)$ is connected if $\langle S\rangle = G$; see [4].
Throughout the rest of the paper we limit ourselves to simple graphs (graphs with at most one edge between vertices).
Let $\widetilde{\Phi}(G;S) = (V;E)$ be a $\mathbb{G}$-graph. For any $g \in G$ one can define the maps $\delta_{g^{-1}} : V \to V$ with $\delta_{g^{-1}}((s)x) = (s)xg^{-1}$, and $\delta^{\#}_{g^{-1}} : E \to E$ with $\delta^{\#}_{g^{-1}}(([\langle s\rangle x; \langle t\rangle y], u)) = ([\langle s\rangle xg^{-1}; \langle t\rangle yg^{-1}], ug^{-1})$.
For $x \in G$ we denote by K_x the set of vertices of $\widetilde{\Phi}(G;S)$ that are endpoints of edges of the form $a = ([\langle s\rangle y; \langle t\rangle z], x)$: K_x is a clique, called a *principal clique*. We have the following results:

Theorem 4.2 *Let $\widetilde{\Phi}(G;S) = (V;E)$ be a $\mathbb{G}$-graph.*

- *For all $g \in G$, $(\delta_{g^{-1}}, \delta^{\#}_{g^{-1}}) \in \mathcal{A}ut_{\pi}(\widetilde{\Phi}(G;S))$.*
- *The map $\delta : G \to \mathcal{A}ut_{\pi}(\widetilde{\Phi}(G;S))$ defined by $\delta(g) = (\delta_{g^{-1}}, \delta^{\#}_{g^{-1}})$ is a morphism.*
- *$\delta(G)$ acts transitively on every V_s, $s \in S$.*
  $Stab_{\delta(G)}((s)) = \delta(\langle s\rangle)$ is a cyclic subgroup of $\delta(G)$ with an order $d_s | o(s)$.
  $Stab_{\delta(G)}((s)x) = \delta(x^{-1}\langle s\rangle x)$ and the stabilizers of the vertices of a principal clique are pairwise distinct.

- *$Ker\delta = \bigcap_{s\in S; x\in G} x\langle s\rangle x^{-1}$.*
  If there exist $s, t \in S$ such that $\langle s\rangle \cap \langle t\rangle = \{e\}$ then δ is injective.
- *If $S = \{s_1,\ s_2, \dots\ s_k\}$ and if for all $i \in \{1,\ 2,\ \dots k\}$ there is $\sigma_i \in Aut\widetilde{\Phi}(G; S)$ such that $\sigma_i((s_i)) = (s_{i+1})$, then $\widetilde{\Phi}(G; S)$ is vertex transitive.*

Theorem 4.3 *Let $\Gamma = (V; E)$ with $V = \sqcup_{i\in I} V_i$ be a k-partite semi regular graph. Assume that there exists a subgroup H of $\mathcal{A}ut_\pi(\Gamma)$ such that:*

- *H acts transitively on each V_i, $i \in I$.*
- *For all $x \in V$, $Stab_H x$ is a cyclic group.*
- *For all i, j, $i \neq j$, H acts transitively on the set of edges incident to V_i and V_j.*
- *The graph Γ contains a clique with k vertices, $K = \{x_1, x_2, \dots x_k\}$ such that $Stab_H x_i$, $i \in \{1, 2, 3 \dots k\}$ are pairwise distinct and $\frac{o(G)}{|Stab_H x_i|} = |V_i|$.*

Then there exists a $(G; S)$ such that $\widetilde{\Phi}(G; S) \backsimeq \Gamma$.

5 Principal clique hypergraphs

In this section we restrict our attention to groups G that are generated by a set S of involutions (i.e., $G = \langle S\rangle$ with $a^2 = e$ for all $a \in S$). These types of groups are very important; for example the Artin group and the braid groups are generated by involutions. It has been conjectured that every non-abelian finite simple group is generated by three involutions. Let $\Phi(G; S)$ be a $\mathbb{G}$-graph. We build a hypergraph in the following way :

- The set of vertices $\mathbf{S}$ is the same as in $\Phi(G; S)$
- The set of hyperedges is the set of principal cliques.

We denote this hypergraph by $\mathcal{H}(\Phi(G; S)) = (\mathbf{S}, E)$, and call it a *principal clique hypergraph*. We now investigate some of its basic properties.

Proposition 5.1 *Let G be a group, and suppose that the set S of generating involutions is minimal. Then:*

- *$\mathcal{H}(\Phi(G; S))$ has the Helly property.*
- *$\mathcal{H}(\Phi(G; S))$ is conformal.*

Proof. It is easy to see that the dual $\mathcal{H}^*(\Phi(G; S))$ of $\mathcal{H}(\Phi(G; S))$ is a simple graph. Because $\mathcal{H}(\Phi(G; S))$ does not contain any triangle, $\mathcal{H}^*(\Phi(G; S))$ does not contain any triangle as well. Consequently the property follows. □

Proposition 5.2 *The hypergraph $\mathcal{H}(\Phi(G; S))$ has the Koenig property.*

Proof. It is easy to see that $\nu(\mathcal{H}) = \frac{n}{2}$ (n being the cardinality of G). Moreover the number of vertices in g_{s_i} is equal to $\frac{n}{2}$. So $\nu(\mathcal{H}) = \tau(n)$. □

6 Conclusion

In this article we have introduced principal clique hypergraphs and have laid out some of their basic properties. We have seen that the properties arise from the group, namely the minimality of S and the cardinality of G. Is it possible to extract some information about $(G; S)$ from the structure of the hypergraph $\mathcal{H}(\Phi(G; S))$? For example, what does the Helly property mean when translated into group theory [7]? Is it possible to generalize the properties given in section 5 for any group? Finally, investigating the automorphism group of the hypergraph also appears to be a promising avenue to explore.

References

[1] A. Bretto, A. Faisant and L. Gillibert, *G-graphs: A new representation of groups.* Journal of Symbolic Computation, Volume 42, Issue 5, May 2007, Pages 549-560.

[2] A. Bretto, L. Gillibert and B. Laget, *Symmetric and Semisymmetric Graphs Construction Using G-graphs*, Proceedings of ISSAC'05, July 24-27 2005, Beijing, China. ACM Proceedings (2005), 61-67.

[3] A. Bretto, L. Gillibert, *A New Upper Bound for the (p,6) cages*, Proceedings of ISSAC'07, July 29-August 1st 2007, Waterloo Canada. ACM Proceedings (2007).

[4] A. Bretto, A. Faisant. *A new way for associating a graph to a group*, Mathematica Slovaca **55** , No.1, (2005), 1-8.

[5] A. Bretto, L. Gillibert. *Symmetry and Connectivity in G-graphs*, Proceedings of the 7th International Colloquium on Graph Theory (ICGT 05), September 12-16 Electronic Notes in Discrete Mathematics Elsevier **22**, 2005, 481-486.

[6] M. Costa Dourado, F. Protti, J. L. Szwarcfiter *The Helly property on subfamilies of limited size.* Information processing letters, **93**, 53-56.

[7] M. C. Dourado, P. Petito, R. B. Teixeira *Helly Property and Sandwich Graphs* Electronic Notes in Discrete Mathematics **22** (2005) 497-500.

Stronger upper and lower bounds for a hard batching problem to feed assembly lines

Alexandre Salles da Cunha , [1,2]

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil

Maurício Cardoso de Souza [3]

Departamento de Engenharia de Produção
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil

Abstract

In this paper we present an Integer Programming reformulation for a hard batching problem encountered in feeding assembly lines. The study was motivated by the real process to feed the production flow through the shop floor in a leading automobile industry in Brazil. The problem consists of deciding the assignment of items to containers and the frequency of moves from the storage area to the line in order to meet demands with minimum cost. Better lower and upper bounds were obtained by a branch-and-bound algorithm based on the proposed reformulation. We also present valid inequalities that may improve such algorithm even further.

Keywords: Assembly lines, Lot-sizing in batches, Integer Programming.

[1] This project is funded by Programa Récem Doutores - UFMG/PRPq

[2] Email: `acunha@dcc.ufmg.br`

[3] Email: `mauricio.souza@pq.cnpq.br`

1 Introduction

Firms employing just-in-time attain consistent quality with low cost due to standardization of work methods. The shop floor is arranged in parallel assembly lines to produce efficiently high volumes, as it happens in automobile assembly plants. A feed process supplies the serial work centers with all the necessary items to complete the required operations. This leads to hard batching problems since materials handling to meet a time based demand must be done in standard sizes of containers. The feed process is subjected to quite restrictive operational constraints: (i) a container must carry only one kind of item; (ii) for each kind of item exactly one container size have to be chosen to handle the item within the whole planning horizon; and, (iii) only full conteiners can be used, i.e., the quantity a container size must always carry when handling a certain kind of item is defined a priori.

The assembly line feed problem (LFP, for brief), consists of jointly deciding the assignment of items to containers and the frequency of moves from the storage area to the line in order to meet work centers' demands with minimum cost. The costs are associated to holding stock of items besides the line and to handling the containers. The LFP, introduced by Souza et al. [3], is a real shop floor problem encountered in a major automobile industry plant located in Brazil. The authors discussed the problem complexity, and proposed an integer programming model and a GRASP heuristic. This problem is related to the lot-sizing with constant batches [1]. In this paper, we propose an IP reformulation and adapt valid inequalities coming from the lot-sizing literature for the LFP problem. The reformulation is compared, from a computational point of view, with the IP formulation presented in [3]. We conclude with suggestions for future investigations.

2 Integer programming formulations

A production plan determines the demand d_{it} for item i, $i = 1, \ldots, I$, to be met in period t, $t = 1, \ldots, T$ of a finite horizon. The line is supplied with a fixed quantity q_{ik} whenever a container of size k arrives with an item i. The handling cost is b_k per movement of a container k, and the holding cost is c_i per unity of item i remaining besides the line from a period to another. We assume that l_k containers of size k are availabe to feed the line at each period.

The IP formulation introduced in [3] used a binary variable x_{ik} assuming value 1 if item i was assigned to container size k, and 0 otherwise. A variable f_{it}^{k} denoted the frequency of movements with container k carrying item i in

t, and s_{it} denoted the stock of item i at the end of period t. The resulting formulation was written as follows:

$$\min \left\{ \sum_{t=1}^{T} \sum_{i=1}^{I} \left(c_i s_{it} + \sum_{k=1}^{K} b_k f_{it}^k \right) \right\} \tag{1}$$

$$s_{i(t-1)} - s_{it} + \sum_{k=1}^{K} q_{ik} f_{it}^k = d_{it}, \ \ \forall i, \forall t, \tag{2}$$

$$\sum_{k=1}^{K} x_{ik} = 1, \ \ \forall i, \tag{3}$$

$$\sum_{t=1}^{T} f_{it}^k - M x_{ik} \leq 0, \ \ \forall i, \forall k, \tag{4}$$

$$\sum_{i=1}^{I} w_{it}^k x_{ik} \leq l_k, \ \ \forall k, \forall t, \tag{5}$$

$$f_{it}^k \in \mathbb{Z}_+, \forall i, \forall t, \forall k, \ x_{ik} \in \{0,1\}, \ \forall i, \forall k, \ s_{it} \in \mathbb{R}_+, \forall i, \forall t, \tag{6}$$

The objective function (1) minimizes the total cost, and constraints (2) are the classical flow balance. Constraints (3) and (4) respectively assign items to container sizes and couple frequency and assignment variables. Parameter M is an upper bound on the frequency variables. In [3], M was estimated as $\max\{\sum_{t=1}^{T} \lceil \frac{d_{it}}{q_{i\bar{k}}} \rceil, i = 1, \ldots, I\}$, where $\bar{k}$ is the smallest container. Constraints (5) impose that at most l_k containers of size k can be used in each period t. Parameter w_{it}^k denotes the number of containers of size k required to feed the line with item i in t (see [3] for a detailed discussion on parameter w_{it}^k).

In the present study, we decompose the assigning variable per period t, i.e., we use new variables z_{it}^k which assume 1 if item i is assigned to container size k in period t, and 0 otherwise. We also redefine upper bound M as a function of i, k and t, i.e., $M_{it}^k := \left\lceil \sum_{t'=t}^{T} \frac{d_{it'}}{q_{ik}} \right\rceil$. After eliminating decision variables s_{it}, we propose the resulting LFP reformulation:

$$\min \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T} \left[-c_i D(i,1,t) + \sum_{k=1}^{K} \left(b_k + (T - t + 1) c_i q_{ik} \right) f_{it}^k \right] \right\} \tag{7}$$

$$\sum_{t'=1}^{t} \sum_{k=1}^{K} q_{ik} f_{it'}^k \geq D(i,1,t), \ \ \forall i, \forall t, \tag{8}$$

$$\sum_{k=1}^{K} z_{it}^k \leq 1, \ \ \forall i, \forall t, \tag{9}$$

$$z_{it}^k = z_{i(t-1)}^k, \ \ \forall i, \forall k, \forall t \in \{2, \dots, T\}, \tag{10}$$

$$\sum_{i=1}^{I} w_{it}^k z_{it}^k \le l_k, \ \ \forall t, \forall k, \tag{11}$$

$$\sum_{t=1}^{T} f_{it}^k - M_{i1}^k z_{i1}^k \le 0, \ \ \forall i, \forall k, \tag{12}$$

$$f_{it}^k - M_{it}^k z_{it}^k \le 0, \ \ \forall i, \forall k, \forall t, \tag{13}$$

$$f_{it}^k \in \mathbb{Z}_+, \ z_{it}^k \in \{0,1\}, \ \forall i, \forall k, \forall t. \tag{14}$$

where $D(i,t',t) := \sum_{j=t'}^{t} d_{ij}$. Constraints (8) ensure stocks are non negative, while (9) and (10) assign items to container sizes. Constraints (13) try to impose a tighter coupling between frequency and assignment variables. This formulation can be strengthened even further by the following results.

Proposition 2.1 *Let $s_{il} = \sum_{k=1}^{K} \sum_{t=1}^{l} f_{it}^k q_{ik} - D(i,1,l)$ be the stock of item i at period l. Given $L = \{1, \dots, l\}$ and $S \subseteq L$, the (i, L, S) inequality*

$$\sum_{j \in S} \sum_{k=1}^{K} f_{ij}^k q_{ik} \le \sum_{j \in S} \sum_{k=1}^{K} D(i,j,l) z_{ij}^k + s_{il} \tag{15}$$

is valid for LFP.

Proof. There exists, in a solution feasible for (7)-(14), only one container size $\overline{k}$ such that $z_{ij}^{\overline{k}} = 1$ for all $j \in S$. If $\overline{t}$ denotes the least period in S, then:

$$\begin{aligned} \sum_{j \in S} \sum_{k=1}^{K} f_{ij}^k q_{ik} &\le \sum_{j=\overline{t}}^{l} \sum_{k=1}^{K} f_{ij}^k q_{ik} \le D(i, \overline{t}, l) + s_{il} \\ &\le \sum_{j \in S} D(i,j,l) z_{ij}^{\overline{k}} + s_{il} \le \sum_{j \in S} \sum_{k=1}^{K} D(i,j,l) z_{ij}^k + s_{il}. \end{aligned} \tag{16}$$

□

Corollary 2.2 *The (i, L, S) inequality can be written in the (z, f) space as:*

$$\sum_{j \in L \setminus S} \sum_{k=1}^{K} f_{ij}^k q_{ik} + \sum_{j \in S} \sum_{k=1}^{K} D(i,j,l) z_{ij}^k \ge D(i,1,l). \tag{17}$$

Inequalities (15) are a generalization of facet-defining inequalities for the uncapacitated single-item lot-sizing problem (see Section 7.4 in [2] for details). Although exponentially many exist, (i, L, S) inequalities (17) can be used to strengthen the LP relaxation bounds of (7)-(14), if they are generated as cutting planes. Given a vector $(\overline{z}, \overline{f}) \in (\mathbb{R}^{K \times I \times T}, \mathbb{R}^{K \times I \times T})$, the associated separation problem can be solved efficiently in $O(IKT^2)$ time by computing, for each $(i,l) \in \{1,\dots,I\}\times\{1,\dots,T\}$, $\alpha_{il} = \sum_{j=1}^{l} \min\{\sum_{k=1}^{K} \overline{f}_{ij}^{k} q_{ik}; \sum_{k=1}^{K} D(i,j,l)\overline{z}_{ij}^{k}\}$. If $\alpha_{il} < D(i,1,l)$ then the separation routine returns i, $L = \{1,\dots,l\}$, and $S = \{j \in L : \sum_{k=1}^{K} \overline{f}_{ij}^{k} q_{ik} > \sum_{k=1}^{K} D(i,j,l)\overline{z}_{ij}^{k}\}$.

3 Preliminary computational results

Our preliminary computational experiments comprise evaluating the formulations presented in the previous section by means of the strength of their LP relaxations and by the overall performance of a branch-and-bound (BB) algorithm based on them. Accordingly, two BB algorithms were tested: BB1, based on the original formulation (1)-(6), and BB2, based on the proposed reformulation (7)-(14). Inequalities (17) were not considered yet. We used `XPRESS-MP` Mosel modeling language. Parameters `PRESOLVE, HEURSTRATEGY, NODESELECTION, CUTSTRATEGY` and `CUTFREQUENCY` were set, respectively, to 0,3,5,2 and 1. We have thus given more priority on finding feasible solutions. Both BB algorithms were ran for one hour of CPU on a 2.4 GHz Intel Pentium IV, with 1Gb of RAM memory, under Linux Operating System.

The experiments were conducted with a set of 24 real FLP instances introduced in [3] ($I = 191$, $K = 3$ and $T = 7$). For each instance, the first (resp. second) formulation involve 2123 (10717) rows, 5922 (8022) columns and 21011 (43548) non zero entries. Table 1 shows numerical results. In the first column of the table, we identify each instance by its name. In the next three columns we present the following ratios: $\frac{LP_2}{LP_1}$, between the lower bounds given by the LP relaxations of (7)-(14) and (1)-(6); $\frac{BLB_2}{BLB_1}$, between the best overall lower bounds obtained by the BB algorithms; $\frac{UB_2}{UB_1}$, between the best upper bounds found within the enumeration tree by each BB algorithm. We then report BLB_2 and UB_2, respectively the best lower and upper bound attained by BB2, followed by the corresponding duality gap $\frac{UB_2 - BLB_2}{BLB_2}$. In the last column we show the upper bounds attained by GRASP [3]. The computation of LP_1 and LP_2 typically takes less than 1 second.

As depicted in Table 1, much better results were obtained when LFP reformulation (7)-(14) was used instead of (1)-(6). From the one hand, LP_2 values were, on the average, 14.6% stronger than LP_1 counterparts, while the best overall lower bounds attained by BB2 were 36.7% tighter. From the other hand, the best feasible solutions found by BB2 were 45.2% cheaper than those found by BB1. BB2 also compares favorably in terms of solution quality with the GRASP proposed in [3]. We were able, with the reformulation proposed in this paper, to improve the best known upper bounds for all the 24 instances (31.3% on average).

Instance	$\frac{LP_2}{LP_1}$	$\frac{BLB_2}{BLB_1}$	$\frac{UB_2}{UB_1}$	BLB_2	UB_2	$\frac{UB_2-BLB_2}{BLB_2}$	UB GRASP
t191_111	1.075	1.432	0.560	152,633.8	206,841	0.355	301,239
t191_112	1.270	1.490	0.611	157,639.8	203,102	0.288	307,551
t191_113	1.747	1.502	0.742	173,484.0	222,904	0.285	324,428
t191_121	1.000	1.254	0.528	172,283.4	190,713	0.107	293,342
t191_122	1.000	1.243	0.599	170,650.7	192,290	0.127	297,568
t191_123	1.001	1.244	0.662	170,913.6	207,914	0.216	300,127
t191_211	1.083	1.333	0.556	165,158.9	213,046	0.290	310,867
t191_212	1.285	1.374	0.559	169,881.0	210,576	0.240	317,445
t191_213	1.769	1.503	0.651	186,150.8	225,963	0.214	336,429
t191_221	1.000	1.224	0.517	183,166.2	198,930	0.086	308,444
t191_222	1.000	1.225	0.529	183,295.6	199,320	0.087	308,554
t191_223	1.002	1.238	0.586	183,794.4	220,180	0.198	311,041
tb191_111	1.395	1.428	0.468	160,562.0	220,235	0.372	303,443
tb191_112	1.006	1.307	0.483	151,504.7	207,971	0.373	307,361
tb191_113	1.000	1.388	0.482	151,598.6	213,953	0.411	309,741
tb191_121	1.000	1.353	0.619	170,837.4	202,148	0.183	298,467
tb191_122	1.039	1.373	0.478	181,087.7	234,213	0.293	326,143
tb191_123	1.039	1.381	0.527	184,726.4	252,720	0.368	337,711
tb191_211	1.648	1.442	0.499	181,312.8	225,353	0.243	331,144
tb191_212	1.057	1.296	0.600	161,783.1	221,008	0.366	316,910
tb191_213	1.000	1.779	0.367	162,167.9	220,534	0.360	319,356
tb191_221	1.000	1.219	0.502	182,610.2	215,093	0.178	310,318
tb191_222	1.039	1.408	0.483	193,404.0	239,685	0.239	338,280
tb191_223	1.039	1.383	0.548	196,946.8	256,420	0.302	349,552
Average	1.146	1.367	0.548	-	-	0.252	-

Table 1

Numerical results on lower and upper bounds

Although much better lower and upper bounds were obtained with a BB algorithm based on the proposed reformulation, we are still unable to solve instances of practical interest to proven optimality within a reasonable amount of computational time. A Branch-and-cut algorithm where inequalities (17) are used as cutting planes may help us to go around this difficulty.

References

[1] Pochet, Y., and L.A. Wolsey, *Lot-sizing with constant batches: Formulations and valid inequalities*, Math. Oper. Res. **14** (1993), 767–785.

[2] Pochet, Y., and L.A. Wolsey, “Production Planning by Mixed Integer Programming”, Springer, 2006.

[3] de Souza, M.C., C.R.V. de Carvalho, and W.B. Brizon, *Packing items to feed assembly lines*, Eur. J. Oper. Res. (2007), doi:10.1016/j.ejor.2006.09.091.

Flow-Critical Graphs

Cândida Nunes da Silva[1], Cláudio L. Lucchesi[2]

Institute of Computing, UNICAMP, Caixa Postal 6176, 13084-971, Campinas, SP, Brasil, Fax: 55 19 3521 5847

Abstract

In this paper we introduce the concept of *k-flow-critical* graphs. These are graphs that do not admit a k-flow but such that any smaller graph obtained from it by contraction of edges or of pairs of vertices is k-flowable. Any counterexample for Tutte's 3-Flow and 5-Flow Conjectures must be 3-flow-critical and 5-flow-critical, respectively. Thus, any progress towards establishing good characterizations of k-flow-critical graphs can represent progress in the study of these conjectures. We present some interesting properties satisfied by k-flow-critical graphs discovered recently.

Keywords: nowhere-zero k-flows, Tutte's Flow Conjectures, edge-k-critical graphs, vertex-k-critical graphs.

1 Introduction

Let k be an integer, $k > 2$. Let D be a digraph and f a mapping $f : E(D) \to \mathbb{Z}$. Let X be a set of vertices of D and the *cut of* X, denoted by ∂X, the set of edges of D having precisely one end in X. We define the *outflow at* X as the

[1] Financially supported by FAPESP and CAPES. Email: `candida@ic.unicamp.br`

[2] Research supported in part by CNPq — Conselho Nacional de Desenvolvimento Cientfico e Tecnolgico, grant #306088/2005-3. Email: `lucchesi@ic.unicamp.br`

sum of the weights of the edges in ∂X having tail in X minus the sum of the weights of the edges in ∂X having head in X. In the particular case in which X has one single vertex v we say it is the *outflow at* v and denote it by $f(v)$. The outflow at X is equal to the sum of the outflows at the vertices in X.

Mapping f is a *k-flow* of D if the following properties are satisfied:

(i) For any edge e, $f(e)$ is not a multiple of k.

(ii) Every vertex v is *balanced*, i.e., $f(v) = 0$.

Similarly, f is a *mod-k-flow* of D if property (i) is satisfied and every vertex is *balanced modulo k*, i.e., $f(v) \equiv 0 \pmod{k}$. Mapping f is a *near-mod-k-flow that misses u and v* if property (i) is satisfied and balance modulo k is achieved at every vertex, except precisely vertices u and v.

If a mapping f satisfies property (i) and all but possibly one vertex v are balanced, then f must be a k-flow. Recall that the outflow at v is the complement of the outflow at $X := V(D) - v$. Since the outflow at $X := V(D) - v$ is 0, then v must be balanced as well. Similarly, if a mapping f satisfies property (i) and all but possibly one vertex v are balanced modulo k, then f must be a mod-k-flow. Using a similar argument we conclude that when f is a near-mod-k-flow that misses u and v and the outflow at u is x, then the outflow at v must be $-x$. In order to specify the outflow at u (and consequently at v) we say f is a *near-mod-k-flow that misses u and v by x*.

When f is either a k-flow or a mod-k-flow then, for every set X, the outflow at X is 0 or 0 mod k, respectively. Thus, D cannot have a cut-edge.

We say a graph G has a k-flow if there is a k-flow for some orientation D of G. Similarly, we say that G has a mod-k-flow if there is a mod-k-flow for some orientation D of G. The following result is well known. Proofs of Theorem 1.1 can be found in papers by Younger [6] and Seymour [4], and in a book by Zhang [7].

Theorem 1.1 *A graph G has a k-flow if and only if G has a mod-k-flow.*

It should be noted, however, that there is a mod-k-flow f for some orientation D of G if and only if there is a mod-k-flow f' for every orientation D' of G. If D and D' differ on a set E' of edges, we obtain f' by simply stating that $f'(e) \equiv -f(e) \pmod{k}$ for all edges in E' and $f'(e) = f(e)$ for the remaining edges.

We denote by G/e or $G - e$, respectively, the graph resulting from contracting or removing edge e. For a pair of distinct vertices u and v of G, G_{uv} is the graph obtained from G by contracting the set $\{u, v\}$ to a single vertex. Graph $G + uv$ is the graph obtained from G by the addition of edge uv. All these definitions can be extended to a digraph D. In this case, graph $D + uv$ is an extension of D in which edge uv is directed from u to v.

A graph G is *edge-k-critical* if it does not admit a k-flow but G/e admits a k-flow, for every edge e of G. Similarly, we say that graph G is *vertex-k-critical* if it does not admit a k-flow but G_{uv} admits a k-flow, for every pair of distinct vertices u and v of G. These definitions can be extended to digraphs. A digraph D is *edge-k-critical* or *vertex-k-critical*, respectively, if its underlying undirected graph is edge-k-critical or vertex-k-critical.

Every loopless vertex-k-critical graph must be edge-k-critical. A graph G that is edge-k-critical must have exactly one non-trivial connected component. This non-trivial component of G must be edge-k-critical itself. It is easy to see that every vertex-k-critical graph is connected. There are edge-k-critical graphs that are not vertex-k-critical, but all examples we know are disconnected.

Conjecture 1.2 *Every connected edge-k-critical graph is vertex-k-critical.*

Graph K_4 is an example of a vertex-3-critical graph. Actually, every odd wheel is vertex-3-critical. Other examples of vertex-3-critical graphs are shown on Figure 1. The Petersen graph is vertex-4-critical. Many other snarks are vertex-4-critical. Examples are Blanuša, Loupekhine, Celmins-Swart, double-star and Szekeres snarks, and flower-snarks J_n for n odd and $5 \leq n \leq 15$ [1]. We used a characterization of vertex-k-critical graphs demonstrated on Section 3 to implement a computer program that checks whether a graph is vertex-k-critical for $k = 3$ or $k = 4$. This program runs in exponential time.

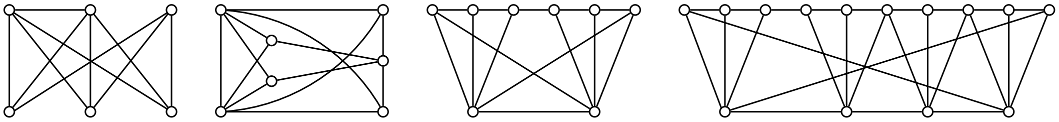


Fig. 1. Examples of vertex-3-critical graphs

2 Motivation

Tutte proposed three well known conjectures concerning 5-, 4- and 3-flows. Tutte's 5-Flow Conjecture states that every 2-edge-connected graph admits a 5-flow. The 4-Flow Conjecture states that every 2-edge-connected graph with no Petersen minors admits a 4-flow. Finally, the 3-Flow Conjecture states that every 4-edge-connected graph admits a 3-flow. All three conjectures are theorems for planar graphs. Robertson, Seymour and Thomas [3] have proved the 4-Flow Conjecture for cubic graphs. The 3-Flow Conjecture has been proved for planar graphs with up to three 3-cuts and for projective planar graphs with at most one 3-cut [5]. There are many more interesting results concerning these conjectures that are not mentioned here. Refer to Seymour [4] or Zhang [7] for a more thorough review on related results. All three conjectures are still open.

Since the contraction of pairs of vertices does not create new cuts, any counterexample for the 5-Flow Conjecture or the 3-Flow Conjecture must be, respectively, vertex-5-critical or vertex-3-critical. Moreover, contraction of edges does not create Petersen minors, thus any counterexample for the 4-Flow Conjecture is edge-4-critical. We do not know, however, whether it is also vertex-4-critical. There are examples of graphs that do not have a Petersen minor for which the contraction of some pair of vertices creates a Petersen minor.

Therefore, the search for good characterizations of edge and vertex-k-critical graphs for $k = 3, 4, 5$ stands up as a new interesting approach towards proving these conjectures.

3 Characterizations of k-critical Graphs

Theorems 3.1 and 3.2 are, respectively, characterizations of edge and vertex-k-critical graphs. We omit here the proof of Theorem 3.2, but it can be proved using essentially the same techniques presented on the proof of Theorem 3.1.

Theorem 3.1 *Let D be a digraph with no mod-k-flow. Then, the following properties are equivalent:*

(i) *Graph D is edge-k-critical.*

(ii) *For every edge e of D, graph $D - e$ has a mod-k-flow.*

(iii) *For every edge $e = uv$ of D, graph D has a near-mod-k-flow that misses u and v.*

Proof.

$(i) \Rightarrow (ii)$: Let e be an edge of D and f a mod-k-flow of D/e. Either f is a mod-k-flow of $D - e$ or f is a near-mod-k-flow of $D - e$ that misses the ends of e. But the latter cannot be the case, otherwise, f can be extended to a mod-k-flow of D, a contradiction. Thus, f is a mod-k-flow of $D - e$.

$(ii) \Rightarrow (iii)$: Let $e = uv$ be an edge of D and f' be a mod-k-flow of $D - e$. Extend f' to a mapping f of D by assigning to e a weight $f(e)$ that is not a multiple of k; f is a near-mod-k-flow of D that misses u and v.

$(iii) \Rightarrow (i)$: Let $e = uv$ be an edge of D and f a near-mod-k-flow of D that misses u and v. Take the restriction f' of f to D/e. All vertices, except perhaps the vertex of contraction uv, are balanced modulo k. Thus, so is vertex uv and f' is a mod-k-flow of D/e. □

Theorem 3.2 *Let D be a digraph with no mod-k-flow. Then, the following properties are equivalent:*

(i) *Graph D is vertex-k-critical.*

(ii) *For every pair $\{u, v\}$ of two distinct vertices of D, the graph $D + uv$ has a mod-k-flow.*

(iii) *For every pair $\{u, v\}$ of two distinct vertices of D, there is a near-mod-k-flow of D that misses u and v.*

4 Composition and Decomposition of k-critical Graphs

Let D_1 and D_2 be two digraphs, $e = u_1v_1$ an edge of D_1 and u_2 and v_2 be two vertices of D_2. The *composition of D_1 and D_2 by (u_1, v_1) and (u_2, v_2)* is the graph obtained from $D_1 - e$ and D_2 by the identification of the pairs of vertices u_1, u_2 into a vertex u and v_1, v_2 into a vertex v. A digraph D is a composition of D_1 and D_2 if for some edge $e = u_1v_1$ of D_1 and for two specified vertices u_2 and v_2 of D_2, D is a composition of D_1 and D_2 by (u_1, v_1) and (u_2, v_2).

Kochol [2, Lemma 1] proves that given two digraphs D_1 and D_2 such that neither D_1 nor D_2 has a mod-k-flow, then no composition of D_1 and D_2 has a mod-k-flow. We thus state a conjecture and present some evidence in support of its validity.

Conjecture 4.1 *Let D_1 and D_2 be two vertex-k-critical digraphs. Then any composition of D_1 and D_2 is vertex-k-critical.*

Theorem 4.2 *Let k be a prime, D_1 an edge-k-critical digraph and D_2 a loopless vertex-k-critical digraph. Then any composition of D_1 and D_2 is edge-k-critical.*

Proof. Let $e = u_1v_1$ be an edge of D_1 and u_2 and v_2 two vertices of D_2 such that D is a composition of D_1 and D_2 by (u_1, v_1) and (u_2, v_2). Let e_2 be an edge of D_2. By hypothesis D_2 is loopless and vertex-k-critical. Thus, it is edge-k-critical. By Theorem 3.1(ii), $D_2 - e_2$ has a mod-k-flow f_2 and $D_1 - e$ has a mod-k-flow f_1. The union of f_1 and f_2 is a mod-k-flow of $D - e_2$.

Now let e_1 be an edge of $D_1 - e$. By Theorem 3.1(ii), $D_1 - e_1$ has a mod-k-flow f_1. Thus, $D_1 - e_1 - e$ has a near mod-k-flow f_1' that misses u_1 and v_1 by $f_1(e) = x$. By Theorem 3.2(iii), D_2 has a near mod-k-flow f_2 that misses u_2 and v_2 by, say, y. Since k is prime, the inverse $z \equiv y^{-1} \pmod{k}$ exists. Multiplying f_2 by $-xz$ we obtain a near mod-k-flow f_2' of D_2 that misses u_2 and v_2 by $-x$ mod k. The union of f_1' and f_2' is a mod-k-flow of $D - e_1$. Every edge of D is either an edge of D_2 or of $D_1 - e$, thus, D is edge-k-critical. □

Given a digraph D, a *decomposition of* D is a pair of graphs D_1 and D_2 such that D is a composition of D_1 and D_2. Clearly, a decomposition of D exists if and only if D has a 2-vertex-cut. Every vertex-k-critical digraph D with a 2-vertex-cut has a decomposition into two vertex-k-critical digraphs.

Theorem 4.3 *Let k be a prime, D a vertex-k-critical digraph, $\{u, v\}$ a 2-vertex-cut of D and D_1 and D_2 subgraphs of D such that $V(D_1) \cap V(D_2) = \{u, v\}$ and $D = D_1 \cup D_2$. Then, either (i) $D_1 + uv$ and D_2 are both vertex-k-critical or (ii) D_1 and $D_2 + uv$ are both vertex-k-critical.*

A similar result holds for edge-k-critical graphs.

References

[1] Chetwynd, A. G. and R. J. Wilson, *Snarks and Supersnarks*, in: Y. Alavi, G. Chartrand, D. L. Goldsmith, L. Lesniak-Foster and D. R. Lick, editors, *The Theory and Applications of Graphs* (1981), pp. 215–241.

[2] Kochol, M., *An equivalent version of the 3-flow conjecture*, Journal of Combinatorial Theory, Series B **83** (2001), pp. 258–261.

[3] Robertson, N., P. D. Seymour and R. Thomas, *Tutte's edge colouring conjecture*, Journal of Combinatorial Theory **B 70** (1997), pp. 166–183.

[4] Seymour, P. D., *Nowhere-zero flows*, in: R. L. Graham and L. L. M. Grtschel, editors, *Handbook of Combinatorics*, Elsevier, 1995 pp. 289–299.

[5] Steinberg, R. and D. H. Younger, *Grötzsch's Theorem for the projective plane*, Ars Combinatoria **28** (1989), pp. 15–31.

[6] Younger, D. H., *Integer flows*, Journal of Graph Theory **7** (1983), pp. 349–357.

[7] Zhang, C.-Q., "Integer Flows and Cycle Covers of Graphs," Marcel Dekker, 1997.

A Polynomial Time Algorithm for Recognizing Near-Bipartite Pfaffian Graphs

Alberto Alexandre Assis Miranda [1,2]
Cláudio Leonardo Lucchesi [3,4]

Institute of Computing
University of Campinas
Campinas, Brazil

Abstract

A *matching covered* graph is a nontrivial connected graph in which every edge is in some perfect matching. A *single ear* of a connected graph G is a path P of odd length in G whose internal vertices, if any, have degree two in G. Let $G - P$ denote the graph obtained from G by deleting the edges and internal vertices of G. A *double ear* of G is a pair (R_1, R_2) of vertex-disjoint single ears of G. A matching covered graph G is *near-bipartite* if it is non-bipartite and there is a double ear (R_1, R_2) such that $G - R_1 - R_2$ is matching covered and bipartite, but $G - R_i$ is not matching covered, for $i = 1, 2$. In 2000, C. Little and I. Fischer characterized Pfaffian near-bipartite graphs in terms of forbidden subgraphs [2]. However, their characterization does not imply a polynomial time algorithm to determine whether a near-bipartite graph is Pfaffian. In this report, we give such an algorithm. Our algorithm is based in an Inclusion-Exclusion principle and uses as a subroutine an algorithm by McCuaig [4] and, independently, by Robertson, Seymour and Thomas [6] that determines whether a bipartite graph is Pfaffian.

Keywords: Pfaffian graphs, polynomial time algorithms, matching theory, matching covered graphs.

1 Introduction

Let $A := (a_{ij})$ be an $n \times n$ skew-symmetric matrix. When n is even, say $n = 2k$, there is a polynomial $P := P(A)$ in the a_{ij} called *Pfaffian* of A. This polynomial is defined as follows:

$$P := \sum \operatorname{sgn}(M) a_{i_1 j_1}\, a_{i_2 j_2}\, \ldots\, a_{i_k j_k},$$

where the sum is taken over the set of all partitions $M := (i_1 j_1, i_2 j_2, \ldots, i_k j_k)$ of $\{1, 2, \ldots, n\}$ into k unordered pairs, and $\operatorname{sgn}(M)$ is the sign of the permutation:

$$\pi(M) := \begin{pmatrix} 1 & 2 & 3 & 4 & \ldots & 2k-1 & 2k \\ i_1 & j_1 & i_2 & j_2 & \ldots & i_k & j_k \end{pmatrix}.$$

It can be seen that the definition of Pfaffian of A given above is independent of the order in which the constituent pairs in a partition M are listed. The order in which the elements in a pair are listed is also immaterial, since A is skew-symmetric.

Now suppose that G is a graph whose set of vertices is $\{1, 2, \ldots, n\}$. Let D be an orientation of G such that A is the adjacency matrix of D. Then each nonzero term in the expansion of the Pfaffian of A corresponds to a perfect matching M of G. Thus, if D is such that $\operatorname{sgn}(M)$ is constant, for all perfect matchings M of G, then $|P|$ is the number of perfect matchings of G.

A directed graph D is *Pfaffian* if all perfect matchings of D have the same sign. An undirected graph G is *Pfaffian* if it admits a Pfaffian orientation.

An edge e of a graph G is *admissible* if G has a perfect matching containing e. A graph is *matching covered* if it is connected and each edge is admissible. An edge $e = uw$ of G is admissible if and only if $G - u - w$ has a perfect matching. Thus, one can determine the set of admissible edges of G in polynomial time. The definition of Pfaffian orientation immediately reduces the problem of Pfaffian orientations to matching covered graphs.

Proposition 1.1 *Let G be a graph and H the graph obtained from G by removing every non-admissible edge of G. An orientation D of G is Pfaffian if*

[1] Research supported by Fapesp – Fundação de Amparo à Pesquisa do Estado de São Paulo, proc. #05/04426-6

[2] Email: `alberto.miranda@ic.unicamp.br`

[3] Research supported in part by CNPq — Conselho Nacional de Desenvolvimento Científico e Tecnológico, grant #306088/2005-3

[4] Email: `cll@ic.unicamp.br`

and only if the restriction of D to each connected component of H is Pfaffian.

1.1 Parities of Circuits

The *parity* of a circuit C of even length in a directed graph D is the parity of the number of its edges that are directed in agreement with a specified sense of orientation of C. As C has an even number of edges, the parity is the same in both senses and thus is well defined. If the parity of C is odd we say C is *oddly oriented* in D. For any two sets X and Y, we denote by $X \triangle Y$ the symmetric difference of X and Y.

Theorem 1.2 ([3] Lemma 8.3.1) *Let D be a directed graph. Let M_1 and M_2 be any two perfect matchings of D and let k denote the number of even parity circuits of $D[M_1 \triangle M_2]$. Then, M_1 and M_2 have the same sign if and only if k is even.*

Let G be a graph and H a subgraph of G. The graph H is *conformal* in G if $G - V(H)$ has a perfect matching.

Proposition 1.3 *Let G be a graph, H a conformal subgraph of G, and J a conformal subgraph of H. Then, J is a conformal subgraph of G.*

Theorem 1.4 ([3] Theorem 8.3.2) *Let D be a directed graph, M a perfect matching of D. Then the following are equivalent:*

- *D is Pfaffian;*
- *Every M-alternating circuit of D is oddly oriented;*
- *Every conformal circuit of D is oddly oriented.*

Corollary 1.5 *Let G be a graph, D a Pfaffian orientation of G, and H a conformal subgraph of G. Then, the restriction $D(H)$ of D to H is a Pfaffian orientation.*

1.2 Near Bipartite Graphs

A *single ear* of a connected graph G is a path $P := (v_0, e_1, v_1, \ldots, e_{2k-1}, v_{2k-1})$ of odd length in G whose internal vertices $v_1, v_2, \ldots, v_{2k-2}$, if any, have degree two in G. If P is a single ear of G then we denote by $G - P$ the graph obtained from G by deleting the edges and internal vertices of P. A *double ear* of G is a pair (R_1, R_2), where R_1 and R_2 are two vertex-disjoint single ears of G. An *ear* of G is either a single ear or a double ear of G. If R is an ear of G then we denote by $G - R$ the graph obtained from G by deleting the edges and internal vertices of the constituent paths of R.

Any edge of G spans a single ear and any pair of nonadjacent edges spans a double ear. A graph G is *near-bipartite* if it is non-bipartite, matching covered and it has a double ear R such that $G - R$ is bipartite and matching covered.

Let G be a near-bipartite graph, R a double ear of G such that $G - R$ is bipartite and matching covered. Clearly, each of the two constituent paths of R may be replaced by a single edge, so that the new graph G' is near-bipartite. We shall refer to those two edges as the *removable doubleton* of G'. Moreover, G is Pfaffian if and only if G' is Pfaffian. In the remainder of this article, we shall assume that each ear of R has length one. Thus, we assume that the near-bipartite graph G has two edges e_1 and e_2, such that $G - e_1 - e_2$ is bipartite and matching covered. Clearly, every perfect matching of G that contains one of e_1 and e_2 contains both e_1 and e_2.

2 The Algorithm

Let S be a set of edges of a graph G. We denote by $V(S)$ the set of ends of edges in S.

Theorem 2.1 (Inclusion-Exclusion Theorem) *Let D be a directed near-bipartite graph, R a removable doubleton of D, and Q a conformal circuit of D that contains some edge of R. Then, D is Pfaffian if and only if each of the following three properties holds:*

(i) *$D - R$ is Pfaffian;*

(ii) *$D - V(R)$ is Pfaffian; and*

(iii) *Q is oddly oriented in D.*

Proof. (only if part) The graphs $D - R$, $D - V(R)$ and Q are conformal subgraphs of D. Therefore, if D is Pfaffian then each of the three directed graphs is also Pfaffian. As Q is a circuit, it must be oddly oriented, by Theorem 1.4.

(if part) To prove the converse, assume that the three properties hold. Let $\mathcal{M}$ be the set of perfect matchings of D. By hypothesis, R is a removable doubleton. Then, a perfect matching of D either contains all the edges of R or no edge of R, as mentioned above. Therefore, $\mathcal{M}$ can be partitioned in two sets $\mathcal{M}_R$ and $\mathcal{M}_{\overline{R}}$, the set of perfect matchings of D that contain all the edges of R and the set of those that contain no edge of R, respectively. Property (i) implies that every perfect matching of $\mathcal{M}_{\overline{R}}$ has the same sign s in D. Every M, N-alternating circuit, for $M, N \in \mathcal{M}_R$, is a conformal circuit of $D - V(R)$. Therefore, Property (ii) implies that every perfect matching of

$\mathcal{M}_R$ has the same sign t in D. Circuit Q is conformal in D. So, let M be the union of a perfect matching of $D - V(Q)$ and a perfect matching of Q. Note that M is a perfect matching of D. Let $M' := M \triangle Q$. As Q contains some edge of R, one of M and M' is in $\mathcal{M}_R$ and the other is in $\mathcal{M}_{\overline{R}}$. On the other hand, Property (iii) implies that $\mathrm{sgn}(M) = \mathrm{sgn}(M')$ in D. Therefore, $s = t$. Thus, every perfect matching of D has the same sign. We deduce that D is Pfaffian. □

Theorem 2.2 ([1] Theorem 3.9) *There exists a polynomial time algorithm that, given a matching covered graph G, determines an orientation D of G such that G is Pfaffian if and only if D is a Pfaffian orientation of G.*

The following result was first proved by Vazirani and Yannakakis [7].

Theorem 2.3 ([1] Corollary 3.11) *The problem of determining whether or not a given orientation D of a matching covered graph G is Pfaffian is polynomially reducible to the problem of deciding whether or not G is Pfaffian.*

The following algorithm, discovered in 1998, is due to McCuaig and, independently, to Robertson, Seymour and Thomas. An alternative proof for this algorithm was given by Miranda in his Master's dissertation [5], written under the supervision of Lucchesi.

Theorem 2.4 ([4,6,5]) *There exists a polynomial time algorithm that, given a (possibly non matching covered) bipartite graph G, determines whether G is Pfaffian.*

Corollary 2.5 *There exists a polynomial time algorithm that, given an orientation D of a (possibly non matching covered) bipartite graph G, decides whether D is a Pfaffian orientation.*

Finally, we are in position to describe our algorithm. Let G be a near-bipartite graph. First, apply Theorem 2.2 to obtain an orientation D such that D is Pfaffian if and only if G is Pfaffian, then apply Theorem 2.1, using twice the algorithm mentioned in Corollary 2.5.

References

[1] de Carvalho, M. H., C. L. Lucchesi and U. S. R. Murty, *On the number of dissimilar Pfaffian orientations of graphs*, Theor. Inform. Appl. **39** (2005), pp. 93–113.

[2] Little, C. H. C. and I. Fischer, *A characterisation of Pfaffian near bipartite graphs*, J. Combin. Theory Ser. B **82** (2001), pp. 175–222.

[3] Lovász, L. and M. D. Plummer, "Matching Theory," Number 29 in Annals of Discrete Mathematics, Elsevier Science, 1986.

[4] McCuaig, W., *Pólya's permanent problem*, The Electronic J. of Combin. **11** (2004).

[5] Miranda, A. A. A., "Orientações Pfaffianas e o Furtivo Grafo de Heawood," Master's thesis, Institute of Computing - University of Campinas, http://www.ic.unicamp.br/˜miranda/dissertacao.pdf (2006).

[6] Robertson, N., P. D. Seymour and R. Thomas, *Permanents, Pfaffian orientations and even directed circuits*, Ann. of Math. (2) **150** (1999), pp. 929–975.

[7] Vazirani, V. V. and M. Yannakakis, *Pfaffian orientation of graphs, 0,1 permanents, and even cycles in digraphs*, Discrete Applied, Math. **25** (1989), pp. 179–180.

Algorithmic Aspects of Monophonic Convexity

Mitre C. Dourado[a] Fábio Protti[b,1] Jayme L. Szwarcfiter[c,1]

[a] *ICE, Universidade Federal Rural do Rio de Janeiro and NCE, UFRJ, Brazil.* mitre@nce.ufrj.br

[b] *IM and NCE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.* fabiop@nce.ufrj.br

[c] *IM, NCE and COPPE/Sistemas, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.* jayme@nce.ufrj.br

Abstract

Let G be a graph, and $u, v \in V(G)$. The *monophonic interval* $J[u, v]$ is the set of vertices of all induced paths linking u and v. If $X \subseteq V(G)$, the *monophonic closure* $J[X]$ of X is defined as $J[X] = \bigcup_{u,v \in X} J[u, v]$. In addition, if $X = J[X]$ then X is said to be *monophonically convex* or simply *m-convex*. The *m-convexity number* of G, denoted by $c_m(G)$, is the cardinality of a maximum proper m-convex subset of $V(G)$. The smallest m-convex set containing X is denoted $J_h[X]$ and called *m-convex hull* of X. A subset $X \subseteq V(G)$ is called a *monophonic set* if $J[X] = V(G)$, and an *m-hull set* if $J_h[X] = V(G)$. The *monophonic number* of G, denoted by $m(G)$, is the cardinality of a minimum monophonic set of G, and the *m-hull number* of G, denoted by $h_m(G)$, is the cardinality of a minimum m-hull set of G. In this work we study the complexity of computing the parameters $c_m(G)$, $m(G)$ and $h_m(G)$.

Keywords: monophonic convexity, m-convex set, monophonic set, m-hull set, m-convexity number, monophonic number, m-hull number

[1] Partially supported by the Conselho Nacional de Desenvolvimento Científico e Tec-

1 Introduction

We consider only connected graphs. Let G be a graph, and $u, v \in V(G)$. A path linking u and v is *induced* when there is no edge linking nonconsecutive vertices in the path. For $X \subseteq V(G)$, write $\overline{X} = V(G) \setminus X$. Denote by $N(v)$ the set of neighbors of $v \in V(G)$ and $N_X(v) = N(v) \cap X$. Following the terminology used in [5,8], the *monophonic interval* $J[u, v]$ is the set of vertices of all induced paths linking u and v. If $X \subseteq V(G)$, the *monophonic closure* $J[X]$ of X is defined as $J[X] = \bigcup_{u,v \in X} J[u, v]$. In addition, if $X = J[X]$ then X is said to be *monophonically convex* or simply *m-convex*. Clearly, if $X = \{v\}$ or $X = V(G)$ then X is m-convex. The *m-convexity number* of G, denoted by $c_m(G)$, is the cardinality of a maximum proper m-convex subset of $V(G)$.

The smallest m-convex set containing X is denoted $J_h[X]$ and called *m-convex hull* of X. Note that $X \subseteq J[X] \subseteq J_h[X] \subseteq V(G)$. A subset $X \subseteq V(G)$ is called a *monophonic set* if $J[X] = V(G)$, and an *m-hull set* if $J_h[X] = V(G)$. The *monophonic number* of G, denoted by $m(G)$, is the cardinality of a minimum monophonic set of G, and the *m-hull number* of G, denoted by $h_m(G)$, is the cardinality of a minimum m-hull set of G. It is clear that $h_m(G) \leq \min\{m(G), c_m(G) + 1\}$, for any graph G. Observe also that every monophonic or m-hull set of a graph G must contain all of its simplicial vertices. This implies that every graph containing a simplicial vertex satisfies $c_m(G) = n - 1$.

Although there are many interesting theoretical results on monophonic convexity, few results on its algorithmic aspects are known. For chordal graphs, it is known that $m(G)$ and $h_m(G)$ can be computed in polynomial time, since in such graphs every nonsimplicial vertex lies on a induced path between two simplicial vertices [5].

Besides the monophonic convexity, other types of path convexities are studied, such as the *geodesic convexity*, the *coarsest convexity*, the *triangle-path convexity* and the *total convexity*. See [2,5,8] for details. In particular, some works have investigated algoritmic aspects of the geodesic convexity. The parameters $c_g(G)$, $g(G)$ and $h_g(G)$ are defined by simply replacing "induced paths" by "shortest paths" in the definitions of the corresponding concepts $c_m(G)$, $m(G)$ and $h_m(G)$. Computing any of these parameters is NP-complete for general graphs [3,4,6], while polynomial-time algorithms are known for computing $h_g(G)$ for unit interval graphs [3] and for finding the three param-

nológico - CNPq, and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro - FAPERJ, Brazil.

eters for cographs [3,4]. We note that $c_m(G)$, $m(G)$ and $h_m(G)$ can also be computed in polynomial time for cographs, because for distance-hereditary graphs the geodesic and monophonic convexities are equivalent, and cographs form a subclass of distance-hereditary graphs.

This work is organized as follows. In Section 2, we present a characterization for m-convex sets that leads to a polynomial-time recognition algorithm, and prove that deciding $c_m(G) \geq k$ is NP-complete. In Section 3, the main result is showing that testing whether a set is monophonic is NP-complete, and deciding $m(G) \leq k$ is NP-hard. (No polynomial-time verifiable certificate is known for this problem). In Section 4 we show a polynomial-time algorithm for computing the m-convex hull of a subset; as a corollary, testing whether a subset is an m-hull set can be done in polynomial time. Still in Section 4, we prove that deciding $h_m(G) \leq k$ is NP-complete. Finally, in Section 5 we show that computing $c_m(G)$, $m(G)$ and $h_m(G)$ can be done in polynomial time for cographs.

2 M-convex sets

We start this section by presenting a characterization for m-convex sets.

Theorem 2.1 *Let G be a graph. A subset $X \subseteq V(G)$ is m-convex if and only for every pair of nonadjacent vertices $u, v \in X$ and every connected component C of $G - X$ it holds that $V(C) \cap N_{\overline{X}}(u) = \emptyset$ or $V(C) \cap N_{\overline{X}}(v) = \emptyset$.*

Proof. Assume that X is m-convex. The existence of a pair of nonadjacent vertices $u, v \in X$ and a connected component C of $G - X$ containing a pair u', v' such that $u' \in N_{\overline{X}}(u)$ and $v' \in N_{\overline{X}}(v)$ implies the existence of a sequence of vertices $w_0 = u, w_1 = u', w_2, \ldots, w_{k-1}, w_k = v', w_{k+1} = v$, with $k \geq 1$, such that: (i) $w_i \notin X$, $1 \leq i \leq k$; (ii) either $u' = v'$ or $(w_i, w_{i+1}) \in E(G - X)$, $1 \leq i \leq k$. Hence, there exists an induced path linking u and v and containing at least one vertex outside X, a contradiction.

Assume now that X is not m-convex. Let $w_0 = u, w_1, \ldots, w_k, w_{k+1} = v$ be an induced path linking a pair of vertices $u, v \in X$ such that $k \geq 1$ and $w_i \notin X$ for some $i \in \{1, \ldots, k\}$. Let j be an index such that $w_{j-1} \in X$ and $w_j, w_{j+1}, \ldots, w_i \in \overline{X}$. (Such an index j exists, since $u' \in X$). Analogously, let ℓ be an index such that $w_i, w_{i+1}, \ldots, w_\ell \in \overline{X}$ and $w_{\ell+1} \in X$. This implies that $w_{j-1}, w_{\ell+1}$ is a pair of nonadjacent vertices in X and there is a connected component C of $G - X$ such that $V(C) \cap N_{\overline{X}}(w_{j-1}) \neq \emptyset$ and $V(C) \cap N_{\overline{X}}(w_{\ell+1}) \neq \emptyset$. □

The previous theorem implies the following result:

Theorem 2.2 *Let G be a graph. Deciding whether a subset $X \subseteq V(G)$ is m-convex can be done in polynomial time.*

Proof. The number of pairs of nonadjacent vertices in X is in the worst case $O(|X|^2)$. Computing the connected components of $G - X$ can be done in $O(n + m)$ time, and in the worst case we have $O(n)$ such components. For each pair u, v of nonadjacent vertices in X and each component C of $G - X$, checking whether $V(C) \cap N_{\overline{X}}(u) = \emptyset$ or $V(C) \cap N_{\overline{X}}(v) = \emptyset$ can be done in $O(n + m)$ time. The overall complexity is thus $O((n + m)n|X|^2)$.

If $V(C) \cap N_{\overline{X}}(u) \neq \emptyset$ and $V(C) \cap N_{\overline{X}}(v) \neq \emptyset$ for some triple u, v, C (that is, X is not m-convex), a "no-certificate" (an induced path linking u and v in C) can be constructed in additional linear time. □

To conclude this section, we prove that deciding $c_m(G) \geq k$ is NP-complete.

Theorem 2.3 *Let G be a graph and k a positive integer such that $k < |V(G)|$. Deciding whether $c_m(G) \geq k$ is NP-complete.*

Proof. The problem is in NP because testing whether a subset $X \subseteq V(G)$ with $|V(G)| > |X| \geq k$ is m-convex can be done in polynomial time by Theorem 2.1.

The hardness proof is a reduction from the CLIQUE problem: given a graph H and a positive integer ℓ, decide whether H contains a clique (complete subset) of size at least ℓ. We may assume that $\ell < |V(H)| - 1$. From the graph H, construct G as follows. Define $V(G) = V(H) \cup \{u, v\}$, where u and v are new vertices, and $E(G) = E(H) \cup \{(u, x), (v, x) \mid x \in V(H)\}$. Also, set $k = \ell + 1$.

If $X \subseteq V(H)$ is a clique with size at least ℓ, then $Y = X \cup \{u\}$ is clearly an m-convex subset of G with size at least $\ell + 1$. Conversely, let Y be a proper m-convex subset of G with size at least $\ell + 1$. Observe that Y cannot contain both u and v, for otherwise $Y = V(G)$. This implies that Y cannot contain two nonadjacent vertices w_1 and w_2, for otherwise u and v would necessarily belong to Y. Hence, $Y \backslash \{u, v\}$ is a clique of size at least ℓ in H. □

3 Monophonic sets

Before presenting the main result of this section, we need some preliminary propositions.

Lemma 3.1 *Let u, v, w be three distinct vertices in a graph G. Deciding whether there is an induced path from u to v passing through w is NP-complete.*

The hardness proof of Lemma 3.1 may be done via a reduction from the following NP-complete problem [1]: given two distinct nonadjacent vertices x and y of a graph H, decide whether there is an induced cycle in H passing through x and y. Another way to prove this lemma is to use a parameterized version of the above problem given in [7].

Three important consequences are implied by Lemma 3.1:

Lemma 3.2 *Let u, v, w be three distinct vertices in a graph G. Deciding whether $w \in J[u,v]$ is NP-complete.*

Lemma 3.3 *Let X be a subset of vertices of a graph G, and let $u \in V(G) \backslash X$. Deciding whether $u \in J[X]$ is NP-complete.*

Lemma 3.4 *Let X, Y be vertex subsets of a graph G. Deciding whether $Y \subseteq J[X]$ is NP-complete.*

Hence computing the monophonic interval is computationally hard. The main result of this section follows from the previous lemmas:

Theorem 3.5 *Let Z be a subset of vertices of a connected graph H. Deciding whether Z is a monophonic set of H is NP-complete.*

Proof. The hardness proof is a reduction from the problem stated in Lemma 3.4: define H by adding to G two new vertices a, b which are adjacent to all the vertices in $V(G) \backslash Y$, and set $Z = X \cup \{a, b\}$. Due to the lack of space, the remainder of the proof is omitted. □

The last result of this section is the NP-hardness of deciding $m(G) \leq k$.

Theorem 3.6 *Let G be a graph and k a positive integer. Deciding whether $m(G) \leq k$ is NP-hard.*

4 M-hull sets

Although computing the monophonic closure is hard, as we have seen in the previous section, now we show that computing the m-convex hull of a given subset is polynomial-time solvable.

Theorem 4.1 *Let G be a graph and $X \subseteq V(G)$. Computing $J_h[X]$ can be done in polynomial time.*

Proof. By Theorem 2.1, the m-convexity test for X takes polynomial time. In the affirmative case, $J_h[X] = X$. Otherwise, the test returns an induced path P linking a pair of vertices of X such that $V(P)$ contains a non-empty subset S satisfying $S \subset \overline{X}$. Set $X \leftarrow X \cup S$ and repeat the above procedure.

At most $O(n)$ iterations are needed to complete the computation of $J_h[X]$. □

An important corollary of the previous theorem is:

Corollary 4.2 *Let X be a subset of vertices of a graph G. Deciding whether X is an m-hull set can be done in polynomial time.*

We can relate the computation of the m-convex hull to the parameter $c_m(G)$:

Lemma 4.3 *Let G be a graph and $k \geq 1$ an integer. Then $c_m(G) < k$ if and only if $J_h[S] = V(G)$ for every $S \subsetneq V(G)$ with k vertices.*

The above lemma implies that, for a fixed k, deciding whether $c_m(G) \geq k$ can be done in polynomial time.

We conclude this section by making the following observation:

Corollary 4.4 *Let G be a connected graph and k a positive integer. Deciding whether $h_m(G) \leq k$ is in NP.*

References

[1] D. Bienstock. On the complexity of testing for odd holes and odd induced paths. *Disc. Math.* 90 (1991) 85–92. (Corrigendum: *Disc. Math.* 102,1 (1992) 109.)

[2] M. Changat, H. M. Mulder and G. Sierksma. Convexities related to path properties on graphs. *Disc. Math.* 290 (2005) 117–131.

[3] M. C. Dourado, J. Gimbel, F. Protti and J. L. Szwarcfiter. On the computation of the hull number of a graph (2006), submitted.

[4] M. C. Dourado, F. Protti and J. L. Szwarcfiter. On the computation of some parameters related to convexity of graphs, *Lecture Notes of the Ramanujan Mathematical Society*, to appear.

[5] M. Farber and R. E. Jamison. Convexity in graphs and hypergraphs. *SIAM J. Alg. Disc. Math.* 7,3 (1986) 433–444.

[6] J. G. Gimbel. Some remarks on the convexity number of a graph. *Graphs and Combinatorics* 19 (2003) 357–361.

[7] R. Haas and M. Hoffmann. Chordless paths through three vertices. *Theoret. Comput. Sci.* (2006) 360–371.

[8] I. M. Pelayo. Generalizing the Krein-Milman property in graph convexity spaces: a short survey. *RMS Lecture Notes Series in Mathematics*, to appear.

Colouring clique-hypergraphs of circulant graphs [1]

C. N. Campos [b,2], S. Dantas [a,3], C. P. de Mello [b,4]

[a] *Instituto de Matemática, Universidade Federal Fluminense, Brazil*

[b] *Instituto de Computação, Universidade de Campinas, Brazil*

Abstract

A clique-colouring of a graph G is a colouring of the vertices of G so that no maximal clique of size at least two is monochromatic. The clique-hypergraph, $\mathcal{H}(G)$, of a graph G has $V(G)$ as its set of vertices and the maximal cliques of G as its hyperedges. A vertex-colouring of $\mathcal{H}(G)$ is a clique-colouring of G. Determining the clique-chromatic number, the least number for which a graph G admits a clique-colouring, is known to be NP-hard. We establish that the clique-chromatic number for powers of cycles is equal to two, except for odd cycles of size at least five, that need three colours. For odd-seq circulant graphs, we show that their clique-chromatic number is at most four, and determine the cases when it is equal to two.

Keywords: graph and hypergraph colouring, clique-colouring, circulant graphs.

[1] Partially supported by CNPq 300934/2006-8, 470420/2004-9.

[2] Since April 2007 holds a postdoc position at the University of São Paulo, supported by grant FAPESP 06/60177-8. Email: campos@ime.usp.br

[3] Email: sdantas@cos.ufrj.br

[4] Email: celia@ic.unicamp.br

1 Introduction

A *hypergraph* $\mathcal{H}$ is a pair $(V, \mathcal{E})$ where V is a finite set of vertices and $\mathcal{E}$ is a family of non-empty subsets of V called *hyperedges*. A *k-colouring* of $\mathcal{H}$ is a mapping $\phi : V \to \{1, 2, \ldots, k\}$ such that for each $S \in \mathcal{E}$, with $|S| \geq 2$, there exist $u, v \in S$ with $\phi(u) \neq \phi(v)$, that is, there is no monochromatic hyperedge of size at least two. The *chromatic number* $\chi(\mathcal{H})$ of $\mathcal{H}$ is the smallest k for which $\mathcal{H}$ admits a k-colouring.

Let G be an undirected simple graph with vertex set $V(G)$ and edge set $E(G)$. A *clique* is a set of pairwise adjacent vertices of G. The *clique number* of a graph G, $\omega(G)$, is the greatest integer k for which there exists a clique Q with $|Q| = k$. A *maximal clique* of G is a clique not properly contained in any other clique.

Given a graph G, we define the *clique-hypergraph* $\mathcal{H}(G)$ of G as the hypergraph whose vertices are the vertices of G, and whose hyperedges are the maximal cliques of G. A k-colouring of $\mathcal{H}(G)$ is also called a *k-clique-colouring* of G, and the chromatic number $\chi(\mathcal{H}(G))$ of $\mathcal{H}(G)$ is the *clique-chromatic number* of G. If $\chi(\mathcal{H}(G)) = k$, then G is *k-clique-chromatic*. Note that if $\omega(G) = 2$, then $\mathcal{H}(G) = G$, which implies $\chi(\mathcal{H}(G)) = \chi(G)$.

The clique-hypergraph colouring problem was posed by Duffus et al. [6]. Kratochvil and Tuza [9] have proved that determining the bicolourability of clique-hypergraphs of perfect graphs is NP-hard, but solvable in polynomial time for planar graphs. Additionally, the chromatic number of triangle-free graphs is known to be unbounded [12], and so it is their clique-chromatic number. On the other hand, Bacsó et al. [1] proved that almost all perfect graphs are 3-clique-colourable. Other works considering the clique-hypergraph colouring problem in classes of graphs can be found in the literature [5,7].

We study the clique-hypergraph colouring problem on *circulant graphs*, that are graphs whose adjacency matrix is circulant. This class of graphs has several applications in combinatorics and linear algebra, having been extensively studied over the years [4,11,13]. There are different characterizations of these graphs. For instance, circulant graphs are a particular case of Cayley graphs. We postpone the definition used in this work to the following section.

Determining the clique number and the chromatic number of circulant graphs in general is an NP-hard problem [4]. Here, we study two subclasses of circulant graphs. Motivated by recent works in powers of certain classes of graphs [2,3], the first class considered is *powers of cycles*. We prove that its clique-chromatic number is equal to two, except for C_n, n odd and $n \geq 5$, that needs three colours. The second class considered is *odd-seq circulant graphs*.

For this class, we show that its clique-chromatic number is at most four, and determine the cases when it is equal to two. Also, we verify similar bounds for the chromatic number of these graphs.

2 Preliminaries

Let $d_1, \ldots, d_k$ be a (nonempty) sequence of positive integers satisfying $d_1 \leq \cdots \leq d_k \leq \lfloor n/2 \rfloor$. A *circulant graph* $C_n(d_1, \ldots, d_k)$ is a simple graph with $V(G) = \{v_0, \ldots, v_{n-1}\}$ and $E(G) = E^{d_1} \cup \cdots \cup E^{d_k}$, with $\{v_i, v_j\} \in E^{d_l}$ if, and only if, $d_l = (j - i) \bmod n$. If $e \in E^{d_i}$, then edge e has *reach* d_i. Moreover, if the reach of e is even (odd), then e is called an *even (odd) edge*. An important property of circulant graphs [10] is stated next.

Lemma 2.1 *Let $G = C_n(d_1, \ldots, d_k)$, $d_i \leq \lfloor n/2 \rfloor$. Then, for each d_i, the induced subgraph $C_n(d_i)$ is comprised by $\gcd(n, d_i)$ connected components, each one being a cycle of length $n/\gcd(n, d_i)$,* □

Let G be a circulant graph and Q be a clique of G. If $E(G[Q])$ is comprised by edges of same reach, Q is called a *homogeneous clique*; otherwise it is called *non-homogeneous*. Corollary 2.2 is a consequence of Lemma 2.1.

Corollary 2.2 *Let $G = C_n(d_1, \ldots, d_k)$, $d_i \leq \lfloor n/2 \rfloor$, and Q be a maximal homogeneous clique of G. Then, $|Q| \leq 3$. Moreover, $|Q| = 3$ if, and only if, there exists some d_i such that $n/\gcd(n, d_i) = 3$. In addition, d_i is unique and $d_i = n/3$.* □

3 Powers of cycles

A circulant graph $G = C_n(d_1, \ldots, d_k)$ is a *power of cycles* when $d_1 = 1$, $d_i = d_{i-1} + 1$, $d_k < \lfloor n/2 \rfloor$, and it is denoted C_n^k. We show that the clique-chromatic number of C_n^k is equal to two, except for odd cycles with $n \geq 5$. Note that cycles C_n, i.e., powers of cycles with $k = 1$, have $\chi(\mathcal{H}(C_n)) = 3$ if n is odd and $n \geq 5$; and $\chi(\mathcal{H}(C_n)) = 2$ otherwise.

Let Q be a clique of C_n^k. If every vertex $v_i \in Q$ has even (odd) index, then Q is an *even (odd) clique*. The next two lemmas determine the existence of maximal cliques in a power of cycles that are even or odd.

Lemma 3.1 *Let $G = C_n^k$, $k \geq 2$, n odd, be a power of cycles. Then, there does not exist a maximal clique in G that is even or odd.*

Lemma 3.2 *Let $G = C_n^k$, $k \geq 2$, n even, be a power of cycles. Graph G has a maximal clique that is even or odd if, and only if, k is even and $k = n\left(\frac{i}{2i+1}\right)$, $i \geq 1$, integer.* □

Theorem 3.3 *Let $G = C_n^k$, $k \geq 2$, be a power of cycles. Then, $\chi(\mathcal{H}(G)) = 2$.*

Sketch of the proof: We consider two cases and in each case we construct π, a 2-clique colouring for G.

If n is even with $k \neq n\left(\frac{i}{2i+1}\right)$, or n is odd, then $\pi(v_i) := i \bmod 2$. By Lemma 3.1 and by Lemma 3.2, we conclude that there are no maximal clique in G that is even or odd. Therefore, each maximal clique of G has at least one vertex of even index and one vertex of odd index.

If n is even and $k = n\left(\frac{i}{2i+1}\right)$, then π is defined as: $\pi(v_i) = 0$, if $0 \leq i \leq \lfloor n/3 \rfloor - 1$; $\pi(v_i) = 1$, if $\lfloor n/3 \rfloor \leq i \leq 2\lfloor n/3 \rfloor - 1$; and $\pi(v_i) = i \bmod 2$, if $2\lfloor n/3 \rfloor \leq i \leq n-1$. In this case we prove that each maximal clique comprised by $k+1$ consecutive vertices (modulo n) are not monochromatic. Afterwards, we consider the maximal cliques comprised by non-consecutive vertices and conclude that these ones are also non-monochromatic. □

4 Odd-Seq Circulant graphs

A circulant graph $C_n(d_1, \ldots, d_k)$ is an *odd-seq circulant graph* when each d_i, $1 \leq d_i \leq d_k$, is odd. We analyse two cases depending on the parity of n.

Let $G = C_n(d_1, \ldots, d_k)$ be an odd-seq circulant graph with n even. These graphs were shown bipartite by Heuberger [8]. Therefore, $\omega(G)$ is 2 and each maximal clique in G is maximum. We conclude that there exists a 2-clique colouring for odd-seq circulant graphs with n even.

Consider now $G = C_n(d_1, \ldots, d_k)$, an odd-seq circulant graph with n odd. We analyse some cases, according to the clique number of G. Lemma 4.1 establishes conditions for which a graph G has $\omega(G)$ equal to 2 or 3. The next result is a structural property of odd-seq circulant graphs.

Property 4.1 *Let $G = C_n(d_1, \ldots, d_k)$ be an odd-seq circulant graph with n odd. Then, each cycle of size 3 has at least one vertex v_i, with $0 \leq i \leq \lfloor n/2 \rfloor$, and at least one vertex v_j, with $\lceil n/2 \rceil \leq j \leq n-1$.* □

Lemma 4.1 *Let $G = C_n(d_1, \ldots, d_k)$ be an odd-seq circulant graph with n odd. Then, $\omega(G) = 3$ if, and only if, there exist $d_i, d_j, d_l \in \{d_1, \ldots, d_k\}$, not necessarily distinct, such that $d_i + d_j + d_l = n$. Otherwise, $\omega(G) = 2$.*

Proof. Let $V(G) = \{v_0, \ldots, v_{n-1}\}$. First, we prove that every clique of G has size at most 3. Suppose that Q is a clique in G and that $|Q| = 4$. Adjust notation so that $v_0 \in Q$. Let v_i, v_j, v_k be the other vertices of Q. Assume that $i < j < k$. Because set Q is a clique, $(v_0, v_i, v_j, v_k, v_0)$ is a cycle (not induced) in G. Since every edge of G is odd, we conclude that i and k are odd and j is even. However, for v_l adjacent to v_0, if l is odd, then $l \leq \lfloor n/2 \rfloor$; otherwise $l \geq \lceil n/2 \rceil$. Therefore, we conclude that $i, k \leq \lfloor n/2 \rfloor$ and $j \geq \lceil n/2 \rceil$, a contradiction, since $i < j < k$.

Now, assume that $Q = \{v_i, v_j, v_l\}$ is a clique in G. Adjust notation so that $0 \leq i < j < l \leq n-1$. By Property 4.1, we can assume that $i \leq \lfloor n/2 \rfloor$ and $l \geq \lceil n/2 \rceil$. Let d_i, d_j and d_l be the reaches of $v_i v_j$, $v_j v_l$, and $v_l v_i$, respectively. Thus, $d_i = j - i$, $d_j = l - j$ and $d_l = n - l + i$. Therefore, $d_i + d_j + d_l = n$.

Consider now that there exist $d_i, d_j, d_l \in \{d_i, \ldots, d_k\}$ such that $d_i + d_j + d_l = n$. Edges $v_0 v_{d_i}$, $v_{d_i} v_{d_i + d_j}$ and $v_{d_i + d_j} v_{d_i + d_j + d_l}$ belong to $E(G)$. Since $d_i + d_j + d_l = n$, we have that $v_{d_i + d_j + d_l} = v_0$. Therefore, $(v_0, v_{d_i}, v_{d_i + d_j}, v_0)$ is a cycle and $\{v_0, v_{d_i}, v_{d_i + d_j}\}$ is a clique in G. We remark that, whenever $d_i = d_j = d_l$, we have a homogeneous clique in G.

In order to conclude the proof note that we have already proved that $\omega(G) \leq 3$. However, if $\omega(G) \neq 3$, then $\omega(G) = 2$ because $E(G) \neq \emptyset$. □

We proceed considering first odd-seq circulant graphs G with $\omega(G) = 3$ for which every maximal clique is also maximum. Afterwards, we assume that G has maximal cliques of size two and establish bounds to the clique-chromatic number of G in this case. We close this section with Corollary 4.4, that extends the bounds of Theorem 4.3 to the chromatic number of odd-seq circulant graphs.

Theorem 4.2 *Let $G = C_n(d_1, \ldots, d_k)$ be an odd-seq circulant graph with n odd. If every maximal clique of G has size three, then $\chi(\mathcal{H}(G)) = 2$.* □

Theorem 4.3 *Let $G = C_n(d_1, \ldots, d_k)$ be an odd-seq circulant graph with n odd. If G has maximal cliques of size two, then $3 \leq \chi(\mathcal{H}(G)) \leq 4$.*

Proof. We start by showing that $\chi(\mathcal{H}(G)) \leq 4$. Consider the following 4-colour assignment π to vertices of G: $\pi(v_i) = i \mod 2$, if $0 \leq i \leq \lfloor n/2 \rfloor$; and $\pi(v_i) = 2 + (i \bmod 2)$, if $\lceil n/2 \rceil \leq i \leq n-1$. The validity of π follows from the fact that vertices of same colour are non-adjacent.

It remains to show that $\chi(\mathcal{H}(G)) \geq 3$. Let $Q = \{u, v\}$ be a maximal clique. By definition of odd-seq circulant graphs, edge uv belongs to an odd cycle C. By symmetry of circulant graphs, uv can be any edge of C. By maximality of Q, $|C| > 3$. Therefore, in order to construct a clique-colouring for this cycle at least three colours are needed. □

Corollary 4.4 *If G is an odd-seq circulant graph with n odd, then $3 \leq \chi(G) \leq 4$.* □

It is important to notice that the bounds obtained in Theorem 4.3 are tight. For instance, $G = C_{21}(1,5,9)$ has $\chi(\mathcal{H}(G)) = 4$ and $G = C_{21}(1,3,7)$ has $\chi(\mathcal{H}(G)) = 3$.

References

[1] Bacsó, G., S. Gravier, A. Gyárfás, M. Preissmann and A. Sebö, *Coloring the maximal cliques of graphs*, SIAM J. Discrete Math. **17** (2004), pp. 361–376.

[2] Brandstädt, A., F. F. Dragan and F. Nicolai, *LexBFS-orderings and powers of chordal graphs*, Discrete Math. **171** (1997), pp. 27–42.

[3] Campos, C. N. and C. P. de Mello, *A result on the total colouring of powers of cycles*, Discrete Appl. Math. **155** (2007), pp. 585–597.

[4] Codenotti, B., I. Gerace and S. Vigna, *Hardness results and spectral techniques for combinatorial problems on circulant graphs*, Linear Algebra Appl. **285** (1998), pp. 123–142.

[5] Défossez, D., "Coloration d'hypergraphes et clique-coloration," Ph.D. thesis, Université Joseph-Fourier - Grenoble I (2006).

[6] Duffus, D., B. Sands, N. Sauer and R. E. Woodrow, *Two-colouring all two-element maximal antichains*, J. Comb. Theory Ser. A **57** (1991), pp. 109–116.

[7] Gravier, S., C. T. Hoàng and F. Maffray, *Coloring the hypergraph of maximal cliques of a graph with no long path*, Discrete Math. **272** (2003), pp. 285–290.

[8] Heuberger, C., *On planarity and colorability of circulant graphs*, Discrete Math. **268** (2003), pp. 153–169.

[9] Kratochvíl, J. and Z. Tuza, *On the complexity of bicoloring clique hypergraphs of graphs*, J. Algorithms **45** (2002), pp. 40–54.

[10] Meidanis, J., *Edge coloring of cycle powers is easy*, available at http://www.ic.unicamp.br/~meidanis. Last visited 21/08/2007.

[11] Muzychuk, M. and G. Tinhoffer, *Recognizing circulant graphs of prime order in polynomial time*, The Electronic Journal of Combinatorics **3** (1998).

[12] Mycielski, J., *Sur le coloriage des graphes*, Colloq. Math. **3** (1955), pp. 161–162.

[13] Obradoviç, N., J. Peters and G. Ruzic, *Minimum chromaticity of circulant graphs*, Discrete Math. **299** (2005), pp. 288–296.

The nonidealness index of circulant matrices[1]

Gabriela R. Argiroffo[2]

Departamento de Matemática.
Facultad de Ciencias Exactas, Ingeniería y Agrimensura.
Universidad Nacional de Rosario. Rosario, Argentina

Silvia M. Bianchi[3]

Departamento de Matemática.
Facultad de Ciencias Exactas, Ingeniería y Agrimensura.
Universidad Nacional de Rosario. Rosario, Argentina

Abstract

Ideal matrices are precisely those matrices M where the set covering polyhedron $Q^*(M)$ equals the polyhedron $Q(M) = \{x : Mx \geq 1, x \geq 0\}$. In a previous work (2006) we defined a nonidealness index equivalent to $\max\{t : Q(M) \subset tQ^*(M)\}$. Given an arbitrary matrix M the nonideal index is NP-hard to compute and for most matrices it remains unknown.

A well known family of minimally nonideal matrices is the one of the incidence matrices of chordless odd cycles. A natural generalization of them is given by circulant matrices. Circulant ideal matrices have been completely identified by Cornuéjols and Novick (1994). In this work we obtain a bound for the nonidealness index of circulant matrices and determine it for some particular cases.

Keywords: set covering polyhedron, circulant matrix, nonidealness index.

1 Introduction

Given a $n \times m$ $0,1$ matrix M without domination rows, we denote by $Q(M)$ the polyhedron

$$Q(M) = \{x \in \mathbb{R}^n : Mx \geq \mathbf{1},\ x \geq 0\},$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector with all entries at value one. If $conv(X)$ denotes the convex hull of X, the *set covering polyhedron* of M is defined as

$$Q^*(M) = conv(Q(M) \cap \mathbb{Z}^n).$$

A matrix M is *ideal* if $Q^*(M) = Q(M)$.

Contracting $j \in \{1, \dots, n\}$ means that column j is removed from M as well as the resulting dominating rows and hence, corresponds to setting $x_j = 0$ in the constraints $Mx \geq \mathbf{1}$. Deleting $j \in \{1, \dots, n\}$ means that column j is removed from M as well as all the rows with a 1 in column j and this corresponds to setting $x_j = 1$ in the constraints $Mx \geq \mathbf{1}$. We denote by M/j and $M \setminus j$ the matrices obtained by contraction and deletion of column j respectively. Then, given M and $V_1, V_2 \subset \{1, \dots, n\}$ disjoint, we will say that $M/V_1 \setminus V_2$ is a *minor* of M. If a matrix is ideal then so are all its minors [3].

Given a $m \times n$ matrix M and a column $j \in \{1, \dots, n\}$, let us consider the matrix obtained from M by *duplication* of j, as the matrix $M * j$ having $n+1$ columns, whose rows are constructed as follows: for every row m of M such that $m_j = 0$ there is a row m' of $M * j$ defined as: $m'_i = m_i$ for all $i = 1, \dots, n$ and $m'_{n+1} = 0$, and for every row m of M such that $m_j = 1$ there are two rows m' and m'' of $M * j$ defined as: $m'_i = m''_i = m_i$ for all $i = 1, \dots, n$ when $i \neq j$, $m'_j = 1$, $m'_{n+1} = 0$ and $m''_j = 0$, $m''_{n+1} = 1$. A matrix obtained from M by a sequence of deletions and duplications is a *parallelization* of M, and it is easy to check that the order in which these operations are performed is irrelevant. Then, parallelizations of M can be associated with vectors $w \in \mathbb{Z}^n_+$ in the following way: M^w is the matrix obtained by deletion of columns i with $w_i = 0$ and duplicating $w_i - 1$ times any column with $w_i \geq 1$.

A *cover* of M is a $0,1$ $n-$dimensional vector in $Q(M)$ and the *covering number* of M is

$$\tau(M) = \min\{\mathbf{1}x : x \text{ cover of } M\},$$

[1] Partially supported by CONICET grant PIP 5810

[2] Email: garua@fceia.unr.edu.ar

[3] Email: sbianchi@fceia.unr.edu.ar

and clearly $\tau(M) = \min\{\mathbf{1}x : x \in Q^*(M)\}$. We denote by $\tau_f(M)$ the *fractional covering number of* M, defined by

$$\tau_f(M) = \min\{\mathbf{1}x : x \in Q(M)\} = \min\{\mathbf{1}x : Mx \geq \mathbf{1}\}.$$

For any $0,1$ matrix M we have that $Q^*(M) \subset Q(M)$ and trivially $\tau_f(M) \leq \tau(M)$. It is known (see [3]) that a matrix M is ideal if and only if $\tau_f(M^w) = \tau(M^w)$ for all $w \in \mathbb{Z}^n_+$.

The blocker of M is the matrix denoted by $b(M)$ whose rows are the minimal covers of M. Also (see [3]), $b(b(M)) = M$, $b(M/i) = b(M) \setminus i$ and $b(M \setminus i) = b(M)/i$ and, in [8], Lehman proved that a matrix M is ideal if and only if its blocker is.

A *matching* in a matrix M is a $0,1$ vector y such that $yM \leq \mathbf{1}$, and the *matching number* $\nu(M)$ is

$$\nu(M) = \max\left\{y\mathbf{1} : yM \leq \mathbf{1}, y \in \mathbb{Z}^m_+\right\}.$$

Defining $\nu_f(M)$, the *fractional matching number of* M as

$$\nu_f(M) = \max\left\{y\mathbf{1} : yM \leq \mathbf{1}, y \in \mathbb{R}^m_+\right\},$$

by linear programming duality we have $\tau_f(M) = \nu_f(M)$.

Therefore, given $w \in \mathbb{Z}^n_+$ we have that $0 < \frac{\nu_f(M^w)}{\tau(M^w)} \leq 1$. It follows that M is ideal if and only if $\frac{\nu_f(M^w)}{\tau(M^w)} = 1$ for all $w \in \mathbb{Z}^n_+$. Equivalently, M is not ideal if and only if there exists some $w \in \mathbb{Z}^n_+$ for which $\frac{\nu_f(M^w)}{\tau(M^w)} < 1$. We introduced in [2] the *nonidealness index* of a matrix M, denoted by $ini(M)$, as follows:

$$ini(M) = \inf\left\{\frac{\nu_f(M^w)}{\tau(M^w)} \text{ for all } w \in \mathbb{Z}^n_+,\ w \neq 0\right\}.$$

This nonidealness index has equivalent definitions

Theorem 1.1 **([2])** *For any matrix* M*,*

$$ini(M) = \max\{r \in \mathbb{R} : Q(M) \subset rQ^*(M)\} = \min\{xy : x \in Q(M), y \in Q(b(M))\}.$$

In addition, the nonidealness index has the following properties:

Theorem 1.2 **([2])** *For any matrix* M*,*

(i) $ini(M) = ini(b(M))$*.*

(ii) For every minor M' *of* M *,* $ini(M) \leq ini(M')$*.*

(iii) $ini(M) \leq \frac{n}{\tau(M)\tau(b(M))}$*.*

A matrix M is *minimally nonideal* (*mni*, for short), if it is not ideal but all its proper minors are. A starting point of the study of *mni* matrices is

Lehman's work (see [8] and [9]). In particular, in [8], Lehman proved that the incidence matrices of chordless odd cycles C_n^2 and their blockers are *mni*.

A natural generalization of this family of matrices is given by *circulant* matrices, denoted by C_n^k and defined as matrices having n columns and whose rows are the incidence vectors of $\{i, i+1, \ldots, i+k-1\}$ for $i \in \{1, \cdots, n\}$, where additions are taken modulo n. Actually, it is known that:

Theorem 1.3 ([4], [8]) *Let k and n be integer numbers such that $2 \leq k \leq n-2$, then*

i) *the only ideal circulant matrices are C_6^3, C_9^3, C_8^4 and C_n^2, for even $n \geq 4$.*

ii) *the only* mni *circulant matrices are C_n^2, for odd $n \geq 3$ and*

$$C_5^3, C_8^3, C_{11}^3, C_{14}^3, C_{17}^3, C_7^4, C_{11}^4, C_9^5, C_{11}^6, C_{13}^7.$$

As the equivalent definitions of the nonidealness index given in Theorem 1.1, involve the polyhedral structure of $Q(M)$ we will take into account the following result that completely describes the fractional extreme points of $Q(M)$ when M is any circulant matrix.

Theorem 1.4 ([1]) *Let $\bar{x} \in Q(C_n^k)$ and $N_{\bar{x}} = \{i \in \{1, \cdots, n\} : x_i = 0\}$. The point $\bar{x}$ is a fractional extreme point of $Q(C_n^k)$ if and only if $C_n^k / N_{\bar{x}}$ is isomorphic to $C_{n'}^{k'}$, where n' y k' are relative prime numbers. Moreover, $\bar{x}$ is*

$$\bar{x}_i = \begin{cases} \frac{1}{k'} & i \notin N_{\bar{x}}, \\ 0 & i \in N_{\bar{x}}. \end{cases}$$

2 The nonidealness index of circulant matrices

Let M be a matrix having M_i $i = 1, \ldots, s$ as minors. Combining (ii) and (iii) of Theorem 1.2, we obtain

$$ini(M) \leq \min \left\{ \frac{n_i}{\tau(M_i)\tau(b(M_i))}, i = 1, \ldots, s \right\}, \tag{1}$$

where n_i is the number of columns of matrix M_i for $i = 1 \ldots, s$.

A natural question is *for which nonideal matrices we obtain equality?*

When a matrix M is *mni* it is known [3], that $Q(M)$ has a unique fractional extreme point. In addition, in the case of a *mni* circulant matrix C_n^k, this point is $\frac{1}{\tau(b(M))}\mathbf{1} \in \mathbb{R}^n$. Then, after Theorem 1.1, it is not hard to see that

$$ini(C_n^k) = \frac{n}{\tau(b(C_n^k))\tau(C_n^k)}.$$

As every circulant matrix C_n^k has $\tau(b(M)) = k$ and $\tau(M) = \lceil \frac{n}{k} \rceil$, we obtain that the nonidealness index of a *mni* circulant matrix C_n^k is

$$ini(C_n^k) = \frac{n}{k \lceil \frac{n}{k} \rceil}.$$

We will see that *mni* circulant matrices are not the only ones for which equality in equation (1) holds.

Let C_n^k be a nonideal circulant matrix and from now on let $\mathcal{C}$ be the set of nonideal circulant minors $C_{n'}^{k'}$ of C_n^k with n' and k' relative prime numbers. Let $B = b(C_n^k)$ and $B' = b(C_{n'}^{k'})$ for $C_{n'}^{k'} \in \mathcal{C}$. Hence, from Theorem 1.4 and blocking duality (see [6]),

$$Q^*(B) = Q(B) \cap \{x : \sum_{j \in \{1,\ldots,n'\}} x_j \geq \tau(B'), b(B') \in \mathcal{C}\}.$$

With the help of Theorem 1.1 we obtain that

Theorem 2.1 *If $B = b(C_n^k)$, then $ini(B) = \min\{ini(B') : b(B') \in \mathcal{C}\}$.*

From the previous result and Theorem 1.2, and exploting algebraic properties of circulant matrices, we have

Theorem 2.2 *Let C_n^k be a circulant matrix. Then $ini(C_n^k) = \min\{ini(C_{n'}^{k'}) : C_{n'}^{k'} \in \mathcal{C}\}$. Moreover, if the minimum in the previous equation is achieved for $C_{n_0}^{k_0}$, then*

$$ini(C_n^k) = ini(C_{n_0}^{k_0}) = \frac{n_0}{k_0 \lceil \frac{n_0}{k_0} \rceil}.$$

Then we can ensure that for every circulant matrix equality (1) holds. Moreover,

Theorem 2.3 *Let C_n^k be a circulant matrix. If for every $C_{n'}^{k'} \in \mathcal{C}$ is $\lceil \frac{n'}{k'} \rceil = \lceil \frac{n}{k} \rceil$, then $ini(C_n^k) = \frac{n}{k \lceil \frac{n}{k} \rceil}$.*

In fact,

Corollary 2.4 *If $k \geq \lfloor \frac{2}{3} n \rfloor$ then, $ini(C_n^k) = \frac{n}{k \lceil \frac{n}{k} \rceil}$.*

On the other hand, observe that when $n = sk$, the minor $C_{n'}^{k'}$ in Theorem 2.2 is a proper minor of C_{sk}^k.

Theorem 2.5 *Let C_{sk}^k be a circulant matrix with $s, k \geq 3$, then*

$$ini(C_{sk}^k) = ini(C_{(k-1)s+1}^{k-1}).$$

Many polydedral aspects associated with the stable set polytopes of webs and antiwebs have their counterpart in the set covering polyhedra associated with circulant matrices and their blockers. As we have pointed out in [1], some polyhedral results can be transferred from one problem to the other. In fact, an analogous result to Theorem 2.2 has been established by Coulonges et al. in [5] for webs and antiwebs using an imperfection index due to Gerke and McDiarmid [7] which can be viewed as the symmetric concept of the nonidealness index used in this work.

The connection between properties of nonideal and imperfect matrices is an interesting task to study and these results, in particular, offer a promising line for further research.

References

[1] G. R. Argiroffo, S. M. Bianchi, *On the set covering polyhedron of circulant matrices*, preprint.

[2] G. R. Argiroffo, S. M. Bianchi, G. L. Nasini, *Clutter nonidealness*, Discrete Applied Mathematics, **154** (2006), 1324–1334.

[3] G. Cornuéjols, *Combinatorial Optimization: Packing and Covering*, SIAM CBMS 74 (2001), 132pp.

[4] G. Cornuéjols, B. Novick *Ideal* $0,1$ *Matrices*, Journal of Combinatorial Theory B **60** (1994), 145–157.

[5] S. Coulonges, A. Pêcher, A. Wagler, *Characterizing and bounding the imperfection ratio for some classes of graphs*, to appear in Mathematical Programming A.

[6] D. Fulkerson, *Blocking polyhedra*, B. Haris (Ed.) Graph Theory and its Applications, Academic Press, New York (1970), 93–112.

[7] S. Gerke, C. McDiarmid, *Graph imperfection*, Journal of Combinatorial Theory (B) **83** (2001), 58–78.

[8] A. Lehman, *On the width-length inequality*, Mathematical Programming **17** (1979), 403–417.

[9] A. Lehman, *On the width-length inequality and degenerate projective planes*, In W. Cook and P. D. Seymour, editors, Polyhedral Combinatorics, Volume 1 of DIMACS Series in Discrete Math. and Theoretical Computer Science, (1990), 101–105.

Clique-coloring UE and UEH graphs

Márcia R. Cerioli [a,1] and Priscila Petito [b,2]

[a] *Instituto de Matemática and COPPE/Sistemas e Computação,*
Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

[b] *COPPE/Sistemas e Computação,*
Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

Abstract

We consider the clique-coloring problem, that is, coloring the vertices of a given graph such that no maximal clique of size at least two is monocolored. More specifically, we investigate the problem of giving a class of graphs, to determine if there exists a constant C such that every graph in this class is C-clique-colorable. We consider the classes of UE and UEH graphs.

A graph G is called an UE graph if it is the edge intersection graph of a family of paths in a tree. If this family satisfies the Helly Property we say that G is an UEH graph.

We show that every UEH graph is 3-clique-colorable. Moreover our proof is constructive and provides a polynomial-time algorithm. We also describe, for each $k \geq 2$, an UE graph that is not k-clique-colorable. The UE graphs form one of the few known classes for which the clique-chromatic number is unbounded.

Keywords: Clique-coloring; maximal cliques; intersection graphs; UE graphs; UEH graphs.

[1] Partially supported by CAPES, CNPq and FAPERJ. Email: `cerioli@cos.ufrj.br`

[2] Supported by a CAPES fellowship. Email: `ppetito@cos.ufrj.br`

1 Introduction

Let $G = (V, E)$ be a graph. A *clique* of G is a maximal complete induced subgraph of G. We identify a clique with its vertex set. We denote by $\mathcal{C}$ the set of all cliques of a given graph. K_n denotes the complete graph with n vertices, C_n is the chordless cycle of length n, and S_n is a star with n spokes, that is a tree with one universal vertex of degree n and n leaves.

A *k-clique-coloring* of a graph G is a function $c : V(G) \to \{1, ..., k\}$, such that for every clique C of G with at least 2 vertices, there are $u, v \in C$ such that $c(u) \neq c(v)$. We say that G is *k-clique-colorable* if such a function exists. The *clique-chromatic number* of G is the smallest k for which G is k-clique-colorable. As usual, a *clique-coloring* is a k-clique-coloring for some k.

Clique-coloring has some similarities with usual coloring (that is coloring the vertices of a graph such that two adjacent vertices get different colors), for example, any coloring is also a clique-coloring, and optimal colorings and clique-colorings coincide in the case of triangle-free graphs. But there are also essential differences, for example, a clique-coloring of a graph may not be a clique-coloring for its subgraphs. Subgraphs may have a greater clique-chromatic number than the original graph. Another difference is that even a 2-clique-colorable graph can contain an arbitrarily large clique. Actually, the 2-clique-coloring problem is NP-Complete. This result holds even for perfect graphs [4] which are however known for their good algorithmic properties.

Since the chromatic number of triangle-free graphs is known to be unbounded [2] we get that this is also the case for the clique-chromatic number. On the other hand, it is conjectured that there exist a constant C such that every perfect graph is C-clique-colorable (see [1]). Although this conjecture remains open so far, several subclasses have positive answer and mostly of the obtained results prove that for special classes of perfect graphs C is equal to either 2 or 3.

It is not hard to find a bound for the well known class of chordal graphs, that is, graphs that do not contain C_n, $n \geq 4$ as subgraphs. In fact, every chordal graph is 2-clique colorable.

We are interested in graph classes defined as the intersection graphs of a family of paths in a tree. More specifically, the *intersection graph on edges* of a family of paths $\mathcal{P}$ (not necessarily containing all paths) in a tree is the graph G with a vertex for each path in $\mathcal{P}$ and an edge between two different vertices if and only if the correspondent paths have a common edge. A graph G is an *UE graph* if there exists a family of paths $\mathcal{P}$ in a tree T such that G is the intersection graph on edges of $\mathcal{P}$. A such pair $(\mathcal{P}, T)$ is called an

UE representation of G. These graphs were introduced as *EPT graphs* in [3]. Figure 1 shows a graph G and two different UE representations of it.

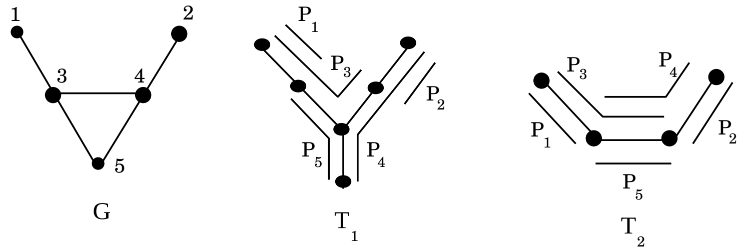


Fig. 1. An UE graph G and two of its UE representations.

A family of sets *satisfies the Helly property* when every one of its subfamilies whose members pairwise intersect has a common element. A graph G is an *UEH graph* if it has an UE representation $(\mathcal{P}, T)$ such that $\mathcal{P}$ satisfies the Helly property. In this case, we say that $(\mathcal{P}, T)$ is an *UEH representation* of G. The graph G in Figure 1 is an UEH graph. The Hajós graph (graph G_e in [5]) is an UE graph which is not UEH.

The UE graphs model conflict of message routes in communication networks. A x-y-path corresponds to a message route from x to y. We say that two paths are conflicting when they use some same link, i.e., when they have a common edge. The intersection graph of this family of paths with the proviso that the paths are given by its edge sets, models this conflict situation as proposed by Golumbic and Jamison in [3] where the class of UE graphs was by the first time considered.

The graph class UV is defined when we consider that the paths in a tree are given by its vertex sets instead of edge sets. Monma and Wei in [5] present an extensive and comparative study between all these classes of intersection graphs of paths in a tree. Since every UV graph is chordal, UV graphs are 2-clique-colorable as well.

In Section 2 we show that every UEH graph is 3-clique-colorable. In Section 3 we present a family of UE graphs containing, for all $k \geq 2$, a graph that is not k-clique-colorable.

2 Clique-coloring UEH graphs

UEH graphs can be characterized in terms of a very special UE representation, as shown in Theorem 2.1, where $\mathcal{C}_v$ denotes the set of cliques in $\mathcal{C}$ containing the particular vertex v and $T(\mathcal{C}_v)$ denotes the subgraph of T whose edges correspond to $\mathcal{C}_v$. A tree satisfying Theorem 2.1 is a *clique tree* of G. Tree T_2 in Figure 1 is a clique tree of G.

Theorem 2.1 **([5])** *A graph $G = (V, E)$ is UEH if and only if there exists a tree T with edge set $\mathcal{C}$, such that for every $v \in V$, $T(\mathcal{C}_v)$ is a path in T.*

A clique C is a *clique separator* of a connected graph G if $G \setminus C$ is not connected. Monma and Wei characterized the UEH graphs that do not have clique separators. More specifically, these graphs are the ones having an UEH representation $(\mathcal{P}, T)$ in which T is a star [5]. The theorem below implies that the clique-coloring problem is completely solved for this class.

Theorem 2.2 *If G is a connected UEH graph without clique separators, then G is 3-clique-colorable. Moreover G is 3-clique-chromatic if and only if $G = C_k$, for some odd k, $k > 3$.*

In a clique tree, every edge not incident to a leaf corresponds to a clique separator of G. Moreover, assign colors to paths of an UEH representation given by a clique tree is the same as assign colors to vertices in G. For each edge of the clique tree, there are two paths having different colors sharing this edge if and only if the corresponding coloring in G is a clique-coloring. Based on these facts we have that the clique-chromatic number is bounded for UEH graphs.

Theorem 2.3 *If G is an UEH graph then G is 3-clique-colorable.*

Proof. Whithout loss of generality, assume that G is connected. Let $(\mathcal{P}, T)$ be an UEH representation of G where T is a clique tree. The proof is by induction on the number of clique separators of G. If G has no clique separator, then T is a star and, by Theorem 2.2 we have that G is 3-clique-colorable.

Now, suppose that T is not a star. Then there exists an edge $e = (v, w)$ in T such that all neighbors of w, except v, are leaves. Consider T' and T'' the subtrees of T such that T' is induced by v, w and all vertices in T that are not neighbors of w, and T'' is the star induced by w and its neighbors in T.

Let $\mathcal{P}'$ and $\mathcal{P}''$ be the subsets of $\mathcal{P}$ such that P is in $\mathcal{P}'$ if and only if P has an edge of T', and P is in $\mathcal{P}''$ if and only if P has an edge of T''. Let G' and G'' be the intersection graphs on edges of $\mathcal{P}'$ and $\mathcal{P}''$, respectively.

By the induction hypothesis, G' is 3-clique-colorable. Take the partial coloring of $\mathcal{P}$ that corresponds to the given assignment of colors in $\{1, 2, 3\}$ to the vertices of G'. The paths in $\mathcal{P}''$ that have the edge e are already colored. To extend this to the whole set of paths, we first classify the edges in $T'' - e$.

An edge is type 2 when there exist two paths in $\mathcal{P}''$, both containing this edge, having different colors; an edge is type 1 if all the colored paths in $\mathcal{P}''$ that contain this edge have the same color and there exists at least one such path; finally when none of the paths containing an edge is colored, we say that it is type 0. Set A_i, $i \in \{0, 1, 2\}$, as the set of edges of type i. Since each edge in T corresponds to a clique of G, when an edge $a \in A_1$ there exists a non-colored path $P \in \mathcal{P}''$ containing a. This property is an invariant to the procedure below. To assign colors to the paths in $\mathcal{P}''$ that are not yet colored we proceed as follows:

Step 1: While $A_1 \neq \emptyset$ choose an edge $a \in A_1$ and a non-colored path P_a in $\mathcal{P}''$ that contains a. Suppose, without loss of generality, that all colored paths that contains a are in color 1. If P_a consists only of the edge a, then assign color 2 to P_a and move a to A_2. Otherwise, let $P_a = (a, b)$. If b is type 2, proceed as in the previous case. If b is type 0, then proceed as in the previous case but also move b to A_1. Now, assume b is type 1. If all colored paths in $\mathcal{P}$ that contains b are in color 1, then assign color 2 to P_a, else assign the missing color to P_a and, in any case move both a and b to A_2.

Step 2: If $A_0 \neq \emptyset$, choose a non-colored path P that contains a type 0 edge. Assign color 1 to P, move all type 0 edges in P to A_1, and go back to Step 1.

Step 3: If $A_0 = \emptyset$, then assign color 1 to every non-colored path that still remains.

Since $A_1 = \emptyset$ when Step 3 is reached, for each edge in T'', there exist two paths in different colors that contain this edge. Therefore, G is 3-clique-colorable. □

The proof of Theorem 2.3 leads to a polynomial-time algorithm to obtain a 3-clique-coloring of an UEH graph. This greedy algorithm may use 3 colors in graphs that are in actual fact 2-clique-chromatic. We observe also that there exist 3-clique-chromatic UEH graphs, whose subgraphs G' and G'' as specified in the proof of the theorem are both 2-clique-chromatic.

3 On the clique-coloring problem for UE graphs

For each n, we consider the very special UE graph G_n defined by the UE representation $(\mathcal{P}_n, S_n)$, where S_n is a star and $\mathcal{P}_n$ is the set of all possible two-edges paths of S_n.

In this way G_n has exactly $\binom{n}{2}$ vertices, each one labeled by an unordered pair $ab \subseteq \{1, \ldots, n\}$. Notice that G_n has $\binom{n}{3}$ cliques that are triangles, thus each k-clique-coloring of G_n corresponds to an edge coloring of the complete graph K_n in such a way that every triangle of K_n is not monocolored.

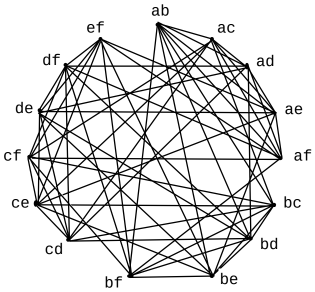


Fig. 2. The UE graph G_6.

Since the Ramsey number $r(3,3) = 6$ we have that G_6 is not 2-clique-colorable. Now, let $r(k_1, k_2, \ldots, k_m)$ be the smallest integer n such that every edge coloring $(E_1, E_2, \ldots, E_m)$ of K_n contains, for some i, a complete subgraph on k_i vertices, all of whose edges are in color i. These are natural generalizations of the Ramsey numbers for which an upper bound is known [2]. When we take $n = r(3, 3, \ldots, 3)$ for some m, we have that G_n has no m-clique-coloring.

As a consequence we have:

Theorem 3.1 *There is no constant C such that all UE graphs are C-clique-colorable.*

A natural question that remains is to determine the complexity, for a given k, of the k-clique-coloring problem for UE graphs.

References

[1] Bacsó, G., S. Gravier, A. Gyárfás, M. Preissmann, and A. Sebö, *Coloring the maximal cliques of graphs*, SIAM Journal on Discrete Mathematics **17(3)** (2004), 361–376.

[2] Bondy, J.A., and U.S.R. Murty, "Graph Theory with Applications," Macmillan, London, 1976.

[3] Golumbic, M.C., and R.E. Jamison, *The edge intersection graphs of paths in a tree*, Journal of Combinatorial Theory (Series B) **38** (1985), 8–22.

[4] Kratochvíl, J., and Zs. Tuza, *On the complexity of bicoloring clique hypergraphs of graphs*, Journal of Algorithms **45** (2002), 40–54.

[5] Monma, C.L., and V.K. Wei, *Intersection graphs of paths in a tree*, Journal of Combinatorial Theory (Series B) **41** (1986), 141–181.

On a Conjecture of Víctor Neumann-Lara

Bernardo Llano [1]

Departamento de Matemáticas, Universidad Autónoma Metropolitana, Iztapalapa, México, D.F.

Mika Olsen [2]

Departamento de Matemáticas Aplicadas y Sistemas, Universidad Autónoma Metropolitana, Cuajimalpa, México, D.F.

Abstract

We disprove the following conjecture due to Víctor Neumann-Lara: for every couple of integers (r, s) such that $r \geq s \geq 2$ there is an infinite set of circulant tournaments T such that the dichromatic number and the acyclic disconnection of T are equal to r and s respectively. We show that for every integer $s \geq 2$ there exists a sharp lower bound $b(s)$ for the dichromatic number r such that for every $r < b(s)$ there is no circulant tournament T satisfying the conjecture with these parameters. We give an upper bound $B(s)$ for the dichromatic number r such that for every $r \geq B(s)$ there exists an infinite set of circulant tournaments for which the conjecture is valid.

Keywords: Circulant tournament, acyclic disconnection, dichromatic number.

[1] llano@xanum.uam.mx

[2] olsen@correo.cua.uam.mx

1 Introduction

The dichromatic number and the acyclic (resp. cyclic triangle free) disconnection were introduced as measures of the complexity of the cyclic structure of digraphs. A large value of the dichromatic number and, oppositely, a small value of the acyclic disconnection express a more complex cyclic structure of a given digraph. Among other papers, see [1,3,4,5,6] for old and recent results on the study of these parameters as well as open problems.

We define the *dichromatic number* of a digraph D, denoted by $dc(D)$, as the minimum number of colors with which one can color the vertices of D such that each chromatic class induces an acyclic subdigraph of D (that is, a subdigraph containing no directed cycles). On the other hand, the *cyclic triangle free* (briefly, the $\overrightarrow{C}_3$*-free*) *disconnection* of a digraph D, denoted by $\overrightarrow{\omega}_3(D)$, is defined to be the maximum number of colors with which one can color the vertices of D not producing heterochromatic cyclic triangles.

In 1999, V. Neumann-Lara posed the following

Conjecture 1.1 ([5], Conjecture 5.8) *For every couple of integers (r, s) such that $r \geq s \geq 2$ there is an infinite set of circulant tournaments T such that $dc(T) = r$ and $\overrightarrow{\omega}_3(T) = s$.*

In [3], the authors positively answer the conjecture for the special case when $r = 3$ and $s = 2$. However, in this paper the conjecture is disproved in general. We show that for every integer $s \geq 2$ there exists a lower bound $b(s)$ for the dichromatic number r such that for every $r < b(s)$ there is no a circulant tournament T satisfying the conjecture with these parameters. We construct an infinite set of circulant tournaments T such that $dc(T) = b(s)$ and $\overrightarrow{\omega}_3(T) = s$ and give an upper bound $B(s)$ for the dichromatic number r such that for every $r \geq B(s)$ there exists an infinite set of circulant tournaments for which the conjecture is valid. Some infinite sets of circulant tournaments confirming the conjecture are given for $b(s) < r < B(s)$. The construction of the remaining cases in this interval is an open problem since the tools used in the paper do not apply for them.

For the terminology on digraphs and tournaments, see [2].

2 Preliminaries

Let $D = (V, A)$ be a digraph. For any $v \in V(D)$ we denote by $N^+(v)$ and $N^-(v)$ the out- and in-neighborhood of v in D respectively. A digraph D is said to be *acyclic* if D contains no directed cycles. A subset $S \subseteq V(D)$ is *acyclic* if the induced subdigraph $D[S]$ of D by the set S is acyclic. The maximal cardinality of an acyclic set of vertices of D is denoted by $\beta(D)$.

An r-coloring $\varphi : V(D) \to \{1, 2, ..., r\}$ of a digraph D is a surjective function defined on the vertices. A subdigraph D' of D is *heterochromatic* if every pair of vertices of D' receive different colors from φ. A subset S of vertices of D that receive the same color from φ is called a *chromatic class* and it is *singular* if $|S| = 1$. We say that a proper r-coloring φ of a digraph D is $\overrightarrow{C}_3$*-free* if it does not produce heterochromatic cyclic triangles.

Let D and F be digraphs and F_v a family of mutually disjoint isomorphic copies of F for all $v \in V(D)$. The *composition* (or *lexicographic product*) $D[F]$ of the digraphs D and F is defined by $V(D[F]) = \bigcup_{v \in V(D)} V(F_v)$ and $A(D[F]) = \bigcup_{v \in V(D)} A(F_v) \cup \{(i, j) : i \in V(F_v), j \in V(F_w) \text{ and } (v, w) \in A(D)\}$. It is easy to prove that the composition of digraphs is an associative but not a commutative operation.

A tournament T is said to be *tight* if $\overrightarrow{\omega}_3(T) = 2$. We define $\mathcal{F}_{r,s} = \{T : dc(T) = r, \overrightarrow{\omega}_3(T) = s\}$.

Let $\mathbb{Z}_{2m+1}$ be the cyclic group of integers modulo $2m + 1$ $(m \geq 1)$ and J a nonempty subset of $\mathbb{Z}_{2m+1} \setminus \{0\}$ such that $|\{-i, i\} \cap J| = 1$ (and therefore $|J| = m$). A *circulant* (or *rotational) tournament* $\overrightarrow{C}_{2m+1}(J)$ is defined by $V(\overrightarrow{C}_{2m+1}(J)) = \mathbb{Z}_{2m+1}$ and $A(\overrightarrow{C}_{2m+1}(J)) = \{(i, j) : i, j \in \mathbb{Z}_{2m+1} \text{ and } j - i \in J\}$. Let $I_m = \{1, 2, ..., m\}$. We denote by $\overrightarrow{C}_{2m+1}\langle\varnothing\rangle$ and $\overrightarrow{C}_{2n+1}\langle j\rangle$ the circulant tournaments $\overrightarrow{C}_{2m+1}(J)$ where $J = I_m$ and $J = (I_m \diagdown \{j\}) \cup \{-j\}$ respectively.

We use the following definition taken from *[5]*.

A digraph D is said to be $\overrightarrow{\omega}_3$*-keen* if there exists a proper $\overrightarrow{C}_3$-free r-coloring φ of D such that φ is optimal (that is, it uses the maximum number of colors), there exists a unique singular chromatic class and no proper $\overrightarrow{C}_3$-free r-coloring of D leaves more than one such a class.

3 An infinite family of tight r-dichromatic circulant tournaments for all $r \geq 2$

Let $\overrightarrow{C}_{n(p+1)+1}(J)$ be the family of circulant tournaments defined in [1], where $p = (n-1)(t+1)+1$, $n \geq 2$, $t \geq 0$ and $J = \{1, 2, \ldots, p\} \cup \{p+2, p+3, \ldots, 2p-t\} \cup \{2p+3, \ldots, 3p-2t\} \cup \ldots \cup \{(n-1)p+n\}$.

Lemma 3.1 ([1], Theorem 6) $dc\left(\overrightarrow{C}_{n(p+1)+1}(\triangledown_{p,t})\right) = n + 1.$

Theorem 3.2 $\overrightarrow{C}_{n(p+1)+1}(\triangledown_{p,t})$ *is tight.*

Corollary 3.3 *For all $r \geq 2$ there exists an infinite family of tight r-dichromatic circulant tournaments.*

4 Infinite families for $s \geq 3$

Let $H = (V, E)$ a finite hypergraph. A hypergraph H is t-*uniform* (or simply a t-*graph*) if all of its edges have cardinality t. A hypergraph H is called *circulant* if it has an automorphism which is a cyclic permutation of $V(H)$. If $t \leq m$, the circulant t-graph $\Lambda_{m,t}$ is defined by $V(\Lambda_{m,t}) = \mathbb{Z}_m$ and $E(\Lambda_{m,t}) = \{\alpha_j : j \in \mathbb{Z}_m\}$ where $\alpha_j = \{j, j+1, \ldots, j+t-1\}$ for $j \in \mathbb{Z}_m$.

Proposition 4.1 ([6], Propositions 3.2(iii) and 3.4, Corollary 4.3) *Let T, U be circulant tournaments, T of order $2m+1$, and U a r-dichromatic tournament. Then $dc(T[U]) \geq \left\lceil \frac{r(2m+1)}{m+1} \right\rceil$. Moreover, if $\beta(T) = t$ and T contains a isomorphic copy of $\Lambda_{2m+1,t}$ as a spanning subhypergraph, then $dc(T[U]) = \left\lceil \frac{r(2m+1)}{t} \right\rceil$.*

Proposition 4.2 ([5], Proposition 3.6(i)) *Let T be an ω_3-keen tournament and let U be a tournament. Then $\overrightarrow{\omega}_3(T[U]) = \overrightarrow{\omega}_3(T) + \overrightarrow{\omega}_3(U) - 1$.*

Proposition 4.3 ([7], Corollary 1) *Consider the recurrence relation $D_n = \left\lceil \frac{3}{2} D_{n-1} \right\rceil$ $(n \geq 1,\ D_0 = 1)$, then $D_n = \left\lfloor K \left(\frac{3}{2}\right)^n \right\rfloor \qquad (n = 1, 2, \ldots)$, where $K = 1.62227\,05028\,84767\,31595\,69509\,82899\,32411\ \ldots$*

Let $s \in \mathbb{N}$ and define $\overrightarrow{C}_3^{s} = \overrightarrow{C}_3^{s-1}\left[\overrightarrow{C}_3\right]$, where $\overrightarrow{C}_3^{1} = \overrightarrow{C}_3$ by convention. As a consequence of Proposition 4.2, we have the following

Corollary 4.4 *Let T be a circulant tournament such that $\overrightarrow{\omega}_3(T) = s$, then there exist $s-1$ tight circulant tournaments $T_1, T_2, \ldots, T_{s-1}$ such that $T \cong T_1[T_2[\ldots[T_{s-1}]]]$.*

Theorem 4.5 *Let T be a circulant tournament such that $dc(T) = r$ and $\overrightarrow{\omega}_3(T) = s$ $(2 \leq s \leq r)$. Let $b(s) = D_{s-1}$. Then $r \geq b(s) = \left\lceil \frac{3}{2} D_{s-2} \right\rceil = \left\lfloor K \left(\frac{3}{2}\right)^{s-1} \right\rfloor$ Moreover, $\left\{ \overrightarrow{C}_3^{s-2}[\alpha], \alpha \in \mathcal{F}_{2,2} \right\}$ is an infinite set of r-dichromatic circulant tournaments such that $r = b(s)$ and $\overrightarrow{\omega}_3\left(\overrightarrow{C}_3^{s-2}[\alpha]\right) = s$.*

Note that $b(s)$ is a lower bound for $dc(T) = r$ when $\overrightarrow{\omega}_3(T) = s$ and there are no r-dichromatic circulant tournaments for which $r < b(s)$.

For example, if $\overrightarrow{\omega}_3(T) = 4$, then $dc(T) \geq \left\lfloor K \left(\frac{3}{2}\right)^3 \right\rfloor = 5 = b(4)$.

Let D be a digraph. The hypergraph $H_1(D)$ is defined by $V(H_1(D)) = V(D)$ and $E(H_1(D)) = \{U \subseteq V(D) : U \text{ is a maximal acyclic set}\}$.

Lemma 4.6 *Let $m \geq 2$. Then $H_1\left(\overrightarrow{C}_{2m+1}\langle m-1\rangle\right) \supseteq \Lambda_{2m+1,m-1}$ and $\beta(\overrightarrow{C}_{2m+1}\langle m-1\rangle) = m-1$.*

As a consequence of Proposition 4.1 and Lemma 4.6, we have

Corollary 4.7 *Let α be a r-dichromatic tournament, then $dc\left(\overrightarrow{C}_{2m+1}\langle m-1\rangle[\alpha]\right) = \left\lceil \frac{r(2m+1)}{m-1} \right\rceil$. Moreover, if $m \geq 3r+1$, then $dc\left(\overrightarrow{C}_{2m+1}\langle m-1\rangle[\alpha]\right) = 2r+1$.*

Let f_i, $f_i' : \mathbb{N}^2 \longrightarrow \mathbb{N}$ $(i = 0, 1, 2)$ be functions defined by

$$f_0'(q,m) = q(m+1) - 1,\ \ f_0(q,m) = q(2m+1) - 1\ \ (q \geq 1, m \geq 2),$$

$$f_1'(q,m) = qm + 1,\ \ f_1(q,m) = q(2m+1) + 3\ \ (q \geq 1, m \geq 3),$$

$$f_2'(q,m) = 2q + m,\ \ f_2(q,m) = 3q + m + 1\ (q \geq 1, 1 \leq m \leq 2.$$

We define the (infinite) digraphs D_i for $i = 0, 1, 2$ as follows:$V(D_i) = \{v \in \mathbb{Z} : v \geq 3\}$ and $A(D_i) = \{(f_i'(q,m), f_i(q,m))\}$. Let $D = D_0 \cup D_1 \cup D_2$. Clearly, D_0, D_1 and D_2 are arc-disjoint and acyclic. We point out that f_i, f_i' and D_i $(i = 0, 1)$ were defined in [6], where more details can be found.

Lemma 4.8 *For each positive integer $n \geq 3$, there is a directed path in D from a vertex in $S = \{3, 4, 5, 7\}$ to n.*

Remark 4.9 The set of vertices of D with indegree 0 is $\{3, 4, 5, 7\}$.

Proposition 4.10 *(i) $\mathcal{F}_{2r,s+1} = \left\{\overrightarrow{C}_{2r+1}\langle\emptyset\rangle[\alpha] : \alpha \in \mathcal{F}_{r,s}\right\}$.*

(ii) $\mathcal{F}_{2r+1,s+1} = \left\{\overrightarrow{C}_{2(3r+1)+1}\langle 3r\rangle[\alpha] : \alpha \in \mathcal{F}_{r,s}\right\}$.

(iii) $\mathcal{F}_{3r,s+1} = \left\{\overrightarrow{C}_3[\alpha] : \alpha \in \mathcal{F}_{2r,s}\right\}$.

(iv) $\mathcal{F}_{3r+2,s+1} = \left\{\overrightarrow{C}_3[\alpha] : \alpha \in \mathcal{F}_{2r+1,s}\right\}$.

Let $B(s)$ $(s \geq 2)$ be a positive integer such that for all $r \geq B(s)$ there exists an infinite set of circulant tournaments $T \in \mathcal{F}_{r,s}$. Clearly, $B(2) = 2$ (see Corollary 3.3).

Proposition 4.11 *$B(s) \leq 2B(s-1) - 1 \leq 2^{s-1}$ for all $s \geq 3$.*

Theorem 4.12 *For the following couples of integers (r, s) there exists an infinite set $\mathcal{F}_{r,s}$ of circulant tournaments.*

(i) $3 \leq s \leq 5$ and all $r \geq b(s)$.

(ii) $s = 6$ *and all* $r \geq 12$, $r \neq 13$.
(iii) $s = 7$ *and* $r = 18, 21$ *and all* $r \geq 23$.
(iv) $s = 8$ *and* $r = 27, 32$ *and all* $r \geq 35$.
(v) $s = 9$ *and* $r = 41, 48, 51$ *and all* $r \geq 53$.
(vi) $s = 10$ *and* $r = 62, 72, 76, 77$ *and all* $r \geq 80$.
(vii) $s = 11$ *and* $r = 93, 121$ *and all* $r \geq 123$.

We conjecture that $B(s) \leq 2b(s-1) - 1$ for all $s \geq 3$.

References

[1] G. Araujo, M. Olsen, *A conjecture of Neumann-Lara on infinite families of r-dichromatic circulant tournaments* (submitted).

[2] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications* (American Elsevier Pub. Co., 1976).

[3] B. Llano, M. Olsen, *Infinite families of tight regular tournaments* (to appear in Matematicae Discussiones).

[4] V. Neumann-Lara, *The dichromatic number of a digraph,* J. Combin. Theory (B) **33** (1982) 265-270.

[5] V. Neumann-Lara, *The acyclic disconnection of a digraph*, Discr Math. **197/198** (1999) 617-632.

[6] V. Neumann-Lara, *Dichromatic number, circulant tournaments and Zykov sums of digraphs,* Discuss. Math. Graph Theory **20** (2000) 197-207.

[7] A. Odlyzko, H. Wilf, *Functional iteration and the Josephus problem*, Glasgow Math. J. **33** (1991) 235-240.

Online Bounded Coloring of Permutation and Overlap Graphs

Marc Demange [2]

Département SID, ESSEC
Cergy Pontoise, France

Gabriele Di Stefano [1,3]

Dipartimento DIEI, Università dell'Aquila
L'Aquila, Italy

Benjamin Leroy-Beaulieu [4]

ROSE, École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland

Abstract

We consider a track assignment problem in a train depot leading to an online bounded coloring problem on permutation graphs or on overlap graphs. For permutation graphs we study the competitiveness of a First Fit-based algorithm and we show it matches the competitive ratio of the problem. For overlap graphs, even the unbounded case does not admit a constant competitive ratio.

Keywords: Online algorithms, Competitive analysis, Bounded coloring, Permutation graphs, Overlap graphs, First-Fit

1 Introduction

This work is motivated by the track assignment problem in a train depot. Some aspects of this problem have been studied before [3,4,5,8].

In a train depot, trains must be stored on tracks. Furthermore, the number of available tracks is known and fixed to k. The tracks are organized as stacks, such that the last train to enter must be the first to leave. In order to save upkeep costs, one wants to use as few tracks as possible. Meanwhile, in order to save time and energy, one wants to make sure that, when a train departs, it is always at the top of the stack. Since the trains have accumulated lateness before arriving at the depot, the time of arrival of each train is unpredictable. The tracks must thus be assigned online, as the trains arrive, on the basis of departure times and previous assignments.

Our main contribution to this problem resides in the introduction of a finite length to the tracks: each one of them may contain at most b trains, where b is a fixed integer.

One can represent on a time-axis the time-intervals during which each train must be stored in the depot. Thus, each interval corresponds to one train. Two trains may be on the same track if their intervals are completely disjoint or if the interval of one train is contained in the interval of the other train (the first train arrives before and leaves after the second). One can make a graph where each vertex represents an interval and two vertices are joined by an edge if and only if the two intervals overlap but have no containment relationship. Such a graph is called an *overlap graph* [7]. To model our track assignment problem, we can thus use online coloring of overlap graphs with the constraint that each color may not contain more than b elements. However, since one gets the knowledge of a train at the time when it arrives, the online model must state that the vertices of the overlap graph are presented in the order of the left extremity of their corresponding intervals along the time axis.

It is natural to consider the same problem for night depots: we can assume that all trains are in the depot before the first scheduled departure in the morning. This is called the *midnight condition* and it means that the intervals of the trains all share at least one point. It has been proved [6] that in this particular case, the overlap graph is a *permutation graph*. It is thus natural

[1] This work was partially supported by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL).

[2] Email: demange@essec.fr

[3] Email: gabriele@ing.univaq.it

[4] Email: benjamin.leroy-beaulieu@epfl.ch

to study the online bounded coloring of permutation graphs.

2 Definitions and notations

Any term related to graph theory not defined here can be found in [1].

We consider online versions of the graph coloring problem. An online problem can be seen as a two players game involving an adversary and an algorithm. The adversary presents the instance and the algorithm gives the solution. The online problem is characterized by the underlying offline problem and two sets of rules that have to be respected by the adversary and the algorithm, respectively.

Online algorithms are evaluated according to their *competitive ratio*. Let A be an online algorithm and P a minimization problem. Then $p_A(I)$ denotes the maximum score A achieves for P over the online presentations of the instance I respecting the given rules. An online algorithm is said to guarantee a competitive ratio of c (or to be c-competitive) if, for every instance I, $p_A(I) \leqslant c \cdot p(I)$, where $p(I)$ is the offline optimal solution for P on I. It is called *exact* if it guarantees an optimal offline solution. It is called *optimal* if its competitive ratio can not be improved by any other online algorithm. The *competitive ratio of a problem* is the competitive ratio of an optimal algorithm for the problem.

Overlap graphs and *permutation graphs* are defined in [7]. In this paper, we will use a very common representation for permutation graphs, called the *lattice representation*. The permutation is represented by points on a two-dimensional lattice. A point (x, y) on the lattice means that the element y is at position x in the permutation. Given a set of points in the lattice as above, the corresponding permutation graph is such that the points are the vertices and two points (x, y) and (x', y') are adjacent if $x < x'$ and $y > y'$. The (bounded) coloring of permutation graph is equivalent to partitioning a permutation into (bounded) increasing subsequences.

If a graph is colorable with k colors such that each color contains at most b elements, we call this graph (k, b)*-colorable*.

A very common algorithm for coloring graphs is the greedy algorithm *First-Fit*, denoted by FF. It considers the vertices one after the other following a given order. It assigns to vertex v the first color that was not assigned to an adjacent vertex of v. Chvátal [2] characterized the *perfect orderings* that are such that FF solves the graph coloring problem on the graph and all its induced subgraphs optimally. For permutation graphs, it is the case when the vertices are presented following the order in the permutation from the left to the

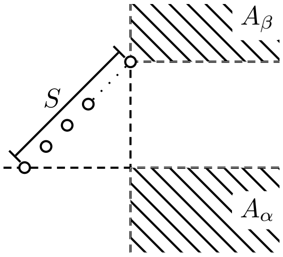


Fig. 1. Zone A after presenting one increasing sequence S. The zone denoted A_α represents A if S is of type α and the zone denoted A_β represents A if S is of type β.

right. Note however that for arbitrary vertex-orderings the FF algorithm can behave arbitrarily bad even on permutation graphs [8]. Moreover, it is known that, in the online case, receiving a permutation gives more informations that receiving the corresponding permutation graph [4]. We consider that the graph is presented in the form of the corresponding permutation, which is natural in the track assignment problem. In this paper we also consider the b-bounded First-Fit algorithm (b-FF): it colors a new vertex with the smallest possible color used less than b times.

3 Bounds

We start by studying the particular case respecting the midnight condition.

Theorem 3.1 *The performance ratio c of the online coloring of a (k, b)-colorable permutation is $c = 2 - 1/l$, where $l = \min\{b, k\}$, and b-FF achieves this ratio.*

Proof. We prove Theorem 3.1 with two claims. □

Claim 3.2 *A lower bound for the online bounded coloring of a permutation is $2 - 1/l$, where $l = \min\{b, k\}$.*

Proof. Suppose that k is known and fixed a priori. We prove Claim 3.2 with a permutation presented on its lattice representation. The zone A denotes the admissible zone where any element in the future will be presented. At the beginning, A is the complete plan.

(i) Present a stable set S (an increasing sequence) of size b in A.

(ii) If the elements of S are colored with two or more different colors, define A as the part of the plan which is below and to the right of S, such that every element on A will be in a clique with every element in S. S is said to be of type α. If all elements of S are colored with the same color, define A as the part of the plan which is above and to the right of S, such that every element on A will be in a stable set with every element in S. S is said to be of type β. See Fig. 1.

(iii) Repeat steps i and ii until $l-1$ sequences S have been presented.

(iv) Let N_β be the number of sequences S of type β. Present a clique C_γ of size $N_\beta + 1$ in A.

□

Remark 3.3 If k is unknown in advance, we can consider the same instance as above with $k \geqslant b$ and thus have $c \geqslant 2 - 1/b$.

Claim 3.4 *The performance ratio of FF for the online bounded coloring of a (k, b)-colorable permutation delivered from left to right is at most $2 - 1/l$, where $l = \min\{b, k\}$.*

Proof. Let $\lambda_{b\text{-FF}}(\pi)$ be the number of colors used by FF for the online bounded-coloring of the permutation π presented from left to right. b is the bound imposed by the problem.

Let N_S be the number of colors which are saturated at the end of the execution, i.e., the colors which contain exactly b vertices. Let $N_{\bar{S}}$ be the number of colors which are not saturated. □

Remark 3.5 If one considers only the permutation π' induced by elements colored by colors in $N_{\bar{S}}$, we have $\lambda_{b\text{-FF}}(\pi') = \lambda_{\text{FF}}(\pi') = \chi(\pi')$ where $\lambda_{\text{FF}}(\cdot)$ denotes the number of colors used by the FF algorithm for usual coloring and where the last equality holds since the permutation is presented from left to right (a perfect order).

Let χ_b be the exact bounded-chromatic number of the permutation π for the bound b.

$$\lambda_{b\text{-FF}}(\pi) = N_S + N_{\bar{S}} \tag{1}$$

Since colors in $N_{\bar{S}}$ are non-empty, we have:

$$n \geqslant bN_S + N_{\bar{S}} \tag{2}$$

$$\chi_b \geqslant \left\lceil \frac{n}{b} \right\rceil \geqslant \left\lceil \frac{bN_S + N_{\bar{S}}}{b} \right\rceil = N_S + \left\lceil \frac{N_{\bar{S}}}{b} \right\rceil \tag{3}$$

$$N_S \leqslant \chi_b - \left\lceil \frac{N_{\bar{S}}}{b} \right\rceil \tag{4}$$

We deduce from remark 3.5 that $N_{\bar{S}} \leqslant \chi_b$. Thus,

$$\lambda_{b\text{-FF}}(\pi) \leqslant 2\chi_b - \left\lceil \frac{\chi_b}{b} \right\rceil \tag{5}$$

Calculations, skipped here for the sake of space, lead to the conclusion that

$$c_b\text{-FF} \leqslant \frac{\lambda_b\text{-FF}(\pi)}{\chi_b} \leqslant 2 - \frac{1}{b} \leqslant 2 - \frac{1}{l} \tag{6}$$

This ends the proof of Claim 3.4. Together, Claims 3.2 and 3.4 end the proof of Theorem 3.1. □

In the case of unbounded coloring, FF is exact on a permutation if it is delivered from left to right [7]. For overlap graphs, on the contrary, no constant competitive ratio can be guaranteed:

Theorem 3.6 *For any online coloring algorithm, it is possible to force any positive number of colors on a bipartite overlap graph presented online from left to right, according the the left-most point of each interval.*

Proof. For the sake of space, the proof is omitted. □

References

[1] Berge, C., "Graphs and Hypergraphs," North Holland, Amsterdam, 1976.

[2] Chvátal, V., *Perfectly ordered graphs*, in: *Topics on Perfect Graphs*, Annals of Discrete Mathematics **21** (1984), pp. 63–65.

[3] Cornelsen, S. and G. Di Stefano, *Track assignment*, Journal of Discrete Algorithms **5** (2007), pp. 250–261.

[4] Demange, M. and B. Leroy-Beaulieu, *Some results on the online coloring of comparability graphs* (2006), international Symposium on Mathematical Programming (ISMP), Rio de Janeiro.

[5] Di Stefano, G. and M. L. Koci, *A graph theoretical approach to the shunting problem*, in: B. Gerards, editor, *Proceedings of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003)*, Electronic Notes in Theoretical Computer Science **92**, 2004.

[6] Gavril, F., *Algorithms for a maximum clique and a maximum independent set of a circle graph*, Networks **3** (1973), pp. 261–273.

[7] Golumbic, M. C., "Algorithmic Graph Theory and Perfect Graphs," Computer Science and Applied Mathematics, Academic Press, 1980.

[8] Nikolopoulos, S. D. and C. Papadopoulos, *On the performance of the first-fit coloring algorithm on permutation graphs*, Information Processing Letters **75** (2000), pp. 265–273.

A robust P2P streaming architecture and its application to a high quality live-video service

Héctor Cancela [a], Franco Robledo Amoza [a],
Pablo Rodríguez-Bocca [a,b,1], Gerardo Rubino [b] and
Ariel Sabiguero [a,b]

[a] *Instituto de Computación, Universidad de la República, Montevideo, Uruguay.*
[b] *IRISA - INRIA, Rennes, France.*

Abstract

This paper explores the design of a P2P architecture that sends real-time video over the Internet. The aim is to provide good quality levels in a highly dynamic P2P topology, where the frequent connections/disconnections of the nodes makes it difficult to offer the Quality-of-Experience (QoE) needed by the client.

We study a multi-source streaming approach where the stream is decomposed into several flows sent by different peers to each client including some level of redundancy, in order to cope with the fluctuations in network connectivity. We employ the recently proposed PSQA technology for evaluating automatically the perceived quality at the client side. We introduce a mathematical programming model to maximize the global expected QoE of the network (evaluated using PSQA), selecting a P2P connection scheme which enhances topology robustness. In addition, we provide an approximated algorithm to solve the proposed model, and we apply it to solve a case study based on real life data.

[1] Email: prbocca@fing.edu.uy

1 P2P Robust Assignment Model

This work studies the characteristics of a P2P based solution for live video broadcasting. In particular, in this section we develop a mathematical programming model for the stream assignment problem in order to maximize the expected perceived quality taking into account the network dynamics.

Time. The system is reconfigurated at discrete points in time, every Δt. Let us use Δt as unit of time, and denote by I_t the tth interval $(t, t+1]$.

Distribution scheme. The stream is originally sent to the clients (or terminals) by a brodcaster node s. Some clients act also as servers, relaying streams to other clients. For this purpose, each node v has an available output bandwidth BW_v^{out}. The system distributes a single stream of live video by means of K sub-streams denoted by $\sigma_1, \sigma_2, ..., \sigma_K$. Each substream σ_k is sent with a constant bandwidth bw_k. The total bandwidth of the stream is $\sum_{k=1}^{K} bw_k$. When a client receives the K substreams, it reproduces perfectly the stream. If it does not receive all the K substreams, it will reproduce a stream that can have a lower quality, depending on which substreams it receives and which redundant scheme is used (not discussed here).

Dynamics. The evolution of the system from t to $t+1$ is as follows: some nodes leave in I_t, possibly disconnecting other clients in some substreams; at the same time, some nodes enter the network requesting for connection; they remain isolated from the rest of the nodes until $t+1$ when the new reconfiguration action is performed. The goal of the reconfiguration is to reconnect the disconnected nodes and to connect the new arrivals to the network. The connexion scheme always builds trees of peers. At time $(t+1)^-$, just before the reconfiguration, the general situation is the following. For each substream σ_k there is a *principal tree*, $\mathcal{P}_k$, containing (at least) the source s; all its nodes receive substream σ_k. There are also $M_k \geq 0$ other trees, disjoint between them and with $\mathcal{P}_k$, denoted by $\tau_{k,1}, \tau_{k,2}, \cdots, \tau_{k,M_k}$; their nodes do not receive σ_k. The set of trees associated with substream σ_k is called *plane k*. A *perfect network* is a system where for each substream σ_k there is only one directed tree ($\mathcal{P}_k$, the principal one), meaning that $M_k = 0$. One of the goals of the reconfiguration action is to build a perfect network.

Quality. To evaluate the quality at the client side, we use the PSQA technology. Pseudo-Subjective Quality Assessment (PSQA) [3] is a general procedure that automatically measures the perceived quality, accurately and in real time. For instance, if we assume that the PSQA metric is scaled on [0..1] and if the network is perfect, its instantaneous total quality is equal to the number of connected clients (because $PSQA = 1$ for each client).

Optimization. As said before, the reconfiguration action will try to build a perfect network, and, among the huge number of possible perfect newtorks, it will try to build a robust one. For this purpose, we keep statistics about the nodes' behavior allowing us to evaluate their expected departure times from the network. Specifically, we maintain an estimate p_i of the probability that node i remains connected in the current period, when it is connected at the beginning (see below).

Formal Model. We propose now an Integer Mathematical Programming Model which contemplates the P2P dynamics (described above) in a time interval $(t, t+1]$. Consider the system at time t^+ and let $\mathcal{N}(t)$ be the set of connected nodes at t^+ ($N = \|\mathcal{N}(t)\|$). Define for each $k \in \{1, 2, \cdots, K\}$ and $i, j \in \mathcal{N}(t)$,

$$x_{i,j}^k = \begin{cases} 1 & \text{if node } i \text{ sends } \sigma_k \text{ to node } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{i,j}^k = \begin{cases} 1 & \text{if node } i \text{ precedes node } j \text{ in the tree containing } j \text{ in plane } k, \\ 0 & \text{otherwise.} \end{cases}$$

Since the perceived quality at node i depends on which substream is received by i, we assume available a function $f()$ of K variables such that the quality at time t experienced by node $i \in \mathcal{N}(t)$ (and measured using PSQA) is $Q_i = f(y_{s,i}^1, y_{s,i}^2, .., y_{s,i}^K)$.

For all $i \in \mathcal{N}(t)$, let z_i be the binary random variable equal to 1 if node i remains connected until $t+1$, 0 otherwise, where $\Pr(z_i = 1) = p_i$. The sets $\left\{x_{ij}^k\right\}$ and $\left\{y_{ij}^k\right\}$ specify the network configuration at t whereas the network configuration at $t+1$ is determined by the variables $\left\{\tilde{x}_{ij}^k\right\}$ and $\left\{\tilde{y}_{ij}^k\right\}$ satisfying the following relations: $\tilde{x}_{i,j}^k = z_i x_{i,j}^k z_j$, $\tilde{y}_{i,j}^k = \tilde{x}_{i,j}^k + \sum_{l=1}^N z_i \tilde{y}_{i,l}^k \tilde{x}_{l,j}^k$, and $\tilde{y}_{i,j}^k \le y_{i,j}^k$.

The PSQA evaluation of the quality as perceived by node i at time $t+1$ is a random variable which is a function of $\left\{\tilde{y}_{s,i}^k\right\}$ and r.v. $\{z_i\}$. We will maximize its expected value, $\mathrm{E}\{\tilde{Q}_i\}$, where $\tilde{Q}_i = f(\tilde{y}_{s,i}^1, \tilde{y}_{s,i}^2, .., \tilde{y}_{s,i}^K)$ (actually, we use a scaled expression, see Fig. 1, which shows the complete model).

2 Algorithmic solution based on GRASP

GRASP [4] is a well known metaheuristic that has been successfully used to solve many hard combinatorial optimization problems. It is an iterative process which operates in two phases. In the *Construction Phase* an initial feasible solution is built whose neighborhood is then explored in the *Local Search Phase*. Next, we present a GRASP customized to solve our problem.

$$\max \mathsf{E}\left\{\frac{\sum_{i=1}^{N}\tilde{Q}_i}{\sum_{i=1}^{N} z_i}\right\} \quad \text{// global expected perceived quality}$$

st:

$$y_{i,j}^k + y_{j,i}^k \leq 1, \forall i,j \in N, \forall k \in K, \quad \text{// loops are not allowed}$$

$$y_{i,j}^k = x_{i,j}^k + \sum_{l=1}^{N} y_{i,l}^k x_{l,j}^k, \forall i \in N, \forall j \in N - \{s\}, \forall k \in K, \quad \text{// precedence constraints}$$

$$\sum_{i=1}^{N} x_{i,j}^k \leq 1, \forall j \in N, \forall k \in K, \quad \text{// each substream arrives from an only node}$$

$$\sum_{k=1}^{K} \{bw^k \sum_{j=1}^{N} x_{i,j}^k\} \leq BW^{out}(i), \forall i \in N, \quad \text{// bandwidth capacity constraints}$$

$$x_{i,j}^k = 1, \forall i \in N, \forall j \in N, \forall k \in K | x_{i,j}^k \in E_t^k, \quad \text{// network configuration at } t$$

$$\tilde{x}_{i,j}^k = z_i x_{i,j}^k z_j, \forall i \in N, \forall j \in N, \forall k \in K, \quad \text{// a link is preserved if source and terminal do not leave}$$

$$\tilde{y}_{i,j}^k = \tilde{x}_{i,j}^k + \sum_{l=1}^{N} z_i \tilde{y}_{i,l}^k \tilde{x}_{l,j}^k, \forall i \in N, \forall j \in N - \{s\}, \forall k \in K, \quad \text{// } \tilde{y} \text{ represents the precedence at } t+1$$

$$\tilde{y}_{i,j}^k \leq y_{i,j}^k, \forall i \in N, \forall j \in N, \forall k \in K,$$

$$z_i \sim Bern(p_i), \forall i \in N, \quad \text{// random variables with Bernoulli distribution (parameters } p_i\text{)}$$

$$x_{i,j}^k, y_{i,j}^k \in \{0,1\}, \forall i \in N, \forall j \in N, \forall k \in K$$

Fig. 1. Mathematical Programming Model.

Construction phase. Let's assume that the initial graph is not connected, and that the distribution trees were pruned in some arcs. We will identify disconnected subtrees $\tau_{k,i}$ by their roots for each substream k and nodes i with available bandwidth. The tree that contains the stream source will be called the main tree $\mathcal{P}_k$, for each substream k. Let *p_a_list* be the set of all possible assignments that can be performed in that state, which reconnect the graph. Each component of the assignment is a triplet (source node, target node, substream), and belongs to the set if and only if: (i) the source node belongs to the main tree, (ii) the target node is root of some disconnected tree, (iii) the source node has enough bandwidth to transmit the substream. Fig. 2 describes the construction of the initial solution. The procedure constructively builds the initial solution starting from the disconnected graph. A set, RCL, of the eligible assignments which result in the larger improvements to the current partial solution is constructed. The parameter n determines the size of the set RCL. At each iteration one of the elements from RCL is added to the current solution, this decision is made determining randomly the element *current_a* from the set RCL. The set *p_a_list* has to be updated due to the assignation of *current_a*. The *p_a_list* set may become empty if no disconnected trees

```
Procedure GRASP_P2P_Network;
Input: $G$, $k$;

$a\_list \leftarrow$ empty_list;
$current\_psqa \leftarrow 0$; $best\_psqa \leftarrow -1$;
for $i = 1$ to MAXITER do
    $[\mathcal{G}_{sol}, a\_list] \leftarrow$ P2P_Const_Phase($G, a\_list$);
    $[\mathcal{G}_{sol}, a\_list] \leftarrow$ P2P_Local_Search($\mathcal{G}_{sol}, k$);
    $current\_psqa \leftarrow$ PSQA_Expected($\mathcal{G}_{sol}$) ;
    if ($current\_psqa > best\_psqa$) then
        $best\_psqa \leftarrow current\_psqa$;
        $\mathcal{G}_{best} \leftarrow \mathcal{G}_{sol}$;
    end_if;
    $a\_list \leftarrow$ empty_list;
end_for
return $\mathcal{G}_{best}$;
```

```
Procedure P2P_Const_Phase;
Input: $G$; $a\_list, n$;

$p\_a\_list \leftarrow$ Possible_Assignments($G$) ;
while not_empty($p\_a\_list$) do
    $RCL \leftarrow$ Compute_RCL($p\_a\_list, n$);
    $current\_a \leftarrow$ SelectRandom($RCL$);
    $a\_list \leftarrow$ Add_Assignment($a\_list, current\_a$);
    $p\_a\_list \leftarrow$ Update_PA($G, current\_a, p\_a\_list$);
end_while;
$\mathcal{G}_{sol} \leftarrow$ Apply_Assignments($G, a\_list$);
return $\mathcal{G}_{sol}$, $a\_list$;
```

Fig. 2. Customized GRASP and Construction Phase.

remain. It might be also the case that there is no more available bandwidth in any node of the main tree, thus, no allocation is possible and p_a_list becomes empty. These cases are the stop conditions.

Local search phase. The previous procedure generates a random solution a_list. To improve the solution constructed, a local search is applied. Following directly the ideas in [1], we use a RNN in the local search phase. After the m executions of the algorithm, $\hat{m}$ different initial solutions are obtained: $a\_list_1, a\_list_2, \ldots, a\_list_{\hat{m}}$.

Metric for enhancement of a given assignment. Selecting one solution amongst the $\hat{m}$ constructed is done by a Monte Carlo estimation of their future PSQA (by randomly simulating sequences of nodes entries and exits). The selected solution will be the one with higher mean PSQA.

3 Numerical Results and Discussion

We compare the performance of our P2P assignment metaheuristic with a traditional Content Delivery Network (CDN) [2] based implementation. Two important variables have to be considered in the analysis: the global perceived quality of the network, and the total bandwidth consumption (at the broadcaster and at the peers). We have statistical data coming from a live video delivery service of a medium size ISP, with 10.000 access of different users per month, and an average of 100 concurrent users per live-TV channel.

In the CDN architecture case, the broadcaster is a set of servers in the ISP datacenter, where the bandwidth consumption is the most expensive component cost of the service. The broadcaster absorbs all the load of the clients, with a stream of 512 Kbps; this means at least 50 Mbps of bandwidth in peak use. The broadcaster of the CDN study case has no failures in the service. If we consider the packet losses in the network negligible, we can assume that the CDN network has a perfect global quality (i.e $Q = 1.00$). We simulate a P2P architecture with the same clients' behavior (connections/disconnections) than the real CDN. Our results show that the broadcaster absorbs only 5.6 Mbps, and the peers the rest of the load (in average 0.6 Mbps per peer). The quality is not considerably degraded, with a worst case of $Q = 0.966$ on the average.

As main conclusions, we think that the numerical results obtained show the interest of employing a live-video P2P distribution system following a multi-source procedure. The PSQA technique allows to automatically measure the perceived quality as seen by the final users, and this in turn is a key issue in developing a formalized model and algorithmic solution procedures to define the topology and flow assignment of such a system.

References

[1] Cancela, H., F. Robledo and G. Rubino, *A GRASP algorithm with RNN based local search for designing a WAN access network*, Electronic Notes on Discrete Mathematics **18C** (2004), pp. 53–58.

[2] Cranor, C., M. Green, C. Kalmanek, D. Shur, S. Sibal and J. V. der Merwe, *Enhanced streaming services in a content distribution network*, IEEE Internet Computing (2001), pp. 66–75.

[3] Mohamed, S. and G. Rubino, *A Study of Real–time Packet Video Quality Using Random Neural Networks*, IEEE Transactions On Circuits and Systems for Video Technology **12** (2002), pp. 1071 –1083.

[4] Resende, M. G. C. and C. C. Ribeiro, "Greedy Randomized Adaptive Search Procedures," In F. Glover and G. Kochenberger, editors, Handbook of Methaheuristics, Kluwer Academic Publishers, 2003, 219-249 pp.

Integer Programming Formulations for the k-Cardinality Tree Problem

Frederico P. Quintão, Alexandre Salles da Cunha [1]
Geraldo R. Mateus

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais – Belo Horizonte, Brasil,
{fred,acunha,mateus}@dcc.ufmg.br

Abstract

In this paper, we present two Integer Programming formulations for the k-Cardinality Tree Problem. The first is a multiflow formulation while the second uses a lifting of the Miller-Tucker-Zemlin constraints. Based on our computational experience, we suggest a two-phase exact solution approach that combines two different solution techniques, each one exploring one of the proposed formulations.

Keywords: k-Cardinality Tree Problem, Integer Programming

1 Introduction

Let $G = (V, E)$ be a connected undirected graph with a set of vertices V ($n = |V|$) and a set of edges E ($m = |E|$). Assume that costs $\{c_e : e \in E\}$ as well as weights $\{d_v : v \in V\}$ are respectively assigned to the edges and vertices of G. The cost of a tree $T = (V_T, E_T)$ in G is given by the sum of its edges' costs $c(E_T) := \sum_{e \in E_T} c_e$ plus the sum of the weights of its spanned

[1] Research supported by Programa Recém Doutores - UFMG/PRPq-2006

vertices $d(V_T) := \sum_{v \in V_T} d_v$. In the k-Cardinality Tree Problem (KCT), one seeks a minimal cost tree of G with exactly k edges.

Recently, KCT has been extensively studied in the literature. Most of the studies concentrate, however, in the design of heuristic and metaheuristic procedures: the Local Searches in [2], the Tabu Search in [5], the VNS in [3], just to name a few. The complexity of the problem has been investigated in [1]; KCT being proved to be NP-hard if $2 \leq k \leq n-2$. To the best of our knowledge, reference [1] is the only study where exact solution methods are discussed. In [1], a polyhedral study and an Integer Programming formulation based on the Generalized Subtour Elimination Constraints (GSECs) were presented for KCT. Finally, the works in [7] and [6] are dedicated to the rooted version of the problem.

The aim of this paper is to present two compact IP formulations for KCT. In Section 2, we present these formulations. Preliminary computational results of Branch-and-bound algorithms based on them are discussed in Section 3. Based on our computational results, we also introduce a two-phase exact solution approach that combines Lagrangian Relaxation and Local Branching [10]. The paper is closed in Section 4.

2 Integer Programming Formulations

In this Section, we formulate KCT as an arborescence in a conveniently defined digraph $D = (\overline{V}, A)$ obtained from the vertices and edges of G as follows. First, let's introduce in V two new vertices, $n+1$ and $n+2$, resulting $\overline{V} := V \cup \{n+1, n+2\}$. Now let A be the union of the following sets: (i) the set of arcs leaving $n+1$, $\{[n+1, i] : i \in V\}$ and $\{[n+1, n+2]\}$, with costs $\{c_{[n+1,i]} = 0 : i \in V\}$ and $\{c_{[n+1,n+2]} = 0\}$ (ii) the set of arcs leaving $n+2$, $\{[n+2, i] : i \in V\}$, with costs $\{c_{[n+2,i]} = d_i : i \in V\}$ and (iii) the set of pairs of arcs obtained from each edge of E, $A_E = \{[i, j], [j, i] : (i, j) \in E\}$, with costs $c_{[i,j]} = c_{(i,j)} + d_j$ and $c_{[j,i]} = c_{(i,j)} + d_i$.

Note that one can map each feasible tree T for KCT in G to an arborescence rooted at $n+1$ in D, satisfying additional constraints (see Figure 1): (i) the $n - (k+1)$ vertices not spanned by T (dashed circles in the Figure) are connected to vertex $n+1$ by arc $[n+1, i]$; (ii) there is a path in D from $n+1$ to every spanned vertex that necessarily includes the arc $[n+1, n+2]$ and (iii) among those vertices spanned by T, only one is directly connected to vertex $n+2$.

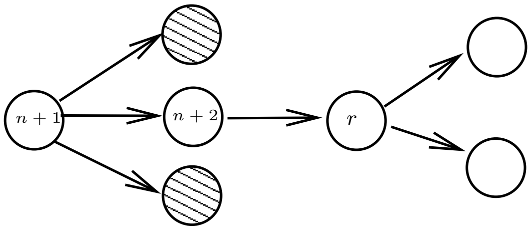


Fig. 1. A feasible solution in D with $n = 5$ and $k = 2$

2.1 A Multiflow Formulation

To each vertex $i \in V' := V \cup \{n+2\}$, we associate one unity of a commodity i, available at root $n+1$, that must be delivered to its corresponding destination vertex through a path in D. In this model, we use binary decision variables $\{x_a : a \in A\}$ assuming value 1 if arc a is used in any path (0, otherwise) and nonnegative real valued variables f_a^q representing the quantity of commodity $q \in V'$ that flows through arc $a \in A$. Before proceeding, assume that $\delta^+(\{i\}) = \delta^+(i)$ and $\delta^-(\{i\}) = \delta^-(i)$ denote, respectively, the set of arcs of A leaving and entering $i \in \overline{V}$. The multiflow formulation (MFF) is:

$$\min \sum_{a \in A} c_a x_a \tag{1}$$

$$\sum_{a \in \delta^+(n+1)} f_a^q = 1, \;\; \forall q \in V', \tag{2}$$

$$\sum_{a \in \delta^+(i)} f_a^q - \sum_{a \in \delta^-(i)} f_a^q = 0, \;\; \forall q, i \in V', \;\; i \neq q, \tag{3}$$

$$\sum_{a \in \delta^+(q)} f_a^q - \sum_{a \in \delta^-(q)} f_a^q = -1, \;\; \forall q \in V', \tag{4}$$

$$\sum_{q \in V'} f_{[n+1,n+2]}^q = k + 2, \tag{5}$$

$$\sum_{a \in \delta^+(n+2)} x_a = 1, \tag{6}$$

$$\sum_{a \in \delta^+(n+1)} x_a = n - k, \tag{7}$$

$$f_a^q \leq x_a, \;\; \forall q \in V', \;\; \forall a \in A, \tag{8}$$

$$x_a \in \mathbb{B}, \; \forall a \in A, \;\; f_a^q \in \mathbb{R}_+, \; \forall a \in A, \; \forall q \in V'. \tag{9}$$

Note that according to previously defined arc costs, the objective function (1) correctly captures the cost of a KCT solution. Constraints (2)-(4) impose flow balance of each commodity at every vertex in $\overline{V}$ and eliminate cycles from the solutions. Constraints (8) ensure that f_a^q assume positive values only if arc a is selected. Finally, constraints (5)-(7) guarantee that a feasible solution must include k arcs $[i, j] \in A_E$.

2.2 A formulation based on the Miller-Tucker-Zemlin constraints

Assume now that set A also includes zero cost arcs $\{[i, n+1] : i \in V\}$ and $\{[i, n+2] : i \in V\}$. In addition to binary decision variables x_a, our second model uses level variables $u_i \in \mathbb{R}_+$, $i \in \overline{V}$ indicating the number of arcs in the path between $n+1$ and i in any feasible arborescence in D. The MTZF formulation is then:

$$\min \sum_{a \in A} c_a x_a \tag{10}$$

$$\sum_{a \in \delta^-(j)} x_a = 1, \forall j \in V', \tag{11}$$

$$\sum_{i \in V'} x_{[n+1,i]} = n - k, \tag{12}$$

$$\sum_{a \in A_E} x_a = k, \tag{13}$$

$$x_{[n+1,i]} + x_{[i,j]} \leq 1, \forall [i,j] \in A, \tag{14}$$

$$(k+3)x_{[i,j]} + u_i - u_j + (k+1)x_{[j,i]} \leq (k+2), \forall [i,j] \in A, \tag{15}$$

$$x_{[n+1,n+2]} = 1, \; x_a \in \mathbb{B}, \forall a \in A, \tag{16}$$

$$u_{n+1} = 0, \;\; u_i \geq 0, \forall i \in V'. \tag{17}$$

Inequalities (15) are a lifting of the well known Miller-Tucker-Zemlin constraints [8,9] that guarantee the solution is cycle-free.

3 Preliminary Computational Results

The computational results we carried out comprise evaluating the formulations presented here by the strength of their Linear Programming (LP) relaxation bounds and also by the overall performance of Branch-and-bound (BB) algorithms based on them. These algorithms were implemented using CPLEX package, version 9.1.3, with default parameters. Our tests were carried out on an Intel Pentium IV machine with 2.4GHz, 1GB of RAM and 512KB of cache, under Linux Operating System. Two sets of instances were used to evaluate the proposed formulations: KLIB, coming from the KCTLib [2], and GRID, grid instances generated as proposed in [3]. For reasons of space, we do not quote in Table 1 results for all instances in our test bed. Since CPLEX ran out of memory when attempting to solve some KLIB instances (g-400-4-02, g-400-4-04 and g-400-4-05) with MFF formulation, we only report results for those instances with 400 vertices that were solved by both models. All other KLIB instances with $n < 400$ were solved by MTZF and MFF.

[2] KCT instance on-line archive – `http://iridia.ulb.ac.be/~cblum/kctlib/`

					MFF			MTZF	
Instance	$\|V\|$	$\|E\|$	k	Opt	$t(s)$	$t_{LP}(s)$	gap(%)	$t(s)$	gap(%)
g200-4-01	200	400	20	308	3270.50	1085.84	0.83	388.63	86.67
g200-4-02	200	400	20	299	2099.29	1012.18	0.50	149.53	113.57
g200-4-03	200	400	20	300	1287.81	1246.2	0.00	59.10	156.41
g200-4-04	200	399	20	304	113141.96	1124.94	3.05	215.96	112.59
g200-4-05	200	399	20	357	13405.98	1122.81	0.97	418.27	111.24
g400-4-01	400	800	20	253	3804.38	835.00	0.06	576.61	145.63
g400-4-03	400	799	20	302	7880.70	884.19	0.89	9924.65	137.80
grid_10_10_1	100	180	10	24746	11609.20	207.22	1.55	93.61	76.05
grid_10_20_1	100	180	20	34409	1152.94	106.17	0.40	1009.45	47.40
grid_10_30_1	100	180	30	68401	6422.59	140.01	1.26	422.14	14.27
grid_15_15_1	225	420	15	23594	710.84	698.34	0.00	2247.56	123.57
grid_15_15_4	225	420	15	27411	1445.71	1443.42	0.00	3137.46	125.90
grid_15_15_6	225	420	15	24017	892.43	532.67	0.10	238.48	79.85

Table 1

Computational results for some KLIB and GRID instances.

In the first column of Table 1, instances are indentified by their names. The next three columns are $|V|$ and $|E|$, followed by k. In the next column, optimal objective function values (OPT) are shown. In the following three columns, we present results for MFF. They are: the CPU time taken to run the BB algorithm to completion, $t(s)$, the CPU time required to evaluate the LP relaxation of (1)-(9), $t_{LP}(s)$, and the duality gaps between these LP bounds and optimal objective values. Finally, in the last two columns, we report on similar entries for MTZF: the total BB CPU time, $t(s)$, and the duality gap. All CPU times are quoted in seconds.

All KLIB and GRID instances indicated in Table 1 were solved to proven optimality by both BB algorithms. To the best of our knowledge, no optimality certificates were previously available to these KLIB instances in the literature.

The results in Table 1 suggest MFF LP bounds are stronger than MTZF counterparts. However, they are very expensive to evaluate. When these factors are balanced together, the MTZF based BB algorithm proved to be, in most cases in our computational testings, much faster than the BB that uses MFF. Therefore, we propose a two-phase exact solution method: the first phase is a Lagrangian heuristic using MFF while the second is a Local Branching [10] approach using MTZF. In doing so, our aim is to take the most out of each formulation by a conveniently chosen solution method. In the first phase, inequalities (2)-(5) are going to be relaxed in a Lagrangian fashion and (Lagrangian) modified costs will be used to drive a constructive heuristic to find feasible KCT solutions. Since MFF LP bounds were good in our testings and the heuristic is driven by dual information, these upper bounds should also be good. If the first phase does not solve KCT alone, it will then work as a starter to the next. With the Lagrangian upper bound in hands, a Local Branching algorithm based on MTZF should run faster than a BB algorithm based on MTZF alone.

4 Final remarks

We presented two IP formulations for the k-Cardinality Tree Problem. The first is multiflow formulation and the second uses the Miller-Tucker-Zemlin constraints. Our computational experience suggests MFF provides bounds stronger than MTZF counterparts at a higher computational time. As a result, we suggested a two-step hybrid solution approach that uses both formulations. We plan to compare the proposed method with a Branch-and-cut algorithm where GSECs are used as cutting planes as in [1].

References

[1] M. Fischetti, H. Hamacher, K. Jornsten, and F. Maffioli. Weighted k-Cardinality Trees: Complexity and Polyhedral Structure". *Networks*, 24:11–21, 1994.

[2] C. Blum and M. Ehrgott. Local Search Algorithms for the k-Cardinality Tree Problem. *Discrete Appl. Math.*, 128(2-3):511–540, 2003.

[3] J. Brimberg, D. Urosevic, and N. Mladenovic. Variable Neighborhood Search for the Vertex Weighted k-Cardinality Tree Problem. *EJOR*, 171:74–84, 2006.

[4] M. Ehrgott, J. Freitag, H. Hamacher, and F. Maffioli. Heuristics for the k-Cardinality Tree and Subgraph Problems. *Asia Pacific J. Oper. Res.*, 14(1):87–114, 1997.

[5] K. Jornsten and A. Lokketangen. Tabu Search for Weighted k-Cardinality Trees. *Asia Pacific J. Oper. Res.*, 14(1):9–26, 1997.

[6] S. Kataoka, N. Araki, and T. Yamada. Upper and Lower Bounding Procedures for Minimum Rooted k-subtree Problem. *EJOR*, 122:561–569, 2000.

[7] J. M. M. Coimbra. Limites Inferiores e Limites Superiores para o Problema da q-Árvore de Custo Máximo. Master Thesis, Universidade de Lisboa, 1995.

[8] C. Miller, A. Tucker and R. Zemlin. Integer Programming Formulations and Travelling Salesman Problems. *Journal of the ACM*, 7:326-329, 1960.

[9] M. Desrochers and G. Laporte. Improvements and Extensions to the Miller-Tucker-Zemlin Subtour Elimination Constraints. *Ops. Res. Lett.*, 10:27-36, 1991.

[10] M. Fischetti and A. Lodi. Local Branching. *Math. Programming*, 98:23-47, 2003.

Totally Multicolored diamonds

Juan José Montellano-Ballesteros [1]

Instituto de Matemáticas
Universidad Nacional Autónoma de México
México, D.F. México

Abstract

Let G be a graph of order p. For every $n \geq p$ let $f(n, G)$ be the minimum integer k such that for every edge-coloring of the complete graph of order n which uses exactly k colors, there is at least one copy of G all whose edges have different colors. Let $\mathbb{F}$ be a set of graphs. For every $n \geq 3$ let $ext(n, \mathbb{F})$ be the maximum number of edges of a graph on n vertices with no subgraph isomorphic to an element of $\mathbb{F}$. Here we study the relation between $f(n, G)$ and $ext(n, \mathbb{C}(G))$ when G is a graph with chromatic number 3 obtained by adding an edge (a chord) to a cycle, and $\mathbb{C}(G)$ is the set of cycles which are subgraphs of G. In particular, an upperbound and a lowerbound of $f(n, G)$ are given; and in the case when G is the diamond (C_4 with a chord), we prove that the supremum and infimum limits of $\frac{f(n,G)}{n\sqrt{n}}$ are bounded by $\frac{2}{3}$ and $\frac{1}{2\sqrt{2}}$ respectively, and we conjecture that for every $n \geq 4$, $f(n, G) = ext(n, \{C_3, C_4\}) + 2$.

Keywords: Colorings, totally multicolored, anti-Ramsey.

1 Introduction

Let K_n denotes the complete graph of order n. A subgraph G of K_n is said to be *totally multicolored* (with respect to a given edge-coloring of K_n) if G

[1] Email: juancho@matem.unam.mx

contains no two edges of the same color. The problem of determining, given a graph G and an integer $n \geq 3$, the minimum integer $f(n, G)$ such that every edge-coloring of K_n which uses exactly $f(n, G)$ colors, leaves at least one totally multicolored copy of G (TMC for short), was introduced in [3] and has been widely studied (see, for instance, [1], [4], [5], [6], [7]).

In [3] it is showed that $f(n, G)$ is at least $ext(n, \mathbb{L}(G)) + 2$, where $\mathbb{L}(G)$ is the family of all subgraphs obtained from G by removing an edge, and that $ext(n, \{H_G\})$ is an upperbound for $f(n, G)$, where H_G is obtained in the following way: consider $M \in \mathbb{L}(G)$ of minimum chromatic number. By definition, $M = G - xy$ for some $xy \in E(G)$. Take two copies, M' and M", of M and obtain H_G by identifying the vertices x' with x" and y' with y".

In [2] the asintotic behavior of $ext(n, G)$ is determinated, provide $\mathbb{L}(G)$ contains no bipartite graph. Using this, in [3] the asintotic behavior of $f(n, G)$ is determinated, provide G is not bipartite or G contains no edge $e \in E(G)$ such that $G - e$ is bipartite. Although, as it was proof in [3], for every graph G, $ext(n, \{H_G\}) \geq f(n, G) \geq ext(n, \mathbb{L}(G))$, in the case when G is bipartite or G contains an edge $e \in E(G)$ such that $G - e$ is bipartite, the behavior of $ext(n, \mathbb{L}(G))$ and $ext(n, \{H_G\})$ are, in general, unknown.

Some studies concerning $f(n, G)$ when G is bipartite or G contains an edge $e \in E(G)$ such that $G - e$ is bipartite are [1], [4], [6] and [7].

In this paper we focus our attention in the relation between $f(n, G)$ and $ext(n, \mathbb{C}(G))$ when G is a graph of chromatic number 3 (a 3-*chromatic graph*) obtained by adding a chord to a cycle, and $\mathbb{C}(G)$ is the set of cycles which are subgraphs of G (observe that $\mathbb{C}(G)$ contains always an even cycle). We present a lowerbound of $f(n, G)$ ($ext(n, \mathbb{C}(G)) + 2$); and an upperbound of $f(n, G)$ which definition depends on the maximum of the set of minimum degrees of the graphs which has no element of $\mathbb{C}(G)$ as a subgraph. Clearly, if the behavior of $ext(n, \mathbb{C}(G))$ is unkown, so will be the one of $f(n, G)$, but since the upperbound depends on the minimum degrees of the graphs of that particular family, this approach allow us to study the gap between the given upperbound and lowerbound.

1.1 Notation and preliminary results

Let G be a graph. $V(G)$, $E(G)$ and $\delta(G)$ will denote the set of vertices, the set of edges and the minimum degree of G respectively. Given $S \subseteq E(G)$, the subgraph of G induced by S is the subgraph H of G of minimum order such that $E(H) = S$. Given a family of graphs $\mathbb{P}$, let $\mathbb{F}_n[\mathbb{P}]$ be the family of graphs of order n which contains no element of $\mathbb{P}$ as a subgraph, and let

$\delta(\mathbb{F}_n[\mathbb{P}]) = max\{\delta(G) : G \in \mathbb{F}_n[\mathbb{P}]\}$. A cycle C belongs to $\mathbb{C}(G)$ if and only if there is a subgraph H of G which is isomorphic to C.

Let D be a digraph. $V(D)$, $A(D)$ and $gr(D)$ will denote the set of vertices, the set of arcs and the underlying (simple) graph of D, respectively. If $x \in V(D)$, $N_D^+(x)$ and $N_D^-(x)$ will denote the out-neighborhood and the inner-neighborhood of x in D, respectively; $d_D^+(x)$ and $sd_D^+(x)$ will denote the cardinality of $N_D^+(x)$ (the out-degree of x) and the cardinality of $N_D^+(x) \setminus N_D^-(x)$, respectively. $\delta^+(D)$ will denote the minimum out-degree of D. Given a family of digraphs $\overrightarrow{\mathbb{P}}$, let $\mathbb{F}_n[\overrightarrow{\mathbb{P}}]$ be the family of digraphs of order n which contains no element of $\overrightarrow{\mathbb{P}}$ as a subdigraph, and $\delta^+(\mathbb{F}_n[\overrightarrow{\mathbb{P}}]) = max\{\delta^+(D) : D \in \mathbb{F}_n[\overrightarrow{\mathbb{P}}]\}$.

Let $n \geq 3$ be an integer. As an *edge-coloring of* K_n we will understand a function $\Gamma : E(K_n) \to \mathbb{N}$. Given an edge-coloring Γ of K_n, $\Gamma[K_n]$ denotes the image of Γ. Γ will be called *G-bad* if it leaves no TMC copy of G and will be called *extremal G-bad* if it is G-bad and $|\Gamma[K_n]| = f(n, G) - 1$.

Let Γ be an edge-coloring of K_n. If $x \in V(K_n)$, $\nu(x, \Gamma, K_n)$ will denote the difference $|\Gamma[K_n]| - |\Gamma[K_n \setminus \{x\}]|$, i.e. the number of colors of Γ that only appear in edges which are incidents to x. Given $x \in V(K_n)$, a subset W of $V(K_n) \setminus \{x\}$ will be said to be a *selective* (x, Γ)*-set* provided $|W| = \nu(x, \Gamma, K_n)$ and all the xW-edges have different colors which do not appear in $\Gamma[K_n \setminus \{x\}]$. A digraph D will be called a *selective* (Γ, K_n)*-digraph* provided $V(D) = V(K_n)$ and that for each $x \in V(D)$, $N_D^+(x)$ is a selective (x, Γ)-set.

Observe that given a selective (Γ, K_n)-digraph D, the subgraph of K_n induced by the set of edges of $gr(D)$ is TMC with respect to Γ in K_n; and the colors appearing in such subgraph are exactly those colors $i \in \mathbb{N}$ such that $\Gamma^{-1}(i)$ is a star (maybe an edge). $\Gamma^{-1}(i)$ its the edge xy if and only if $x \in N_D^+(y)$ and $y \in N_D^+(x)$.

Lemma 1.1 *Let Γ be an edge-coloring of K_n, and let D be a selective (Γ, K_n)-digraph. Let H be a subdigraph of D with two non-adjacent vertices, $x, y \in V(H)$, such that $sd_H^+(x) = sd_H^+(y) = 0$.*

Then the subgraph of K_n induced by $E(gr(H)) \cup \{xy\}$ is TMC.

Sketch of the proof. Let suppose there is such subdigraph H in D. By definition of D, this implies the subgraph of K_n induced by $E(gr(H))$ is TMC. If there is $vw \in E(gr(H))$ such that $\Gamma(\{vw\}) = \Gamma(\{xy\})$, again by definition, $|\{x, y\} \cap \{w, v\}| = 1$ and so $sd_H^+(x) > 0$ or $sd_H^+(y) > 0$. □

Lemma 1.2 *Let G be a graph of order p and let Γ be an extremal G-bad edge-coloring of K_{n+1}. If $n \geq p$ then*

$$f(n+1, G) - f(n, G) \leq \nu^*(\Gamma, K_{n+1})$$

where $\nu^*(\Gamma, K_{n+1}) = min\{\nu(x, \Gamma, K_{n+1}) : x \in V(K_{n+1})\}$.

Sketch of the proof. Given $x \in V(K_{n+1})$, the number of colors appearing in $E(K_{n+1})$ $(f(n+1, G) - 1)$ is equal to the number of colors appearing in $E(K_{n+1} - x)$ (which is at most $f(n, G) - 1$) plus the colors that disappear by deleting x, i.e. the number of colors of Γ that only appear in edges which are incidents to x ($\nu(x, \Gamma, K_n)$). □

From Lemmas 1.1 and 1.2 we can see the following: Given a graph G, let $\overrightarrow{\mathbb{P}}(G)$ be the set of digraphs H such that for some $e = xy \in E(G)$, $gr(H) \cong G - e$ and $sd^+_H(x) = sd^+_H(y) = 0$. Then, given a G-bad edge-coloring Γ of K_{n+1}, any selective (Γ, K_{n+1})-digraph D belongs to $\mathbb{F}_{n+1}[\overrightarrow{\mathbb{P}}(G)]$ and, since $\nu^*(\Gamma, K_{n+1}) = \delta^+(D)$, the difference $f(n+1, G) - f(n, G)$ is bounded by $\delta^+(\mathbb{F}_{n+1}[\overrightarrow{\mathbb{P}}(G)])$.

In the case when G is a 3-chromatic graph obtained by adding a chord to a cycle, we can say a little more, as we see in the next lemma.

Lemma 1.3 *Let G be a 3-chromatic graph of order p obtained by adding a chord to the cycle C_p. Let Γ be an extremal G-bad edge-coloring of K_n and let D be a selective (Γ, K_n)-digraph. Then, either*

i) $gr(D)$ has no subgraph isomorphic to an element of $\mathbb{C}(G)$; or

ii) for every $x \in V(K_n)$, $\nu(x, \Gamma, K_n) \leq p - 2$.

Sketch of the proof. Let suppose $\nu(x, \Gamma, K_n) \geq p - 1$. Let $\mathbb{C}(G) = \{C_p, C_q, C_r\}$ and suppose there is a subdigraph H of D such that $gr(H)$ is isomorphic to C_r. For the case $q \geq 4$, let $x, y \in V(H)$ be two adjacent vertices in H. Since every vertex $x \in V(D)$ has out-degree at least $p - 1$, there is not difficult to see that there are two vertex-disjoint directed paths $P_1 = \{x, x_1, \ldots, x_l\}$ and $P_2 = \{y, y_1, \ldots, y_s\}$ in D, with $l + s = q - 2$, which are also vertex-disjoint from $V(H)$. Let M be the subdigraph of D induced by the set of arcs $A(H) \cup A(P_1) \cup A(P_2)$. Since $sd^+_M(x_l) = sd^+_M(y_s) = 0$, by Lemma 1.1 the subgraph of K_n induced by $gr(M) \cup \{x_l y_s\}$ is TMC, but this graph is a copy of G, which is a contradiction. For $q = 3$, let $x \in V(H)$ and $w, v \in V(H)$ be the neighbors of x in H. Since every vertex in D has out-degree at least $p-1$, there is $y \in V(D) \setminus V(H)$ such that $\overrightarrow{xy} \in A(D)$. The subgraph of K_n induced by $E(gr(H)) \cup \{xy\}$ is a TMC copy of $G - e$ with $e = yw$ and $e = yv$. Therefore, the colors $\Gamma(\{yw\})$ and $\Gamma(\{yv\})$ most appear in $\Gamma[E(gr(H)) \cup \{xy\}]$ which, by definition of D, implies that $\Gamma(\{yw\}) = \Gamma(\{wx\})$ and $\Gamma(\{yv\}) = \Gamma(\{vx\}$. So $sd^+_H(x) = 0$, $sd^+_H(w) \geq 1$ and $sd^+_H(v) \geq 1$. Since x is an arbitrary vertex of H, this is not possible.

Finally, let suppose there is a subdigraph H of D such that $gr(H)$ is isomorphic to C_p. Let $x, y \in V(H)$ such that the graph induced by $E(gr(H)) \cup \{xy\}$ is a copy of G. Thus, there is $wv \in E(gr(H))$ such that $\Gamma(\{xy\}) = \Gamma(\{wv\})$ and, by definition of D, we can suppose that $x = w$. But then, the digraph D' obtained by deleting the arc $\overrightarrow{xv}$ and adding the arc $\overrightarrow{xy}$ is a selective (Γ, K_n)-digraph such that $gr(D')$ contains a subgraph isomorphic to C_q or C_r, which is not possible. □

From Lemma 1.3 we see that in the case when G is a 3-chromatic graph of order p obtained by adding a chord to the cycle C_p, the difference $f(n+1, G) - f(n, G)$ is bounded, either by $p-2$, or by $\delta^+(\mathbb{F}_{n+1}[\overrightarrow{\mathbb{C}}(G)])$ where $\overrightarrow{\mathbb{C}}(G)$ is the set of orientations of the elements of $\mathbb{C}(G)$, and so, $\delta^+(\mathbb{F}_{n+1}[\overrightarrow{\mathbb{C}}(G)]) = \delta(\mathbb{F}_{n+1}[\mathbb{C}(G)])$.

2 Main Results

Theorem 2.1 *Let G be a 3-chromatic graph of order p obtained by adding a chord to the cycle C_p. For each $j \geq p+1$ let $h(j) = max\{p-2, \frac{2ext(j,\mathbb{C}(G))}{j}\}$. Then, for every $n \geq p$,*

$$ext(p, \{G\}) + 1 + \sum_{j=p+1}^{n} h(j) \geq f(n, G) \geq ext(n, \mathbb{C}(G)) + 2.$$

Sketch of the proof. Since $\delta(\mathbb{F}_m[\mathbb{C}(G)]) \leq \frac{2ext(m,\mathbb{C}(G))}{m}$, the upperbound follows from Lemmas 1.2 and 1.3. For the lowerbound, consider an edge-coloring of K_n obtained by a TMC copy of a graph $M \in \mathbb{F}_n[\mathbb{C}(G)]$ with $ext(n, \mathbb{C}(G))$ edges, and all the remaining edges colored with one new color. It is not hard to see that this is a G-bad edge-colouring of K_n. □

Theorem 2.2 *Let G be the diamond. Then for every $n \geq 4$,*

$$5 + \frac{4(n-1)^{\frac{3}{2}} + 9(n-1) + 2(n-1)^{\frac{1}{2}}}{6} \geq f(n, G) \geq ext(n, \{C_3, C_4\}) + 2.$$

Sketch of the proof. Analogous as the proof of Theorem 2.1, but just observe that $\delta(\mathbb{F}_m[\mathbb{C}(G)]) = \delta(\mathbb{F}_m[\{C_3, C_4\}]) \leq \sqrt{m-1}$. □

Corollary 2.3 *Let G be the diamond. Then*

$$\frac{2}{3} \geq \sup \lim_{n\to\propto} \frac{f(n,G)}{n\sqrt{n}} \geq \inf \lim_{n\to\propto} \frac{f(n,G)}{n\sqrt{n}} \geq \frac{1}{2\sqrt{2}}.$$

2.1 The conjecture

We conjecture that if G is a diamond, then for every $n \geq 4$

$$f(n,G) = ext(n, \{C_3, C_4\}) + 2.$$

This comes from the following: It is possible to proof that given an extremal G-bad edge-coloring Γ of K_{n+1}, if for some selective (Γ, K_{n+1})-digraph D, $gr(D)$ has diameter at most 3, then Γ uses at most $|E(gr(D)| + 1$ colors and so $f(n+1, G) \leq ext(n+1, \{C_3, C_4\}) + 2$. In other case, $gr(D)$ is not an extremal graph of girth five, and therefore the number of colors appearing in K_{n+1} in the subgraph induced by $E(gr(D))$ is less than $ext(n+1, \{C_3, C_4\})$ and $\nu^*(\Gamma, K_{n+1})$, which bounds the difference between $f(n+1, G)$ and $f(n, G)$, is at most that of the other case.

References

[1] M. Axenovich and T. Jiang. *Anti-Ramsey numbers for small complete bipartite graphs.* Ars Combin. **73** (2004), 311-318.

[2] P. Erdös and M. Simonovits. *A limit theorem in graph theory.* Studia, Sci. Math. Hung., **1** (1966), 51–57.

[3] P. Erdös; M. Simonovits and V.T. Sós. *Anti-Ramsey theorems.* Colloquia Mathematica Societatis János Bolyai **10** Infinite and finite sets, Keszthely, Hungary (1973) 633–643.

[4] T. Jiang. *Edge-colorings with no large polychromatic stars.* Graphs Combin. **18** (2002), no. 2, 303-308.

[5] J. J. Montellano-Ballesteros and V. Neumann-Lara. *An Anti-Ramsey Theorem*, Combinatorica **22** (3), 445–449 (2002).

[6] J.J. Montellano-Ballesteros and V. Neumann-Lara.*An anti-Ramsey theorem on cycles.* Graphs Combin. **21** (2005), no.3, 343-354.

[7] M. Simonovits and V.T. Sós. *On Restricted Colorings of K_n*, Combinatorica **4** (1) (1984) 101–110.

Clique trees of chordal graphs: leafage and 3-asteroidals

M. Gutierrez [a] J. L. Szwarcfiter [b] S. B. Tondato [a]

[a] *Departamento de Matemática, Universidad Nacional de La Plata, CC 172, (1900) La Plata, Argentina*

[b] *Instituto de Matemática and COPPE/Sistemas, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21945-970, Rio de Janeiro, Brazil*

Abstract

Chordal graphs were characterized as those graphs having a tree, called *clique tree*, whose vertices are the cliques of the graph and for every vertex in the graph, the set of cliques that contain it form a subtree of clique tree. In this work, we study the relationship between the clique trees of a chordal graph and its subgraphs. We will prove that clique trees can be described locally and all clique trees of a graph can be obtained from clique trees of subgraphs. In particular, we study the *leafage* of chordal graphs, that is the minimum number of leaves among the clique trees of the graph. It is known that interval graphs are chordal graphs without 3-asteroidals. We will prove a generalization of this result using the framework developed in the present article. We prove that in a clique tree that realizes the leafage, for every vertex of degree at least 3, and every choice of 3 branches incident to it, there is a $3-$asteroidal in these branches.

1 Introduction

Chordal graphs form an important and well studied class of graphs. It is known that for each chordal graph there is a tree, called clique tree, whose vertices are the cliques of the graph and for every vertex in the graph, the set of cliques that contain it form a subtree of clique tree.

Clique trees have many applications, for example it can be used to study protein interactions [7]. Most cellular processes are carried out by multi-proteins complexes, groups of proteins that bind together to perform a specific task. A better understanding of this organization of proteins into overlapping complexes is an important step to unveiling functional and evolutionary mechanisms behind biological networks. This situation can be represented by a graph where the vertices are proteins and two vertices are connected by an edge if the corresponding proteins interact. Complex proteins can be seen as cliques of this graph. Then, when the graph is chordal, a clique tree and the family of subtrees representing the vertices provide a good framework for following the activity of a protein in different complexes.

In this work we study the relationship between the clique trees of a chordal graph and its subgraphs. We will prove that clique trees can be described locally and all clique trees of a graph can be obtained from clique trees of subgraphs. In particular we study the leafage of a chordal graph, that is the minimum number of leaves among all the clique trees of the graph. It is clear that connected interval graphs are exactly those chordal graphs with leafage 2. In an historical work Lekkerkerker and Boland [3] proved that interval graphs are the chordal graphs without 3-asteroidals, 3 vertices such that between any two of them there is a path avoiding the neighborhood of the third. We will prove a generalization of this result using the framework developed in the present article. We prove that in a clique tree that realizes the leafage, for every vertex of degree at least 3, and every choice of 3 branches incident to it, there is a $3-$asteroidal in these branches.

2 Preliminaries

We denote by $G[V]$ the induced subgraph of G whose set of vertices is V. If G_1 and G_2 are graphs, $G_1 \cup G_2$ denotes the graph whose vertices and edges are $V(G_1) \cup V(G_2)$ and $E(G_1) \cup E(G_2)$ respectively. A subset C of V is said to be a clique of G if $G[C]$ is a maximal complete subgraph of G. By $\mathcal{C}(G)$ we denote the set of cliques of G, and for every $v \in V$ we let $C_v = \{C \in \mathcal{C}(G) : v \in C\}$. The intersection graph of a finite family of sets $\mathcal{F} = (F_i)_{i \in I}$ is a graph whose vertices are the sets of the family, with F_i and F_j adjacent whenever $F_i \cap F_j \neq \emptyset$. The *clique graph* of a given graph G, denoted by $K(G)$, is the intersection graph of the family of cliques of G. We consider the clique graphs to be weighted. That is, each edge C_iC_j of it has a weight equal to $|C_i \cap C_j|$.

A graph G is said to be chordal if every cycle in G of length at least four has a chord. Buneman [1], Gavril [2] and Walker [6] independently discovered that a graph is chordal if and only if it is the intersection graph of a family of subtrees of a tree, called *host tree*. It is easy to see that a family of subtrees of a tree has the Helly property, that is every intersecting subfamily has a common vertex. Hence by minimizing the number of vertices of the host tree, preserving the intersection relationship in the family of subtrees, we obtain a host whose set of vertices is $\mathcal{C}(G)$ and the family of subtrees is $(C_v)_{v\in V(G)}$. This is the reason why such tree is called *clique tree*. McKee proved [5] that those trees are exactly the maximum weight spanning trees of $K(G)$.

Theorem 2.1 *G is a chordal graph if and only if there is a maximal spanning tree T of $K(G)$ such that for each v vertex of G, C_v is a subtree of T. Moreover if such tree exists, all maximal spanning trees of $K(G)$ satisfy the condition.*

A *canonical representation* of a chordal graph G is a pair (T, F) where T is a maximal spanning tree of $K(G)$, usually called *clique tree* of G and $F = (C_v)_{v\in V(G)}$. It is easy to prove that $(C_v)_{v\in V(G)}$ is a *separating* family, that is the intersection of all members that contain an element x is $\{x\}$. Moreover, this property characterizes the canonical representations.

Theorem 2.2 *Let (T, F) be a pair where T is a tree and F is a separating family of subtrees of T. Then the intersection graph of F is a chordal graph G, T is a maximal spanning tree of $K(G)$, and $F = (C_v)_{v\in V(G)}$.*

Therefore we can study chordal graphs through their canonical representations or clique trees.

3 Subgraphs of chordal graphs and their clique trees

Clearly, a subgraph G' of a chordal graph G is also chordal. But a clique tree of G' is not necessarily a subtree of a clique tree of G. In Figure 1, there is an example where G is a chordal graph, G' is the subgraph of G induced by the black colored vertices but $K(G')$ is not an induced subgraph of $K(G)$. Moreover, in the example, the clique tree of G' that is a chain is not an induced subtree of the only clique tree of G. However, there is a simples way of taking induced subgraphs G' of a chordal graph G such that each clique tree of G' is a subtree of some clique tree of G.

First we describe some properties of maximum spanning trees of a graph, with weighted edges. Let T be a tree and T_1 a subtree of T, such that $E(T) - E(T_1)$ induces a subtree T_1^c in T. In this case, say that T_1^c is the *complement*

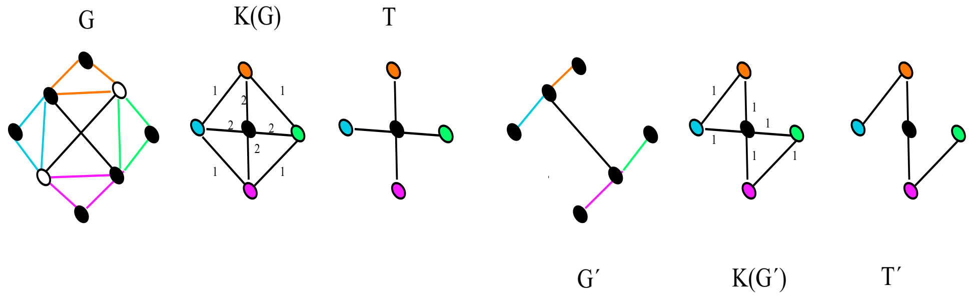


Fig. 1.

of T_1 in T. Clearly, $T = T_1 \cup T_1^c$. If G is a graph whose edges are valuated by positive real numbers, the following theorem implies that the maximum spanning trees can be viewed locally.

Theorem 3.1 *Let T be a maximal spanning tree of G and T_1 a subtree of T that has complement T_1^c. Then $w(T) = w(T_1) + w(T_1^c)$ and T_1 is a maximal spanning tree of $G[V(T_1)]$. Moreover, if T_1' is another maximal spanning tree of $G[V(T_1)]$ then $T_1' \cup T_1^c$ is also a maximal spanning tree of G.*

Therefore, we can obtain subgraphs of chordal graphs as above mentioned. Let G be a chordal graph and T a clique tree of G. If T_1 is a subtree of T, define $V_1 = \{v \in V(G) : T_1 \cap C_v \neq \emptyset\}$ and $G_{T_1} = G[V_1]$.

Theorem 3.2 *Let G be a chordal graph, T a clique tree of G and T_1 a subtree of T having complement. Then G_{T_1} is a chordal graph and T_1 is a clique tree of G_{T_1}.*

The following result says that all clique trees can be obtained by cutting and pasting clique trees.

Theorem 3.3 *Let G be a chordal graph, T a clique tree of G, and T_1 a subtree of T having complement. If T_1' is a clique tree of G_{T_1} then $T_1' \cup T_1^c$ is a clique tree of G.*

Observe that Theorems 3.1, 3.2, 3.3 can be generalized for subtrees with no complements.

Corollary 3.4 *Let G be a chordal graph, T a clique tree of G. If T_1 is a subtree of T then every clique tree of G_{T_1} is a subtree of a clique tree of G.*

4 Leafage of chordal graphs

For a tree T, denote by $ln(T)$ the number of leaves of T. If G is a chordal graph, the *leafage* of G is $l(G) = min\{ln(T),\ T \textit{ a clique tree of } G\}$. A clique tree T is called l-optimal for G if $ln(T) = l(G)$. Clearly, (connected) chordal graphs with leafage equal to 2 are exactly the (connected) interval graphs. On the other hand, as we can see in the example of Figure 1, $l(G') = 2$, $l(G) = 4$ but none of l-optimal clique trees of G' is a subtree of the (unique) clique tree of G. In this section we will prove that if T is l-optimal for G, some of its subtrees are l-optimal for the corresponding subgraph.

Let T_1 be a subtree of a tree T having complement and $v \in V(T_1) \cap V(T_1^c)$. Hence $ln(T) = ln(T_1) + ln(T_1^c)$, if v is not a leaf of T_1 or T_1^c; $ln(T) = ln(T_1) + ln(T_1^c) - 1$, if v is leaf of only one of them; $ln(T) = ln(T_1) + ln(T_1^c) - 2$, if v is a leaf of both.

Theorem 4.1 *Let G be a chordal graph and T a l-optimal clique tree for G. If T_1 is a subtree of T having complement, and $v \in V(T_1) \cap V(T_1^c)$ is not a leaf of T_1 then T_1 is a l-optimal clique tree for G_{T_1}.*

5 3-asteroidals in chordal graphs

Let t be a vertex of a tree T. For each connected component B, obtained by removing t of T, the subtree $T[V(B) \cup \{t\}]$ is a *branch of T incident to t*.

Recall that interval graphs are chordal graphs that have no 3-asteroidals. We will generalize this result proving that for every vertex t of degree at least 3 in a clique tree with minimum number of leaves, there is a 3-asteroidal in branches incident to t.

Theorem 5.1 *Let G be a chordal graph, T a clique tree l-optimal for G, t a vertex of T of degree at least 3 and R_1, R_2, R_3 3 branches of T incident to t. Then there is a 3-asteroidal set $\{x_1, x_2, x_3\}$ of G with $x_i \in R_i$ for all $i \in \{1, 2, 3\}$.*

Proof. By induction on $l(G)$. If $l(G) = 3$ then G is not an interval graph and G has at least a 3-asteroidal [3]. Since $l(G) = 3$, there is only one vertex of degree 3 in T. It is easy to see that no pair of vertices of a 3-asteroidal can be in the same branch of T incident to t. Hence the Theorem is true.

By the inductive hypothesis, the theorem holds if $l(G) < k$. Suppose $l(G) = k$. Then, there is a tree T, such that $ln(T) = l(G)$. Let t be a vertex of T of degree at least 3 and R_1, R_2, R_3 3 branches incident to t. Let $T_1 = R_1 \cup R_2 \cup R_3$. If $T_1 \neq T$ by Theorem 3.2 it follows that G_{T_1} is a chordal graph and T_1 a clique tree of it. Since t is not a leaf of T_1 and $T \neq T_1$, we know that $l(G_{T_1}) < k$. By the inductive hypothesis, since R_1, R_2, R_3 are 3 branches in T_1, there is a 3-asteroidal of G_{T_1}, with a vertex in each R_i. Hence there is a 3-asteroidal of G as required. In the case that $T_1 = T$, then t has degree 3. Since $l(G) > 3$, there is another vertex t' in T of degree at least 3. Suppose that $t' \in R_3$. Let R be a branch of T incident to t', not containing t. Now, we define T_1 as R^c and $T_1^c = R$. Consequently, G_{T_1} is a chordal graph and T_1 is a clique tree of it. Since t' is not a leaf of T_1, by Theorem 4.1, $l(G_{T_1}) = ln(T_1) < ln(T) = k$. By the inductive hypothesis, as t is a vertex of degree 3 in T_1 and $R_1, R_2, R_3' = R_3 \cap T_1$ are three branches of T_1 incident to t. Then there is 3-asteroidal of G_{T_1} in these branches. Hence there is a 3-asteroidal of G, as required. □

References

[1] P. Buneman. A characterization of rigid circuit graphs. Discrete Math., 9:205-212, 1974.

[2] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, *Journal of Combinatorial Theory (Series B)* **16**, 47–56, 1974.

[3] C.G. Lekkerkerker and J.C. Boland, Representation of a finite graph by a set of intervals on the real line, *Fundamenta Mathematicae* **51**, 45–64, 1962.

[4] I. J. Lin, T.A. McKee and D.B. West, *The leafage of a chordal graphs*, Math/CO/9807022v/3 Jul 1998.

[5] T.A. McKee and F.R. McMorris, *Topics in Intersection Graph Theory*, Monographs on Discrete Mathematics and Applications, vol. 2 SIAM, Philadelphia, 1999.

[6] J.R. Walter, Representations of chordal graphs as subtrees of a tree, *Journal of Graph Theory* **2**, 265–267, 1978.

[7] E. Zotenko, K. S. Guimarães, R. Jothi and T. M. Przytycka. Decomposition of overlapping protein complexes: A graph theorical method for analyzing static and dynamic protein associations. Algorithms for Molecular Biology 1:7, 2006.

Repetition-free longest common subsequence

S.S. Adi M.D.V. Braga C.G. Fernandes C.E. Ferreira [1]
F.V. Martinez M.-F. Sagot M.A. Stefanes C. Tjandraatmadja
Y. Wakabayashi

Univ. Fed. do Mato Grosso do Sul, Univ. de São Paulo, Brazil, and Univ. C. Bernard, Lyon I, France

Abstract

We study the problem of, given two finite sequences x and y, finding a repetition-free longest common subsequence of x and y. We show some algorithmic results, a complexity result, and a preliminary experimental study based on the proposed algorithms.

Keywords: Longest common subsequence, APX-hard, approximation algorithms.

1 Introduction

In the genome rearrangement domain, gene duplication is rarely considered as it usually makes the problem at hand harder. Sankoff [6] proposed the so called exemplar model, which consists in searching, for each family of duplicated genes, an exemplar representative in each genome. In biological terms, the exemplar gene may correspond to the original copy of the gene, which later originated all other copies. Following the parsimony principle, the choices of exemplars should be made so as to minimize the reversal distance between the two simpler versions of both genomes, composed only by the exemplar genes.

[1] Email of corresponding author: `cef@ime.usp.br`

An alternative to the exemplar model is the multigene family model, which consists in maximizing the number of paired genes among a family. Again, the gene pairs should be chosen so as to minimize the reversal distance between the genomes. Both exemplar and multigene models were proven to lead to NP-hard problems [2,4].

To compare two sequences, we propose a similarity measure that takes into account the concept of exemplar genes. The measure is the length of a repetition-free longest common subsequence (LCS) of the two sequences. The concept behind the exemplar model is captured by the repetition-free requirement in the sense that at most one representative of each family of duplicated genes is taken into account. The length of an LCS is a measure of similarity between sequences, so the length of a repetition-free LCS can be seen as the edit distance between two sequences where only deletions are allowed and, furthermore, for each family with k duplicated genes, at least $k-1$ of them must be deleted.

The problem we are interested, denoted by RFLCS, consists of the following: given two sequences x and y, find a repetition-free LCS of x and y. We write RFLCS(x, y) when we refer to RFLCS for a generic instance (x, y). We denote by opt(RFLCS(x, y)) the length of an optimal solution of RFLCS(x, y).

Bonizzoni et al. [3] considered some variants of the RFLCS, such as the case where some symbols are required to appear in the sought LCS, possibly more than once. They showed that these variants are APX-hard and that, in some cases, it is NP-complete just to decide whether an instance of the variants is feasible. This second complexity result makes these variants less tractable.

We present some algorithmic and some hardness results for the RFLCS and report on some computational experiments with the algorithms proposed.

2 Algorithmic results

We first mention some polynomially solvable cases of RFLCS(x, y). If each symbol appears at most once either in x or in y then the problem is easy: it is enough to find an LCS of x and y. In this case, any LCS has no repetition and is therefore a solution of RFLCS(x, y). There are polynomial algorithms for LCS, so this case is polynomially solvable.

For each symbol a and a sequence w, let $n(w, a)$ be the number of appearances of a in w. Let $m_a(x, y) = \min\{n(x, a), n(y, a)\}$. The case above is the one in which $m_a(x, y) \leq 1$ for all a. Consider the slightly more general case in which there is a constant bound k on the number of symbols a for which $m_a(x, y) > 1$. It is not hard to see that this case is also polynomially solvable.

Now we describe three simple approximation algorithms for the problem: A1, A2, and A3. Algorithm A1 consists of the following: given x and y, compute an LCS of x and y and remove all repeated symbols but one, in the obtained LCS. Return the resulting sequence. Let m be the maximum value of $m_a(x, y)$ taken over all a. It is not hard to see that Algorithm A1 is an m-approximation for RFLCS(x, y).

Algorithm A2 is probabilistic. It consists of the following: given x and y, for each symbol a, if $m_a(x, y) = n(x, a)$, pick uniformly at random one of the $m_a(x, y)$ occurrences of a in x, and delete all the others from x; if $m_a(x, y) \neq n(x, a)$, pick uniformly at random one of the $m_a(x, y)$ occurrences of a in y, and delete all the others from y. Let x' and y' be the resulting sequences after this clean-up. Compute an LCS w' of x' and y' and return w'.

Algorithm A3 is a variant of Algorithm A2 that uses less random bits. Instead of choosing independently one of the occurrences of each symbol in the sequences, A3 picks uniformly at random only one number in the interval $[0, 1]$ and uses it to decide which occurrence of each symbol will remain. The same number is used to select each of the occurrences of all repeated symbols. The rest of the algorithm is the same as in Algorithm A2.

Theorem 2.1 *Algorithms* A2 *and* A3 *are m-approximations for* RFLCS(x, y)*, where m is the maximum of $m_a(x, y)$, over all symbols a.*

Sketch of the proof. Fix x, y, and a repetition-free LCS w of x and y. Sequence w can be thought of as a specific subsequence of x and y. Roughly speaking, each symbol in w has a chance of at least $1/m$ to be picked in the random process of both algorithms. So the expected length of the LCS between x' and y' is at least $1/m$ of $|w|$. □

3 Hardness result

We show that RFLCS is APX-hard, by presenting an L-reduction [5] to RFLCS from a particular APX-complete version of MAX 2-SAT. Our result implies Theorems 1 and 2 of Bonizzoni et al. [3], as there are no "mandatory" symbols.

The problem MAX 2,3-SAT(V, C) consists of, given a set C of 2-clauses over a set V of boolean variables, where each literal may appear in at most 3 clauses in C, finding an assignment for V that maximizes the number of satisfied clauses in C. This variant of MAX 2-SAT is APX-complete [1,5]. Assume that, for any v in V, no clause is of the form $\{v, \overline{v}\}$. Denote by val(MAX 2,3-SAT$(V, C), a)$ the number of clauses in C that are satisfied by an assignment a and let opt(MAX 2,3-SAT$(V, C)) = \max\{$val(MAX 2,3-SAT$(V, C), a) : a$ is an assignment$\}$.

Theorem 3.1 *The problem* RFLCS *is* APX*-complete even when restricted to instances* (x, y) *in which the number of occurrences of every symbol in both* x *and* y *is bounded by two.*

Sketch of the proof. Algorithm A1 implies that this case of RFLCS(x, y) is in APX and the following is an L-reduction (see [5]) from MAX 2,3-SAT to RFLCS.

For an instance (V, C) of MAX 2,3-SAT, we describe an instance (x, y) of RFLCS. Let $V = \{v_1, v_2, \ldots, v_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$. Let $k = 6(n-1)$ and $D = \{d_1, d_2, \ldots, d_k\}$ be such that $D \cap C = \emptyset$. For each literal ℓ, let $s(\ell)$ be a sequence composed by the clauses in which ℓ is present, taken in an arbitrary order. Let $x = s(v_1)s(\overline{v_1})d_1 \cdots d_6 s(v_2)s(\overline{v_2})d_7 \cdots d_{12} s(v_3)s(\overline{v_3}) \cdots d_k s(v_n)s(\overline{v_n})$ and $y = s(\overline{v_1})s(v_1)d_1 \cdots d_6 s(\overline{v_2})s(v_2)d_7 \cdots d_{12} s(v_3)s(\overline{v_3}) \cdots d_k s(\overline{v_n})s(v_n)$.

For each v in V and an assignment a for V, the sequence $s(v)$ contains the clauses of C that would be satisfied if $a(v) = \mathbf{T}$, and $s(\overline{v})$ contains the clauses of C that would be satisfied if $a(v) = \mathbf{F}$. Since there is no clause $\{v, \overline{v}\}$, then $s(v)$ and $s(\overline{v})$ have no common symbol. In addition, as each literal ℓ may appear in at most three clauses of C, we have that $|s(\ell)| \leq 3$. By definition, C and D are disjoint and each symbol of D occurs once in both x and y. In addition, as each clause c in C has two literals, and, for each literal ℓ, the corresponding sequence $s(\ell)$ appears once in either x or y, it follows that each symbol c occurs twice in x and also twice in y. □

4 An IP based exact algorithm for the problem

We show in this section an IP formulation for RFLCS(x, y). For each symbol a, let $E_a = \{(i, j) : x_i = y_j = a\}$ and let $E = \bigcup_a E_a$. The set E_a represents all possible alignments of the symbol a in x and y. We say (i, j) and (k, l) in E *cross* if $i < k$ and $j > l$. For each (i, j) in E, there is a binary variable z_{ij} and linear restrictions on z_{ij} so that $z_{ij} = 1$ if and only if x_i and y_j are aligned in a repetition-free LCS of x and y. The IP formulation is then as follows.

$$\begin{array}{lrcll} \text{maximize} & \sum_{(i,j)\in E} z_{ij} & & & \\ \text{subject to} & \sum_{(i,j)\in E_a} z_{ij} & \leq & 1 & \text{for each symbol } a, \\ & z_{ij} + z_{kl} & \leq & 1 & \text{for each } (i,j),\ (k,l) \in E \text{ that cross,} \\ & z_{ij} & \in & \{0,1\} & \text{for each } (i,j) \text{ in } E. \end{array} \tag{1}$$

Indeed, the first constraint assures that the set $\{i : z_{ij} = 1 \text{ for some } j\}$ defines a repetition-free subsequence w_x of x and $\{j : z_{ij} = 1 \text{ for some } i\}$ defines a repetition-free subsequence w_y of y. The second constraint assures that the order of appearance of the symbols in w_x and w_y is the same, that is, $w_x = w_y$ and therefore we have a common subsequence. The objective function maximizes the length of such a subsequence.

We used this IP formulation to solve some instances of RFLCS, so that we could evaluate empirically our approximation algorithms. Using a general purpose IP solver, we were not able to solve instances of size over 250. However, with a specific branch-and-cut algorithm that we implemented, we could solve most of the instances to optimality.

5 Computational experiments

We tested the three approximation algorithms on two types of randomly generated instances. In the first type, we considered two parameters: the length of the sequences and the alphabet size as a function of the length. Each position of a randomly generated sequence is one of the symbols of the alphabet chosen uniformly at random. In these sequences, most of the symbols have approximately the same number of occurrences.

In the second type, we considered two parameters: the alphabet size and the maximum number of repetitions of each symbol. For each symbol, we pick uniformly at random the number of repetitions of this symbol in the sequence, respecting the given maximum. There is a linear-time (shuffling) procedure that produces, uniformly at random, a sequence with exactly this number of repetitions of each symbol. Note that the expected length of the generated sequence is half of the alphabet size times the maximum number of repetitions.

Owing to space limitation, we do not include the tables with the experimental results, but just comment on them. A first observation is that Algorithm A3 produces the worst results. Also, Algorithm A2 outperforms A1 for small length (under 50) sequences. For larger sequences, in both experiments, Algorithm A1 is the best. We also considered the algorithm that runs A1, A2 and A3 and outputs the best of their solutions. We refer to it as MAX. It is interesting to note that MAX finds optimal solutions more often than A1, which means that A2 and A3 complement sometimes the behavior of A1. In terms of approximation, the ratio between the (average) optimal length and the (average) length of the solution produced by MAX was always no more than 5/4 (for the instances where we had the optimal value).

We observe that instances with alphabet size between $n/4$ and $3n/8$ seem to become harder earlier (for shorter instances) in the sense that the approximation algorithms do not find an optimal solution so often. Indeed, except for these cases, in all other cases, the ratio above was no more than 11/10.

Similar comments hold for the second type instances. For those, the ratio above is also always at most 5/4.

6 Final remarks

Despite of the not so good theoretical worst case ratio, the experimental results indicate that the performance of the approximation algorithms is quite satisfactory for the instance sizes tested. However, it would be nice to test their performance on larger instances. For them, especially when the sequences have many repetitions (small alphabet) we can obtain the solution of the approximation algorithms very fast, but we are not always able to find the optimal value. We are working on the branch and cut algorithm to solve larger instances and hope to confirm the good performance of the approximation algorithms. In any case, it would be interesting to find out whether there is a constant approximation algorithm for RFLCS. For the complete computational results and some proofs, see `http://www.ime.usp.br/~cris/publ/rflcs.pdf`.

Acknowledgements. The authors thank the financial support of FAPESP (Proc. 2003/09925-5, 2004/14335-5), CNPq (Proc. 490333/2004-4, 478329/2004-0), Fundect (Proc. 41/100.149/2006), and Alßan (Proc. E05D053131BR).

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties.* Springer, 1999.

[2] G. Blin, G. Fertin, and C. Chauve. The breakpoint distance for signed sequences. In *Proceedings of CompBioNets - Text in Algorithms*, volume 3, pages 3–16, 2004.

[3] P. Bonizzoni, G. Della Vedova, R. Dondi, G. Fertin, and S. Vialette. Exemplar longest common subsequence. In *Proceedings of ICCS*, volume 3992 of *Lecture Notes in Computer Science*, pages 622–629. Springer, 2006.

[4] D. Bryant. The complexity of calculating exemplar distances. In D. Sankoff and J.H. Nadeau, editors, *Comparative Genomics*, pages 207–212. Kluwer, 2001.

[5] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[6] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):909–917, 1999.

A Quadratic Algorithm for the 2-Cyclic Robotic Scheduling Problem

Pablo Sáez [1]

Computer Science Department
University of Concepción
Chile

Abstract

We improve on an $O(n^5 \log n)$ algorithm by Kats and Levner [3] for 2-cyclic robotic scheduling. We provide in this work an $O(n^2 \log n)$ algorithm for this problem.

Keywords: m-cyclic robotic scheduling, polynomial algorithm, automated manufacturing process optimization.

1 Introduction

The problem considered in this paper stems from the automatized manufacturing industry, where robots handle parts from one machine to another. A manufacturing plan is to be produced that will be repeated over and over by the manufacturing line. In the problem that we are considering, at every time at most two parts go through the production line simultaneously. Thus manufacturing plans can be considered to output exactly two parts from the line. The shorter this plan is, the greater the throughput of the manufacturing line is. Thus the goal in this problem is to minimize the duration of this

[1] Email:psaez@inf.udec.cl

manufacturing plan. A practical problem of this kind arises in an automated electroplating line for processing Printed Circuit Boards (PCBs), but similar problems can be found in many manufacturing settings. We refer to [1] for a general discussion of scheduling problems in a manufacturing context.

We follow the notation from [3]. A sequential manufacturing line is given, which consists of machines $M_1, M_2, ..., M_n$ and parts go through the line in this order (always). All parts are assumed to be of the same kind. A robot does the job of handling parts from one machine to the next. The processing time of a part at machine i is given by p_i and the time needed by the robot to handle a part from machine i to machine $i+1$ is denoted by d_i. An initial stage M_0 and a final stage M_{n+1} are defined, so that the time needed by the robot to input a part into machine M_1 and to output it from machine M_n are considered to be d_0 and d_n respectively.

The *no-wait condition* for this problem states that a part must be unloaded from a machine and handled to the next machine (by the robot) inmediately after being processed by this machine. This condition is implied by the fact that no buffer is available at the machines. In Printed Circuit Boards (PCBs) processing lines for instance a delay in handling a part can deteriorate the quality of the product.

Thus, given a time Z_0 at which a part enters the manufacturing line, it will exit machine M_1 at time $Z_1 = d_0 + p_1$, it will exit machine M_2 at time $Z_2 = d_0 + p_1 + d_1 + p_2$ and in general it will exit machine M_i at a time

$$Z_i = \Sigma_{j=1}^{i}(d_{j-1} + p_j), i = 1...n.$$

That is, the behaviour of any particular part, from a timing perspective, is completely defined and we call this behaviour an *elementary schedule.* For example, the following is an elementary schedule, where the d_i and the p_i are shown on the line.

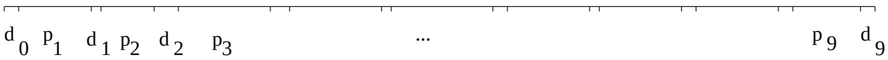


Fig. 1. An elementary schedule.

Thus, if only a single part was allowed to go through the manufacturing line at all times, the general schedule would consist only in a repetition of elementary schedules, over and over, separated by the time the robot needs to return from stage M_{n+1} to stage M_0. But in the 2-cyclic robotic scheduling problem *two* parts may go simultaneously through the manufacturing line, at different machines at the same time (machines are assumed to be able to process only one part at a time). Therefore, the robot needs to permanently go back and forth, handling the parts. The time needed by the robot to go from machine M_i to machine M_j is defined to be r_{ij}.

Given the fact that schedules for single parts are fixed, given by elementary schedules, a schema for a solution to the 2-cyclic robotic scheduling problem can be given by two parallel sequences of elementary schedules, as follows:

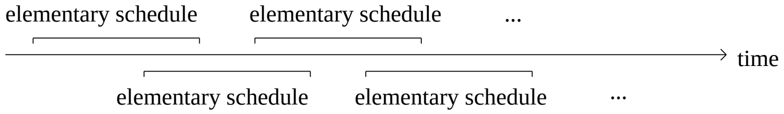


Fig. 2. A general schedule.

We call this schema a *general schedule*. The goal of the problem, as said before, is to tighten as much as possible this general schedule or, equivalently, to maximize the throughput of the whole manufacturing line. Notice that an optimal general schedule is always a repetition of pairs (e_1, e_2) of elementary schedules, whose length is equal to $Z_n + d_n$, where the overlapping between e_1 and e_2 is given by τ_1, the overlapping between e_2 and the next elementary schedule is given by τ_2, both quantities satisfy

$$(1) \quad \tau_1 + \tau_2 + r_{n+1,0} \le Z_n + d_n,$$

and both are constant. In fact, assuming a sequence of elementary schedules that satisfies these conditions, the first appearence of a value $\tau_1' > \tau_1$ for τ_1 implies a non optimal general schedule, since τ_1' could be reduced to τ_1. The same analysis applies for τ_2. For $\tau_1 < \tau_1'$, the analysis can be made the other way around.

Previous studies on this problem can be found in [2,3], but in both cases with $O(n^5 \log n)$ algorithms.

2 Our solution

Since elementary schedules are feasible by themselves, the feasibility of the general schedule is given by the feasibility of the different *overlappings* that arise in time, between elementary schedules. We study the set of all possible (not necessarily feasible) overlappings between two elementary schedules with a graphic as shown in figure 3, next page. We drew the elementary schedule from figure 1 at the top of the triangle and we projected this schedule downwards and diagonally downwards. Therefore any horizontal line across the triangle represents a possible intersection between two elementary schedules, whose length equals the length of the intersection between the horizontal line and the big triangle.

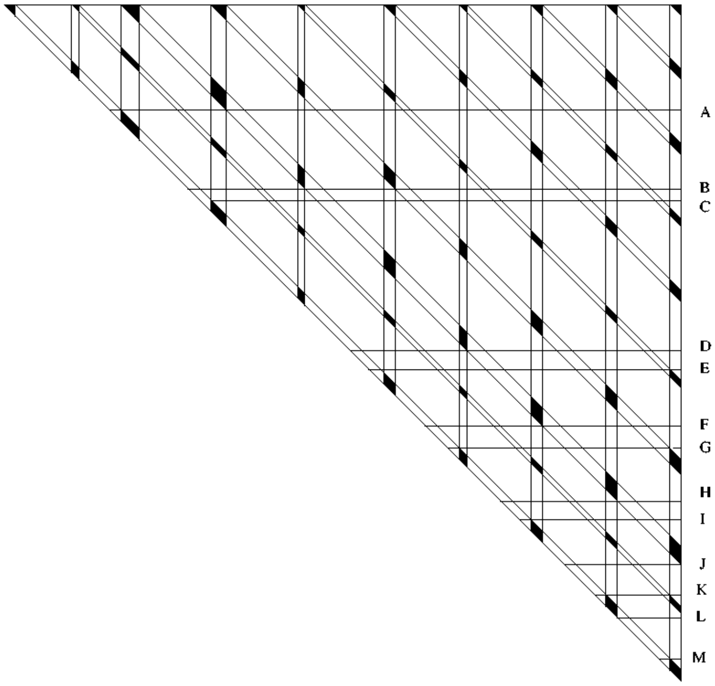


Fig. 3. A schema of all possible elementary schedule intersections.

Clearly a horizontal line that intersects a darkened parallelogram represents an unfeasible ovelapping between two elementary schedules, since this would mean that at a particular moment of time the robot would be performing two handling operations, and this is, by definition of the problem, impossible. This analysis allows us to discard at once in figure 3 all horizontal lines that intersect some darkened parallelogram, that is, everything that lies between lines A and B, or between C and D, or E and F, or G and H, or I and J, or K and L.

Now, with respect to line A, another analysis is needed: for each particular machine M_i, the job consists in: (a) receiving a part from machine M_{i-1}, which takes an amount of time equal to d_{i-1}, (b) processing it, which takes p_i, and (c) letting the part to be handled by the robot to the next machine, and this takes d_i. *Only afterwards* can a new cycle begin, at least with respect to machine M_i. From this observation, we have that no solution above line A is feasible, since A is located at $max_{i=1..n} d_{i-1} + p_i + d_i$.

Given the above considerations, feasible overlappings need to be searched only between B and C, or D and E, and so on.

The last part of the analysis takes into account the r_{ij}, that is, the time the robot needs to move between the machines. We do the analysis based on figure 4. Basically the extra time the robot needs to go from one machine to the other needs to be added to the darkened parallelograms.

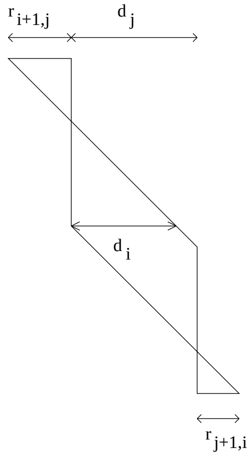


Fig. 4. Enlargened parallelograms.

They need to be enlarged upwards and downwards according to the rule that the parallelogram made from d_i and d_j is enlarged by $r_{i+1,j}$ upwards and by $r_{j+1,i}$ downwards. Actually, they must be enlarged by the minimum between these quantities and the distance that goes to the next darkened parallelogram (going from one darkened parallelogram to the next implies a change in the r_{ij} that needs to be considered).

With everything that has been said until now, we are ready to formulate an algorithm for the considered problem:

(i) Compute $A = max_{i=1...n} d_{i-1} + p_i + d_i$ (this is clearly linear in n in time).

(ii) Generate the set of darkened, enlarged, parallelograms. This set has a quadratic cardinality. Therefore this part of the algorithm is quadratic in time.

(iii) Project this set onto the y-axis, thus producing a set of unfeasible intervals for the ovelapping. Add to this set everything that is above A. We have got all unfeasible overlappings between two elementary schedules.

(iv) Sort this set using an $O(m \log m)$ algorithm for that purpose with $m = n^2$, this step takes $O(n^2 \log n^2) = O(2n^2 \log n) = O(n^2 \log n)$ in time. From this ordered set of intervals compute the set that is equal to its *negation*, by going through it sequentially. Clearly the resulting set is also (at most) quadratic in cardinality and represents the set of feasible overlappings. Call it I.

(v) Find $max_{\tau_1 \in [a_1,b_1],[a_1,b_1] \in I, \tau_2 \in [a_2,b_2],[a_2,b_2] \in I} \tau_1 + \tau_2$ subject to the restriction: $\tau_1 + \tau_2 + r_{n+1,0} \leq Z_n + d_n$ (restriction (1)).

But this last problem is also $O(n^2 \log n)$. For instance the following procedure can be used:

(i) For each interval $[a, b] \in I$:
   (a) Search using a binary search, an interval in I that contains any element in the interval $[Z_n + d_n - r_{n+1,0} - b, Z_n + d_n - r_{n+1,0} - a]$. If such an interval is found then the best possible solution to the problem has been found (exit the loop).
   (b) Otherwise find, again using a binary search, the interval $[c, d] \in I$ with $c > a$ closest to $Z_n + d_n - r_{n+1,0} - a$ and output $(c + a, a, b)$.

The maximum output for $c + a$ is also the optimum for this problem, where the optimal values for τ_1, τ_2 are precisely a and c. In fact, this pair (a, c) maximizes $\tau_1 + \tau_2$, and $c - (Z_n + d_n - r_{n+1,0} - a)$ being positive implies a feasible solution.

3 Conclusions

We presented in this paper a solution to the 2-cyclic robotic schedule optimization problem that runs in $O(n^2 \log n)$ time. This improves previous solutions to the problem, which are $O(n^5 \log n)$. We conjecture that this method can be generalized to $O(n^m \log n)$ algorithms for the $m-$cyclic problem, that is, the same problem where m parts go simultaneously through the production line.

References

[1] Blazewicz, K. H., Ecker, K.H., Pesch, E., Schmidt, G., and G. Schmidt "Scheduling Computer and Manufacturing Processes", Springer Verlag, 1996.

[2] Chu, C., *A Faster Polynomial Algorithm for 2-Cyclic Robotic Scheduling.* Journal of Scheduling, 2006.

[3] Kats, V., and E. Levner, *A Polynomial Algorithm for 2-Cyclic Robotic Scheduling*, In Gelbukh and Reyes-García eds, Proceedings of MICAI'06, pp. 439–449, LNAI **4293**. Springer Verlag, 2006.

On the complexity of feedback set problems in signed digraphs

Marco Montalva[3], Julio Aracena[1,4] and Anahí Gajardo[2,5]

Departamento de Ingeniería Matemática
Universidad de Concepción
Concepción, Chile

Abstract

Given a directed graph $G = (V, E)$ and $w : E \to \{-1, +1\}$ a sign function on the arcs of G, we study the positive feedback vertex set problem (PFVS) which consists on finding a minimum cardinality set of vertices that meets all the cycles with an even number of negative arcs. This problem is closely related with the number of steady states of Regulatory Boolean Networks. We also study the negative feedback vertex set problem which consists on finding a minimum cardinality set of vertices that meets all the cycles with an odd number of negative arcs, and the analogous problems for arc sets. We prove that all of these problems are NP-complete.

Keywords: Feedback Set problems, NP-complete, signed digraph, Regulatory Boolean Networks.

[1] This work was partially supported by FONDECYT #1061008.

[2] This work was partially supported by FONDECYT #1030706.

[3] Email:`marco@ing-mat.udec.cl`

[4] Email:`jaracena@ing-mat.udec.cl`

[5] Email:`anahi@ing-mat.udec.cl`

1 Introduction

The problem of finding a minimum cardinality set of vertices or arcs that meets all the cycles of a graph (directed or undirected) is known as Feedback Set problem. These problems are known to be NP-complete and appear in Karp's seminal paper [5]. There are many variants of this classical problem, some of them consider weights on the vertices or on the arcs. Almost all of them have been proved to be NP-complete (see [3] for a survey). In this paper, we study the complexity of new variants: the Positive and Negative Feedback Set problems.

Let G be a digraph with signs -1 or $+1$ on the arcs. A cycle of G is said to be positive (resp. negative) if it has an even (resp. odd) number of negative arcs. The Positive Feedback Vertex Set problem (PFVS) and Negative Feedback Vertex Set problem (NFVS) consist in finding a minimum cardinality set of vertices that meets all the positive cycles and all the negative cycles, respectively. Positive Feedback Arc Set problem (PFAS) and the Negative Feedback Arc Set problem (NFAS) are similarly defined.

Feedback problems are fundamental in combinatorial optimization, having many applications: circuit design, certain scheduling problems and cryptography are some examples. For this reason, they have been extensively studied [3]. PFVS was introduced in [1], where it was proved that the maximum number of steady states of a Regulatory Boolean Network (RBN) depends on the cardinality of the minimum vertex set that meets all the positive cycles of the connection digraph. The RBNs has been used to model regulatory biological systems like genetic regulatory networks [1,2], where the steady states are associated to different cellular phenotypes. In this way, the solution of PFVS would ease the construction of genetic regulatory networks, which has become one of the major problems in Biology [6].

In this paper, we prove that PFVS and NFVS are both NP-complete by the construction of a polynomial reduction from FVS to PFVS and NFVS, respectively. We use these results to prove that PFAS and NFAS are also NP-complete. However, the positive version of feedback problems appear to be more difficult to solve than the negative one. In fact, the computational complexity of the problem of determining whether a given signed digraph has an even length cycle (which is equivalent to ask whether the digraph has a positive cycle, as we will show) remained unknown for several decades. In 1989 Vazirani and Yannakakis [9] proved that Even Cycle is polynomially equivalent to the problem of testing if a given bipartite graph has a Pfaffian orientation; which was only proved to be polynomial in 1999 by Robertson, Seymour and

Thomas [8]. On the contrary, to determine whether a given signed digraph has a negative cycle is easy to solve. This shows the difficulty of finding explicit polynomial reductions between the positive and negative variants of feedback set problems.

2 Definitions and notation

Let $G = (V, E)$ be a digraph where $V = V(G)$ and $E = E(G) \subseteq V \times V$ are the vertex and arc set respectively. Let $w : E \longrightarrow \{-1, +1\}$ be a sign function on the arcs of G. The couple (G, w) is called *signed digraph.* A *cycle* in G is a sequence of vertices $v_0, v_1, \ldots, v_k$ in G that for each $0 \leq i \leq k-1$, $(v_i, v_{i+1}) \in E$, all the vertices are distinct except $v_0 = v_k$. A cycle C is called *positive* (resp. *negative*) if the number of negative arcs in C is even (resp. odd). A vertex (resp. arc) set $U \subseteq V$ (resp. $A \subseteq E$) is a *positive feedback vertex (resp. arc) set* if $G - U$ (resp. $G - A$) has no positive cycles. *Negative feedback vertex (resp. arc) set* is similarly defined. Thus, we define the following decision problems:

PFVS (resp. NFVS). Given a signed digraph $(G = (V, E), w)$ and $k \in \mathbb{N}$. Does a positive (resp. negative) feedback vertex set U exists such that $|U| \leq k$?.

PFAS (resp. NFAS). Given a signed digraph $(G = (V, E), w)$ and $k \in \mathbb{N}$. Does a positive (resp. negative) feedback arc set A exists such that $|A| \leq k$?.

Positive Cycle (resp. Negative). Given a signed digraph (G, w), determine if there exists a positive (resp. negative) cycle in G.

3 NP-Completeness results

Given a digraph $G = (V, E)$ and a vertex (resp. arc) set $U \subseteq V$ (resp. $W \subseteq E$) we can verify if U (resp. W) is a positive feedback vertex (resp. arc) set by testing whether $G - U$ (resp. $G - W$) has or not positive cycles. Then if Positive Cycle is polynomial, PFVS and PFAS are in NP.

Proposition 3.1 *Positive Cycle is polynomially equivalent to Even Cycle.*

Proof. In fact, let us define the reduction function as follows: given a signed digraph $(G = (V, E), w)$, we define $\psi(G, w) = \widetilde{G}$, where $\widetilde{G}$ is a digraph in which each positive arc of G is replaced by a two length path. In this way, a given cycle C of G is transformed into a longer cycle $\widetilde{C}$ of $\widetilde{G}$. The length of $\widetilde{C}$

is equal to the number of negative arcs of C plus twice the number of positive arcs of C. Then, if C is positive, $\widetilde{C}$ has even length.

On the other hand, a cycle $\widetilde{C}$ of $\widetilde{G}$ corresponds to a unique cycle C of G and $\widetilde{C}$ has even length if and only if C is positive. In this way Positive Cycle polinomially reduces to Even Cycle. The converse reduction is straightforward. □

Theorem 3.2 *PFVS and NFVS are NP-complete.*

Proof. It is easy to see that FVS polynomially reduces to PFVS; it is enough to define w as a constant sign function that assigns $+1$ to every arc. Let us prove that FVS polynomially reduces to NFVS. Given a digraph G, we define the signed digraph $\phi(G) = (\widetilde{G}, w)$ where $\widetilde{G}$ is a digraph in which we added, for each arc of G, a two length path of negative sign, and we assigned $+1$ to all the original arcs (see Figure 1). In this way, the digraph $\widetilde{G}$ has both a positive and a negative cycle for each cycle of G, and every cycle of $\widetilde{G}$ corresponds to a unique cycle of G.

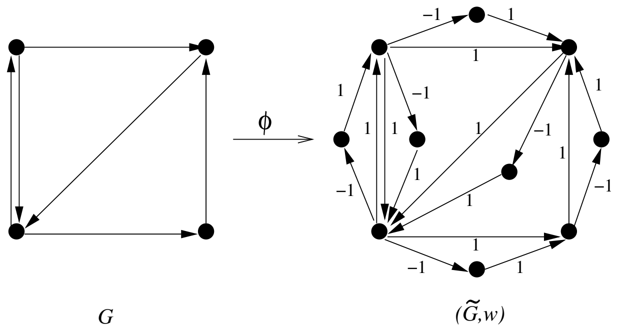


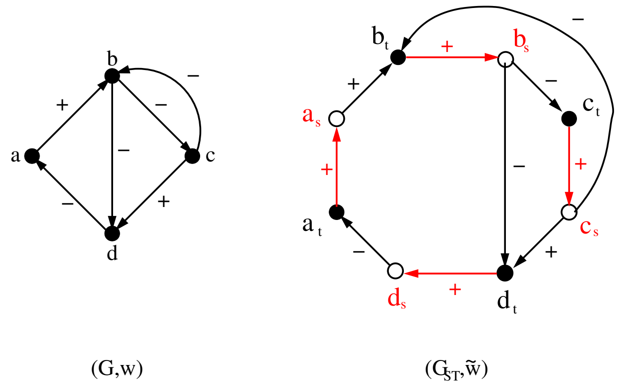
Fig. 1. For each cycle of G, the signed digraph $(\widetilde{G}, w)$ has both a negative and a positive cycle.

If we have a negative feedback vertex set for $\phi(G)$, it can be composed by vertices from G and some new vertices of $\widetilde{G}$. But each new vertex can be replaced by its unique incident vertex, which lies in the original vertex set, obtaining, in this way, a set of the same or smaller size which is a feedback vertex set of G. Conversely, if U meets the cycles of G, then it also meets the negative cycles of $\phi(G)$, which ends the proof. □

Theorem 3.3 *PFAS and NFAS are NP-complete.*

Proof. It is enough to show that PFVS and NFVS polynomially reduces to PFAS and NFAS respectively. Let us define the following reduction function: given a signed digraph $(G = (V, E), w)$, we define $\theta(G, w) = (G_{ST}, \tilde{w})$, where

G_{ST} is as follows: for each vertex $v \in V$, G_{ST} has two new vertices v_s, v_t and a positive arc (v_t, v_s). For each arc $(x, y) \in E(G)$, G_{ST} has the arc (x_s, y_t) with the same sign that the arc (x, y) (see Figure 2). In this way, there is a one to one relation between positive (resp. negative) cycles of G and positive (resp. negative) cycles of G_{ST}. On the one hand, if $S \subseteq V$ meets the positive (resp. negative) cycles of (G, w), then $\tilde{S} = \{(v_{i_t}, v_{i_s}) \in G_{ST} : v_i \in S\}$ meets the positive (resp. negative) cycles of $\theta(G, w)$. On the other, if we have a positive (resp. negative) feedback arc set for $\theta(G, w)$, it can be composed by arcs of the form (x_t, x_s) or (x_s, y_t). In the first case, they correspond to vertices in G; in the second case, all the positive (resp. negative) cycles that contain it, also contain the arc (x_t, x_s). Then we can change each arc of the form (x_s, y_t) by (x_t, x_s). The vertices x in G asociated with these arcs constitute a positive (resp. negative) feedback vertex set for G. We have simultaneously proved that the function θ is a polynomial reduction from PFVS to PFAS and from NFVS to NFAS. □



(G,w)

$(G_{ST}, \tilde{w})$

Fig. 2. For each positive (resp. negative) cycle of (G, w), the signed digraph $(G_{ST}, \tilde{w})$ has a positive (resp. negative) cycle and viceversa.

4 Concluding remarks

We have shown that Positive Cycle is polynomially equivalent to Even Cycle. In a similar way we can prove that Negative Cycle is polynomially equivalent to Odd Cycle [7]. In the case of undirected graphs, these equivalences are still valid, and Even and Odd Cycle are proved to be polinomial too [4]. We can also prove that FVS reduces to PFVS; then PFVS is also NP-complete for this case.

We can also consider graphs with signs on the vertices. In this context, the corresponding PFVS and NFVS problems are polynomially equivalent to the problems studied in this paper.

Regarding the solution of these problems we developed a simple greedy algorithm to solve PFVS. Of course, it is exponential in the worst case; but, we test it with small signed digraphs and it worked fast. The algorithm is based on the heuristic that the higher the degree of a vertex, the higher the probability of participating in an optimal PFVS. This is not always true, but it works in most of cases. Are the regulatory boolean networks simple instances for PFVS? Can we define a family of graphs (characterizing the topology of RBNs) for which PFVS is polinomial?

References

[1] Aracena, J., "Modelos matemáticos discretos asociados a los sistemas biológicos. Aplicación a las redes de regulación génica," Ph.D. thesis, University of Chile & UJF, Santiago, Chile, & Grenoble, France, 2001.

[2] Aracena, J., *On the number of fixed points in regulatory boolean networks*, Bulletin of Mathematical Biology, accepted.

[3] Festa, P., P. M. Pardalos, and M. G. Resende, "Feedback set problems, in: Handbook of Combinatorial Optimization," Supplement Vol. A, Kluwer Academic Publishers, 1999.

[4] M. R. Garey, M. R., and D. S. Johnson, "Computers and intractability: A guide to the theory of NP-Completeness," W.H. Freeman and Company, New York, 1979.

[5] Karp, R. M., "Reducibility among Combinatorial Problems, in: Complexity of Computer Computations," R. E. Miller and J. W. Thatcher, Eds., Plenum Press, New York, 1972.

[6] Kauffman, S.A., "The Origins of Order, Self-Organization and Selection in Evolution," Oxford University Press, 1993.

[7] Montalva, M., "Conjunto de vértices recubridor de los ciclos positivos en un digrafo," Engineering thesis, University of Concepción, Concepción, Chile, 2006.

[8] Robertson, N., P. Seymour, and R. Thomas, *Permanents, pfaffian orientations, and even directed circuits*, Annals of Mathematics **150** (1999), 929–975.

[9] Vazirani, V., and M. Yannakakis, *Pfaffian orientations, 0-1 permanents, and even cycles in directed graphs*, Discrete Applied Mathematics **25** (1989), 179–190.

Flow Hypergraph Reducibility

A. L. P. Guedes [a,1] L. Markenzon [b,2] and L. Faria [c]

[a] *Universidade Federal do Paraná, PR, Brazil*

[b] *Universidade Federal do Rio de Janeiro, RJ, Brazil*

[c] *Universidade do Estado do Rio de Janeiro, RJ, Brazil*

Abstract

Reducible flow graphs were first defined by Hecht and Ullman in terms of intervals; another definition, based on two flow graph transformations, was also presented. In this paper, we extend the notion of reducibility to directed hypergraphs, proving that the interval and the transformation approaches are still equivalent when applied to this family.

1 Introduction

Reducible flow graphs were introduced by [4,5] to model the control flow of computer programs. Although they were initially used in code optimization algorithms, several theoretical and applied problems have been solved for that class [9,11].

The first definition of reducible flow graphs is in terms of intervals [4]. The same authors presented, in [5] another definition, this time based on two transformations over the flow graph. Tarjan [10] used the second one in order to present an efficient algorithm for testing whether a flow graph is reducible.

[1] Partially supported by CNPq 479882/2004-5

[2] Partially supported by CNPq 206893/2006-1

Directed hypergraphs [1] are a generalization of digraphs and they can model binary relations among subsets of a given set. Such relationships appears in different areas such as expert systems [8], parallel programming [7] and scheduling [6]. Structural properties of directed hypergraphs such as planarity [3] and coloring [12] were also studied.

In this paper, we define flow hypergraphs and we extend the notion of reducibility to this family. We define reducible flow hypergraphs by intervals and we introduce two transformations, proving the equivalence of the definitions.

2 Basic Notions - Directed Hypergraphs

Basic concepts about directed hypergraphs were introduced by Gallo et al. [1] and are reproduced here.

Definition 2.1 A **directed hypergraph** $H = (V, A)$ is a pair, where V is a non empty finite set of vertices and A is a collection of hyper-arcs. A **hyper-arc** $a = (X, Y) \in A$ is an ordered pair where X and Y are non empty subsets of V. Set $X = Org(a)$ is called the **origin** and set $Y = Dest(a)$ the **destination** of a.

The notation Org and $Dest$ can be extended to a collection A' of hyper-arcs. So, $Org(A') = \cup_{e \in A'} Org(e)$ and $Dest(A') = \cup_{e \in A'} Dest(e)$.

Definition 2.2 Let $H = (V, A)$ be a directed hypergraph and $v \in V$. We denote:

$BS(v) = \{e \in A \mid v \in Dest(e)\}$, the **backward star** set of v.

$FS(v) = \{e \in A \mid v \in Org(e)\}$, the **forward star** set of v.

Definition 2.3 Let $H = (V, A)$ be a directed hypergraph and u and v be vertices of H. A **B-path** of size k from u to v is a sequence of hyper-arcs $P = (e_{i_1}, e_{i_2}, e_{i_3}, \ldots, e_{i_k})$, such that for each hyper-arc e_{i_p} of P, $1 \le p \le k$, we have:

$Org(e_{i_p}) \subseteq (Dest(e_{i_1}, e_{i_2}, \ldots, e_{i_{p-1}}) \cup \{u\}$

$Dest(e_{i_p}) \cap (Org(e_{i_{p+1}}, e_{i_{p+2}}, \ldots, e_{i_k}) \cup \{v\}) \neq \emptyset.$

$Org(e_{i_1})$ and $Dest(e_{i_k})$ can be denoted by $Org(P)$ and $Dest(P)$, respectively.

Based on the known concept of flow graphs and Def. 2.1 we can define flow hypergraphs.

Definition 2.4 A **flow hypergraph** $H = (V, A, s)$ is a triple, where (V, A) is a directed hypergraph, $s \in V$ is a distinguished source vertex, and there is a B-path from s to every other vertex in V.

Figure 1 shows an example of a flow hypergraph. Note that the sequence (a, c, d) is a B-path from vertex 1 to vertex 6, but the sequence (a, b, g) is not a B-path from vertex 1 to vertex 9.

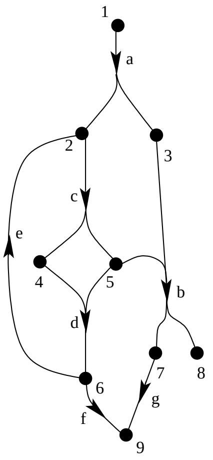


Fig. 1. An example of a flow hypergraph

3 Reducibility of Flow Hypergraphs

In this section, the notion of reducibility of flow hypergraphs is introduced, based on the extension of the two already known approaches. We also show their equivalence.

3.1 Interval Approach

Let $H = (V, A, s)$ be a flow hypergraph. The **interval** $I(v)$ with header v is the maximum subset of V recursively defined as follows:

(i) $v \in I(v)$

(ii) Let $a = (X, Y)$ be a hyper-arc such that $\forall y \in Y - \{v\}$, $Org(BS(y)) \subseteq I(v)$. If $v = s$ or $s \notin Y$ then $Y \subseteq I(v)$.

Let $H = (V, A, s)$ be a flow hypergraph, $I(s_1), I(s_2), \ldots, I(s_k)$ its maximal intervals. The flow hypergraph $I(H) = (V', A', s)$, called **interval hypergraph**, is obtained by contracting each maximal interval of H by their headers. The source vertex s is the same, since s is the header of some interval.

This operation can be applied several times, generating flow hypergraphs H, $I(H)$, $I^2(H)$, …, $I^p(H)$, being p such that $I^{p+1}(H) = I^p(H)$. This final hypergraph is denoted $I^*(H)$. If $I^*(H)$ is a trivial hypergraph (it has only one vertex and no hyper-arcs), then H is called **reducible by intervals**.

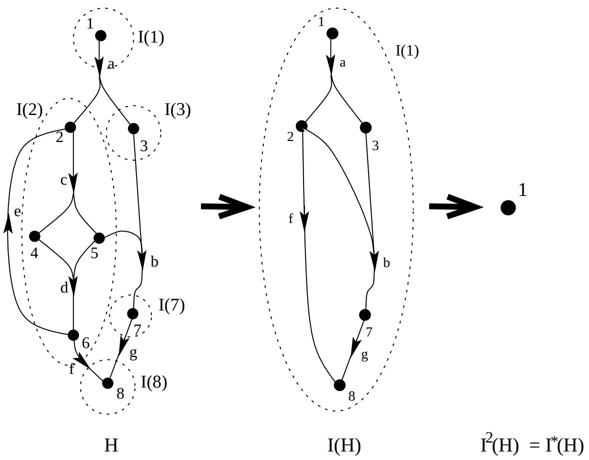


Fig. 2. Sequence of interval hypergraphs

Figure 2 shows the sequence $H, I(H)$, $I^2(H)$. Hypergraph H is reducible by intervals, since $I^2(H)$ is a trivial hypergraph.

3.2 Transformation Approach

Given a flow hypergraph H, two transformations, T_1 and T_2, can be defined on H. These transformations perform the contraction of a hyper-arc.

Let $H = (V, E, s)$ be a flow hypergraph and $a = (\{x\}, \{x\}) \in E$ be a simple loop. Transformation $\boldsymbol{T_1}$ applied to a removes the hyper-arc a from H, resulting in the flow hypergraph $H - a$.

Let $H = (V, E, s)$ be a flow hypergraph and $a = (\{x\}, Y) \in E$ be a hyper-arc with $|Org(a)| = 1$, such that $\forall y \in Y - \{x\}$, $Org(BS(y)) = \{x\}$; $x = s$ or $s \notin Y$; and a is not a simple loop. Transformation $\boldsymbol{T_2}$ applied to a removes from H the hyper-arc a and identifies vertices of Y with x.

There is a unique flow hypergraph $T^*(H)$ given by any sequence of applications of T_1 and T_2 in H in which T_1 and T_2 can not be applied. If $T^*(H)$ is a flow hypergraph with just one vertex and no hyper-arcs, then H is called **reducible by transformations**. See in Figure 3 an example of application of this transformations.

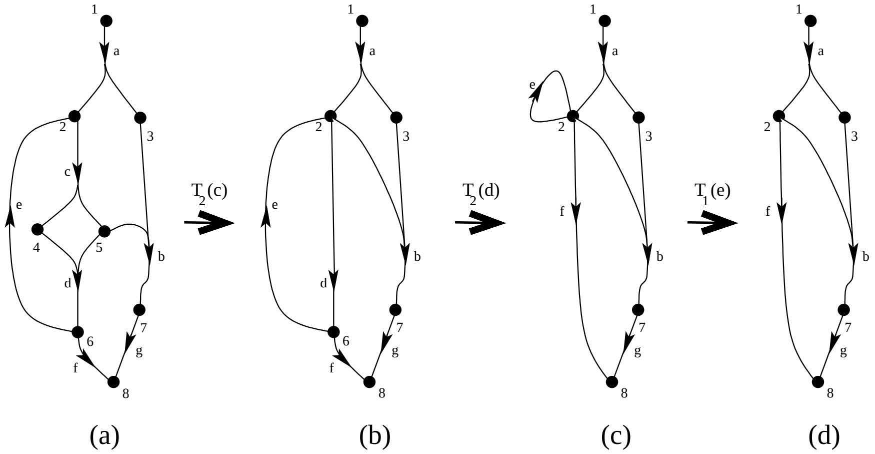


Fig. 3. Example of T_1 and T_2 applications in a flow hypergraph. First (a) T_2 is applied to hyper-arc c, then (b) T_2 in hyper-arc d and (c) T_1 in the resulting loop e

3.3 Equivalence

Hecht and Ullman proved that for reducible flow graphs both approaches are equivalent. We extend this result to flow hypergraphs, as stated in Theorem 3.1.

Theorem 3.1 *A flow hypergraph is reducible by intervals if and only if it is reducible by transformations.*

The operation of contracting an interval $I(v)$ used to construct $I(H)$ can be performed by applications of the transformations T_1 and T_2. There is at least one hyper-arc in $I(v)$, $a = (\{v\}, Y)$, such that T_1 or T_2 can be applied, and by induction all the hyper-arcs are removed.

On the other hand, any sequence of applications of T_1 and T_2 can be reordered resulting in the same hypergraph. So, the sequence of applications of T_1 and T_2 that transforms H into $T^*(H)$ can be reordered to contract the intervals.

Based on Theorem 3.1, we call a flow hypergraph which is reducible by intervals or by transformations simply a **reducible flow hypergraph**.

4 Conclusion

Focusing on the concept of reducibility we are currently working on the identification of reducible flow hypergraph classes, and on the the establishment of the complexity status for the reducible flow hypergraph recognition problem.

References

[1] G. Gallo, G. Longo, S. Nguyen, and S. Pallottino. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42:177–201, 1993.

[2] A. L. P. Guedes. *Hipergrafos Direcionados*. D.Sc. thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, Aug. 2001.

[3] A. L. P. Guedes and L. Markenzon. Directed hypergraph planarity. *Pesquisa Operacional*, 25(3):383–390, Set./Dec. 2005.

[4] M. S. Hecht and J. D. Ullman. Flow graph reducibility. *SIAM Journal of Computing*, 2(2):188–202, Jun. 1972.

[5] M. S. Hecht and J. D. Ullman. Characterizations of reducible flow graphs. *Journal of the Association for Computing Machinery*, 21(3):367–375, Jul. 1974.

[6] W.-L. Lin and M. Sarrafzadeh. A linear arrangement problem with applications. In *ISCAS'95 - Int. Symp. on Circuits and Systems*, vol. 1, pp 67–60. IEEE, 1995.

[7] S. Nguyen, D. Pretolani, and L. Markenzon. On some path problems on oriented hypergraphs. *RAIRO - Informatique Theorique et Applications (Theoretical Informatics and Applications)*, 32(1):1–20, 1998.

[8] M. Ramaswamy, S. Sarkar, and Y.-S. Chen. Using directed hypergraphs to verify rule-based expert systems. *IEEE Trans. on Knowledge and Data Eng.*, 9(2):221–237, Mar./Apr. 1997.

[9] A. Shamir. A linear time algorithm for finding minimum cutsets in reducible graphs. *SIAM J. Comput.*, 8(4):645–655, 1979.

[10] R. Tarjan. Testing flow graph reducibility. In *STOC '73: Proc. of the 5th Annual ACM Symp. on Theory of Computing*, pp 96–107, NY, 1973.

[11] O. Vernet and L. Markenzon. Solving problems for maximal reducible flowgraphs. *Discrete Applied Mathematics*, 136(2–3):341–348, 2004.

[12] A. Vietri. The complexity of arc-colorings for directed hypergraphs. *Discrete Applied Mathematics*, 143(1-3):266–271, 2004.

The maximum number of halving lines and the rectilinear crossing number of K_n for $n \leq 27$

Bernardo M. Ábrego
Silvia Fernández–Merchant

Department of Mathematics, California State University at Northridge

Jesús Leaños
Gelasio Salazar

Instituto de Física, Universidad Autónoma de San Luis Potosí, México

Abstract

For $n \leq 27$ we present exact values for the maximum number $h(n)$ of halving lines and $\widetilde{h}(n)$ of halving pseudolines, determined by n points in the plane. For this range of values of n we also present exact values of the rectilinear $\overline{\mathrm{cr}}(K_n)$ and the pseudolinear $\widetilde{\mathrm{cr}}(K_n)$ crossing numbers of the complete graph K_n. $\widetilde{h}(n)$ and $\widetilde{\mathrm{cr}}(K_n)$ are new for $n \in \{14, 16, 18, 20, 22, 23, 24, 25, 26, 27\}$, $h(n)$ is new for $n \in \{16, 18, 20, 22, 23, 24, 25, 26, 27\}$, and $\overline{\mathrm{cr}}(K_n)$ is new for $n \in \{20, 22, 23, 24, 25, 26, 27\}$.

Keywords: Halving lines, rectilinear crossing number, complete graphs

1 Introduction

Let S be an n-point set in $\mathbb{R}^2$ in general position. A *k-set* of S is a set P of k points in S that can be separated from $S \backslash P$ using a straight line. The so called "k-set problem" asks for the maximum number of k-sets that an

n-element set can have. In a similar fashion we say that a directed segment $\overrightarrow{s_1s_2}$ in S is a *k-edge* if there are exactly k points in S to the right side of s_1s_2. It is easy to see that there is a 1-to-1 correspondence between k-sets and $(k-1)$-edges, so an equivalent problem is to find the maximum number of $(k-1)$-edges determined by n points in the plane. When n is even and $k=(n-2)/2$ the k-edges are called *halving lines,* that is, lines through two points in S leaving $(n-2)/2$ points of S on each side. When n is odd the halving lines leave $(n-3)/2$ and $(n-1)/2$ points of S on each side.

An important open problem in discrete geometry is to find the maximum number $h(n)$ of halving lines that can be determined by n points in the plane. This was first raised by Erdős, Lovász, Simmons, and Straus [11], [14]. Another important and related problem was proposed by Erdős and Guy: find the minimum number of convex quadrilaterals in a set of n points in general position. Equivalently, determine $\overline{\text{cr}}(K_n)$, the *rectilinear crossing number* of K_n [10], that is, the smallest number of crossings in a drawing of the complete graph K_n, in which every edge is drawn as a straight segment. Further references and related problems can be found in [8].

All these problems can be formulated in the more general setting of generalized configurations of points [13]. A *generalized configuration* or a *pseudoconfiguration* consists of a set of points in the plane together with an *arrangement of pseudolines*, such that every pair of points has exactly one pseudoline passing through them. A *pseudoline* is a curve in $\mathbb{P}^2$, the projective plane, whose removal does not disconnect $\mathbb{P}^2$. An *arrangement of pseudolines* is a collection of pseudolines with the property that every two of them intersect each other exactly once. In this new setting we can define by analogy *k-pseudoedges*, *halving pseudolines*, and *pseudolinear crossing number* $\widetilde{\text{cr}}(K_n)$ of K_n. We denote by $\widetilde{h}(n)$ the maximum number of halving pseudolines spanned by generalized configurations of n points in the plane. We also let $\mathcal{N}_k(n)$ and $\mathcal{N}_{\leq k}(n)$ denote the maximum number of $(k-1)$-pseudoedges, $\leq (k-1)$-pseudoedges respectively, determined by pseudoconfigurations of n points. Trivially, $\widetilde{\text{cr}}(K_n) \leq \overline{\text{cr}}(K_n)$ and $h(n) \leq \widetilde{h}(n)$.

Here we report improved lower bounds for $\widetilde{h}(n)$. This improvement is enough to match the geometric constructions that serve as upper bounds in the range $n \in \{14, 16, 18, 20, 22, 23, 24, 25, 26, 27\}$. We also obtain new lower bounds for $\mathcal{N}_{\leq \lfloor n/2 \rfloor -1}(n)$. As a consequence we determine the exact values of $\widetilde{\text{cr}}(K_n), \overline{\text{cr}}(K_n)$, $\widetilde{h}(n)$, and $h(n)$ for the same range. The new values are summarized in Table 1. It is important to note that all of these bounds are shown to be tight thanks to the remarkable (indeed, as we show, optimal) geometric constructions obtained by Aichholzer et al. [4].

n	14	16	18	20	22	23	24	25	26	27
$h(n) = \widetilde{h}(n)$	22	27	33	38	44	75	51	85	57	96
$\mathcal{N}_{\leq \lfloor n/2 \rfloor - 1}(n)$	69	93	120	152	187	178	225	215	268	255
$\overline{cr}(n) = \widetilde{cr}(n)$	324	603	1029	1657	2528	3077	3699	4430	5250	6180

Table 1
New exact values for $h(n), \widetilde{h}(n), \overline{\mathrm{cr}}(K_n)$, and $\widetilde{\mathrm{cr}}(K_n)$.

Here is the previous history about the quest for (small values of) $h(n)$. For $2 \leq n \leq 8$ $h(n)$ is easily obtained, and since all generalized configurations of points with $n \leq 8$ are stretchable [12], then $h(n) = \widetilde{h}(n)$ in this range. Eppstein [9] found point sets with even $10 \leq n \leq 18$ and a large number of halving lines. In particular he matched the upper bound found by Stöckl [16] for $\widetilde{h}(10)$. Andrzejak et al. [6] proved $h(12) = 18$. Later Beygelzimer and Radziszowski [7] extended this to $\widetilde{h}(12) = 18$ and they also proved that $h(14) = 22$. With respect to the odd values, Aichholzer et al. [5] found tight upper bounds for $h(n)$ with n odd, $11 \leq n \leq 21$.

Previous to this work, the exact value of $\overline{\mathrm{cr}}(K_n)$ was known for $n \leq 19$ and for $n = 21$ ([5]). For these values of n, it was recently proved that $\widetilde{\mathrm{cr}}(K_n) = \overline{\mathrm{cr}}(K_n)$ [1]. For general lower and upper bounds see [5], [1], and [3].

2 The Central Bound

In what follows Π denotes a circular sequence on n elements, that is, a doubly infinite sequence $(..., \pi_{-1}, \pi_0, \pi_1, ...)$ of permutations on n elements, such that any two consecutive permutations π_i and π_{i+1} differ by a transposition τ_i of neighboring elements, and such that for every j, π_j is the reversed permutation of $\pi_{j+\binom{n}{2}}$. Goodman and Pollack [13] established a one-to-one correspondence between circular sequences and generalized configurations of points. Thus we say that a circular sequence Π is *associated* to a set of n points S. When Π corresponds to a geometric drawing of K_n (i.e., each pseudoline is a straight line) we say that Π is *stretchable*. In this case S is a set of n points in general position in the plane. Any subsequence of Π consisting of $\binom{n}{2}$ consecutive permutations is an *n-halfperiod*. If τ_j occurs between elements in positions i and $i+1$ we say that τ_j is an *i-transposition*. If $i \leq n/2$ then any i-transposition or $(n-i)$-transposition is called *i-critical*. If Π is a finite subsequence of Π

then $\mathcal{N}_k(\Pi)$ and $\mathcal{N}_{\leq k}(\Pi)$ denote the number of k-critical and $(\leq k)$-critical transpositions in Π respectively. A k-transposition corresponds to a $(k-1)$-pseudoedge which also coincides with a $(k-1)$-edge if Π is stretchable.

We make use of two known results (A in [2] and [15], B in [5] and [1]):

$$\text{(A) } \widetilde{\mathrm{cr}}(\Pi) = \sum_{k=1}^{\lfloor n/2 \rfloor} (n-2k-1)\mathcal{N}_k(\Pi) - (3/4)\binom{n}{3} + (1/8)(1+(-1)^{n+1})\binom{n}{2}.$$

$$\text{(B) } \mathcal{N}_{\leq k}(\Pi) \geq 3\binom{k+1}{2} + 3\binom{k+1-\lfloor n/3 \rfloor}{2} - \max\{0, (k - \lfloor n/3 \rfloor)(n - 3\lfloor n/3 \rfloor)\}.$$

Our main new tool is the following.

Theorem 2.1 *Let Π be a circular sequence associated to a generalized configuration of n points. Then*

$$\mathcal{N}_{\lfloor n/2 \rfloor}(\Pi) \leq \begin{cases} \left\lfloor \frac{1}{2}\binom{n}{2} - \frac{1}{2}\mathcal{N}_{\leq \lfloor n/2 \rfloor - 2}(\Pi) \right\rfloor, \text{if } \ n \text{ is even,} \\ \left\lfloor \frac{2}{3}\binom{n}{2} - \frac{2}{3}\mathcal{N}_{\leq \lfloor n/2 \rfloor - 2}(\Pi) + \frac{1}{3} \right\rfloor, \text{if } n \text{ is odd.} \end{cases}$$

Proof. For even n we prove that there must be at least one $(n/2-1)$-critical transposition between any two consecutive $n/2$-transpositions τ_i and τ_j $(i<j)$. Suppose τ_i transposes a and b. Then before τ_j takes place, at least one element a or b must leave the center (two middle positions, $n/2$ and $n/2+1$). This corresponds to having at least one $(n/2-1)$-critical transposition between τ_i and τ_j. In a given halfperiod the same holds for the last and first $n/2$-transpositions. Thus $\mathcal{N}_{n/2}(\Pi) \leq \mathcal{N}_{n/2-1}(\Pi)$. Since $\mathcal{N}_{\leq \lfloor n/2 \rfloor}(\Pi) = \binom{n}{2}$ then $2\mathcal{N}_{\lfloor n/2 \rfloor}(\Pi) \leq \binom{n}{2} - \mathcal{N}_{\leq \lfloor n/2 \rfloor - 2}(\Pi)$ and the result follows.

For odd n, let $\tau_1', \tau_2', ..., \tau_w'$ with $w = \mathcal{N}_{(n-3)/2}(\Pi)$ be the $(n-3)/2$-critical transpositions of a halfperiod Π ordered by their occurrence within Π. Assume without loss of generality that the first transposition of Π is τ_1'. Let b_i be the number of $(n-1)/2$-critical transpositions that occur after τ_i' and before τ_{i+1}' (or the end of the halfperiod if $i=w$). Note that $b_i \leq 3$ since the three elements in the center (that is, those elements in the middle three positions $(n \pm 1)/2$ and $(n+3)/2$) remain in the center between two consecutive $(n-3)/2$-critical transpositions. We prove that if $b_i = b_j = 3$ for some $i<j$ and no other b in between equals 3, then there is some l between i and j such that $b_l \leq 1$. Thus either at most one $b_i = 3$ or the average of $b_1, b_2, ..., b_w$ is ≤ 2. Thus $\mathcal{N}_{(n-1)/2}(\Pi) = \sum_{i=1}^{w} b_i \leq 2w+1 = 2\mathcal{N}_{(n-3)/2}(\Pi) + 1$.

Now note that $j \neq i+1$ since all three elements in the center were transposed between τ_i' and τ_{i+1}' and two of them remain in the center between τ_{i+1} and τ_{i+2} (or the end of Π). That is, $b_{i+1} \leq 2$. Assume by way of contradiction that $b_l = 2$ for all $i < l < j$. One of the three transpositions after τ'_j does not involve the new element brought to the center by τ'_j. Thus this transposition can take place right before τ'_j without modifying $\mathcal{N}_{\le k}(\Pi)$ (the other two transpositions switched order). But now $b_{j-1} = 3$, i.e., the gap between “threes” was reduced. We can do the same until $b_{i+1} = 3$ which is impossible. Finally since $\mathcal{N}_{\le \lfloor n/2 \rfloor}(\Pi) = \binom{n}{2}$ then $3\mathcal{N}_{\lfloor n/2 \rfloor}(\Pi) \le 2\binom{n}{2} - 2\mathcal{N}_{\le \lfloor n/2 \rfloor - 2}(\Pi) + 1$ and the result follows. □

3 New exact values of $h(n), \widetilde{h}(n), \overline{\mathrm{cr}}(K_n), \widetilde{\mathrm{cr}}(K_n)$

Theorem 1 gives a new upper bound for $\widetilde{h}(n)$ if we use the bound (B) for $\mathcal{N}_{\le \lfloor n/2 \rfloor - 2}(\Pi)$. The numerical values of this bound in our range of interest are shown in Table 1. From Theorem 1 and the fact that $\mathcal{N}_{\le \lfloor n/2 \rfloor - 1}(\Pi) = \binom{n}{2} - \mathcal{N}_{\lfloor n/2 \rfloor}(\Pi)$ we obtain that $\mathcal{N}_{\le \lfloor n/2 \rfloor - 1}(\Pi) \ge \left\lceil \frac{1}{2}\binom{n}{2} + \frac{1}{2}\mathcal{N}_{\le \lfloor n/2 \rfloor - 2}(\Pi) \right\rceil$ if n is even, and $\mathcal{N}_{\le \lfloor n/2 \rfloor - 1}(\Pi) \ge \left\lceil \frac{1}{3}\binom{n}{2} + \frac{2}{3}\mathcal{N}_{\le \lfloor n/2 \rfloor - 2}(\Pi) - \frac{1}{3} \right\rceil$ if n is odd. Then, by applying the bound in (B) for $\mathcal{N}_{\le \lfloor n/2 \rfloor - 2}(\Pi)$, we get for $n \ge 10$

$$\mathcal{N}_{\le \lfloor n/2 \rfloor - 1}(n) \ge \begin{cases} \binom{n}{2} - \left\lfloor \frac{1}{24} n(n+30) - 3 \right\rfloor & \text{if } n \text{ is even,} \\ \binom{n}{2} - \left\lfloor \frac{1}{18}(n-3)(n+45) + \frac{1}{9} \right\rfloor & \text{if } n \text{ is odd.} \end{cases}$$

This lower bound is at least as good as (B) with $k = \lfloor n/2 \rfloor - 1$ for all even $n \ge 10$ and all odd $n \ge 21$. In Table 1 we show the bounds obtained for our range of n values. We also calculate a new lower bound for $\widetilde{\mathrm{cr}}(K_n)$ using (A) with $\mathcal{N}_{\lfloor n/2 \rfloor}(n) = \binom{n}{2}$, the previous bound for $\mathcal{N}_{\le \lfloor n/2 \rfloor - 1}(n)$, and (B) for $k \le \lfloor n/2 \rfloor - 2$. All the bounds shown in Table 1 are attained by Aichholzer's et al. constructions [4], and thus Table 1 actually shows the exact values of $\widetilde{h}(n)$, $h(n)$, $\mathcal{N}_{\le \lfloor n/2 \rfloor - 1}(n)$, $\widetilde{\mathrm{cr}}(K_n)$, and $\overline{\mathrm{cr}}(K_n)$ for n in the specified range.

References

[1] B. M. Ábrego, J. Balogh, S. Fernández-Merchant, J. Leaños, and G. Salazar. An extended lower bound on the number of ($\le k$)-edges to generalized configurations of points and the pseudolinear crossing number of K_n. Submitted (2006).

[2] B. M. Ábrego and S. Fernández-Merchant, A lower bound for the rectilinear crossing number, *Graphs and Comb.* **21** (2005), 293–300.

[3] B. M. Ábrego and S. Fernández-Merchant. Geometric drawings of K_n with few crossings. *J. Combin. Theory Ser. A* **114** (2007), 373-379.

[4] O. Aichholzer. Rectilinear crossing number page. Available at: `http://www.ist.tugraz.at/staff/aichholzer/crossings.html`.

[5] O. Aichholzer, J. García, D. Orden, and P. Ramos, New lower bounds for the number of ($\leq k$)-edges and the rectilinear crossing number of K_n. *Discrete and Comp. Geom.*, to appear.

[6] A. Andrzejack, B. Aronov, S. Har-Peled, R. Seidel, and E. Welzl. Results on k-sets and j-facets via continuous motions. In *Proceedings of the 14th Annual ACM Symposium on Computational Geometry* (1998), 192-199.

[7] A. Beygelzimer and S. Radziszowski, On halving line arrangements. *Discrete Mathematics* **257** (2002), 267-283.

[8] P. Brass, W. Moser and J. Pach, *Research Problems in Discrete Geometry*, Springer-Verlag, New York, 2005.

[9] D. Eppstein, Sets of points with many halving lines. Technical Report ICS-TR-92-86, U. California, Irvine, Dept. of Inf. and Comp. Sci., August 1992.

[10] P. Erdős and R. K. Guy, Crossing number problems, *Amer. Math. Monthly* **80** (1973), 52–58.

[11] P. Erdős, L. Lóvasz, A. Simmons, E. G. Straus, Dissection graphs of planar point sets. In: A survey of Combinatorial Theory, J. N. Srivastava et al., eds., North-Holland 1973, 139-149.

[12] J. E. Goodman, R. Pollack, Proof of Grünbaum's conjecture on the stretchability of certain arrangements of pseudolines. *J. Combin. Theory Ser. A* **29** (1980), no. 3, 385-390.

[13] J. E. Goodman, R. Pollack, On the combinatorial classification of nondegenerate configurations in the plane, *J. Combin. Theory Ser. A* **29** (1980), 220-235.

[14] L. Lóvasz, On the number of halving lines, Ann. Univ. Sci. Budapest Eötvös Sect. Math. **14** (1971) 107-108.

[15] L. Lovász, K. Vesztergombi, U. Wagner, E. Welzl, Convex quadrilaterals and k-sets. In: Pach, J. editor: *Towards a theory of geometric graphs*, Contemporary Mathematics Series, **342**, AMS 2004, 139-148.

[16] G. Stöckl, Gesammelte and neue Ergebnisse über extreme k-Mengen für ebene Punktmengen. Diplomarbeit, Technische Univ. Graz, Inst. für Informationsverarbeitung (1984).

Iterated Clique Graphs and Contractibility

F. Larrión[1]

Instituto de Matemáticas,
Universidad Nacional Autónoma de México.
04510 México D.F., MEXICO.

M.A. Pizaña[2]

Universidad Autónoma Metropolitana, Depto. de Ingeniería Eléctrica.
Av. San Rafael Atlixco 186. Col. Vicentina.
México 09340 D.F. MEXICO.

R. Villarroel-Flores[3]

Centro de Investigación en Matemáticas,
Universidad Autónoma del Estado de Hidalgo.
Carr. Pachuca-Tulancingo km. 4.5, Pachuca Hgo. 42184, MEXICO.

Abstract

To any graph G we can associate a simplicial complex $\Delta(G)$ whose simplices are the complete subgraphs of G, and thus we say that G is *contractible* whenever $\Delta(G)$ is so. We study the relationship between contractibility and K-nullity of G, where G is called *K-null* if some iterated clique graph of G is trivial. We show that there are contractible graphs which are not K-null, and that any graph whose clique graph is a cone is contractible.

Keywords: clique graphs, homotopy type, contractibility.

This extended abstract reports on a part of the paper [16], where more results, details and proofs can be found.

Our graphs are simple, finite, connected and non-empty. Making a noun out of an adjective, we often refer to complete subgraphs just as *completes*. We identify induced subgraphs (hence completes) with their vertex sets. A *clique* of a graph is a maximal complete. The *clique graph* of a graph G is the intersection graph $K(G)$ of the set of all cliques of G. *Iterated clique graphs* $K^n(G)$ are defined by $K^0(G) = G$ and $K^{n+1}(G) = K(K^n(G))$.

We say that G is *K-null* if $K^n(G)$ is the *trivial* (i.e. one-vertex) graph for some $n \geq 0$; if n is minimal, it is called the (*nullity*) *index* of G. More generally, if there are m, n with $m \neq n$ such that $K^m(G) \cong K^n(G)$, we say that G is *K-convergent*. It is easy to see that if G is not K-convergent, then the sequence of orders $|K^n(G)|$ tends to infinity, and in this case we say that G is *K-divergent*. The first examples of K-divergent graphs were given by Neumann-Lara [18]: defining the *n-th octahedron* O_n as the complement of the disjoint union of n copies of K_2, then one has $K(O_n) \cong O_{2^{n-1}}$, and so O_n is K-divergent for $n \geq 3$.

Given a graph G, the *complex of completes* of G is the simplicial complex $\Delta(G)$ whose simplices (or faces) are the complete subgraphs of G. On the other hand, we say that a simplicial complex Δ is *Whitney* if $\Delta = \Delta(G)$ for some graph G (note that the only candidate for G is the 1-skeleton of Δ). Whitney complexes are also called *flag complexes* or *clique complexes* in the literature. We can thus attach topological concepts to G via the geometric realization $|\Delta(G)|$ of its associated complex. For instance, we say that a graph G is a *disk* (or a *sphere*) whenever $|\Delta(G)|$ is so, in which case we can also say that G is a *Whitney triangulation* of the disk (or a sphere). Again, G is *contractible* when $|\Delta(G)|$ is so and, more generally, we refer to the *homotopy type* of G as that of $|\Delta(G)|$. For example, the homotopy type of O_n is that of the sphere S^{n-1}.

The study of the clique operator under the topological viewpoint of the complex of completes was initiated by Prisner in [19] and has been further pursued in [10,11,12,13,15]. In this work, we explore the relation between K-nullity and contractibility of graphs. For a long time, we thought that several examples, results, problems and conjectures in the literature hinted at the equivalence of these concepts. Let us just mention three of them: Trees, which

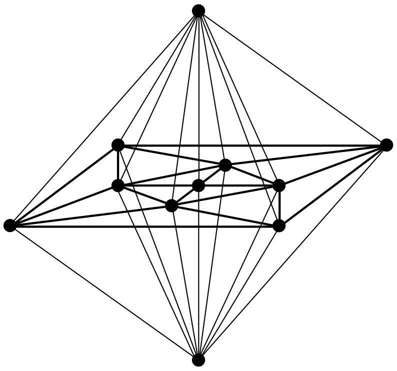
[1] Email: `paco@math.unam.mx`

[2] Email: `map@xanum.uam.mx`

[3] Email: `rafaelv@uaeh.edu.mx`

are the easiest examples of contractible graphs, are known to be K-null since the earliest result on iterated clique graphs: Hedetniemi and Slater proved in [9] that if G is connected, triangleless and with at least three vertices, then $K^2(G)$ is obtained from G by removing the vertices of degree one. Or take the K-null graphs F_n, H_n^i which were studied by Bornstein and Szwarcfiter in [4]: they are Whitney triangulations of the disk, thus contractible. In fact, it is conjectured in [14] that every Whitney triangulation of the disk is K-null, and this has been proved in [13] for the particular case in which each interior vertex has degree at least six.

We proved in this work that K-nullity and contractibility are not equivalent, as there are contractible graphs which are K-divergent:



In fact, this example also disproves a related conjecture we upheld for some time: this graph is contractible but its clique graph is not. By adapting this example we can obtain a comparability graph which is contractible and K-divergent. This gives a new answer to a problem in [8] and [22], and also settles a question that remained unanswered in [14].

The remaining question of whether K-null graphs are always contractible was also tackled. We believe that this is true, but it seems to be difficult to prove and we give only partial results. The first non-trivial case (index 2) is posed by *clique-complete graphs* (i.e. graphs G with $K(G)$ complete) which were previously studied by Lucchesi, de Mello and Szwarcfiter [17]. Using a result that goes back to Prisner [19] one can see that all critical clique-complete graphs are contractible: one uses the classification in [17] and the fact that these graphs have always a dominated vertex whose removal leads to a graph having a universal vertex. But not all clique-complete graphs can be shown to be contractible by this kind of simple argument, as some of the non-critical ones do not have dominated vertices. Thus, we need stronger techniques even for clique-complete graphs.

It turns out that not only all clique-complete graphs are contractible, but that also all *clique-cone graphs* (i.e. graphs G such that $K(G)$ is a *cone*, that is, has a universal vertex) are so. Clique-cone graphs are a large part of the second (index 3) non-trivial case, but they are not all: a graph G has index 3 iff $K^2(G)$ is complete, but this does not imply that $K(G)$ is a cone.

Our result on clique-cone graphs is obtained by first proving a result on simplicial quotients which holds only for Whitney complexes and may be of independent interest: If Δ is a simplicial complex and $\sim$ is an equivalence relation in $V(\Delta)$, the *simplicial quotient* $\Delta/\!\sim$ has vertex set $V(\Delta)/\!\sim$ and faces $\{\, \pi(\sigma) : \sigma \in \Delta \,\}$, where $\pi : V(\Delta) \to V(\Delta/\!\sim)$ is the natural projection. It follows immediately that $\Delta/\!\sim$ is indeed a simplicial complex and that each of its maximal faces is the image under π of a maximal face of Δ. We only use simplicial quotients $\Delta/\!\sim$ where $\sim$ has just one non-singular equivalence class and this class is a face C of Δ; in this case, we denote the quotient as $\Delta \,/\, C$ and say that it is obtained from Δ by *shrinking* the face C. In general, $\Delta \,/\, C$ does not need to have the same homotopy type as Δ. However, for Whitney complexes, shrinking a face does not alter the homotopy type. Indeed, we have the following result, which is the strongest possible simplicial analogue of the Contractible Subcomplex Lemma (see Lemma 10.2 of [3]):

Theorem 0.1 *Let G be a graph and $\Delta = \Delta(G)$ its associated simplicial complex. Let C be a complete subgraph of G and put $\Delta' = \Delta \,/\, C$. Then $\Delta' \simeq \Delta$.*

References

[1] K. Baclawski and A. Björner. *Fixed points in partially ordered sets.* Adv. in Math. **31** (1979) 263–287.

[2] H.J. Bandelt and E. Prisner. *Clique graphs and Helly graphs.* J. Combin. Theory (B) **51** (1991) 34–45.

[3] A Björner. *Topological methods.* In R.L. Graham, M. Grötschel, and L. Lovász, eds. *Handbook of Combinatorics*, vol. II, ch. 34. North-Holland, Amsterdam, 1995.

[4] C.F. Bornstein and J.L. Szwarcfiter. *On clique convergent graphs.* Graphs Combin. **11** (1995) 213–220.

[5] M.E. Frías-Armenta, V. Neumann-Lara and M.A. Pizaña. *Dismantlings and iterated clique graphs.* Discrete Math. **282** (2004) 263–265.

[6] J. Ginsburg. *Dismantlability revisited for ordered sets and graphs and the fixed-clique property.* Canad. Math. Bull. **37** (1994) 473–481.

[7] R. Hamelink. *A partial characterization of clique graphs.* J. Combin Theory (B) **5** (1968) 192–197.

[8] S. Hazan and V. Neumann-Lara. *Fixed points of posets and clique graphs.* Order **13** (1996) 219–225.

[9] S.T. Hedetniemi and P.J. Slater. *Line graphs of triangleless graphs and iterated clique graphs.* Springer Lecture Notes in Mathematics **303** (1972) 139–147.

[10] F. Larrión and V. Neumann-Lara. *Locally C_6 graphs are clique divergent.* Discrete Math. **215** (2000) 159–170.

[11] F. Larrión, V. Neumann-Lara and M.A. Pizaña. *On the homotopy type of the clique graph.* J. of the Brazilian Comp. Soc. **7** (2002) 69–73.

[12] F. Larrión, V. Neumann-Lara and M.A. Pizaña. *Whitney triangulations, local girth and iterated clique graphs.* Discrete Math. **258** (2002) 123–135.

[13] F. Larrión, V. Neumann-Lara and M.A. Pizaña. *Clique convergent surface triangulations.* Mat. Contemp. **25** (2003) 135–143.

[14] F. Larrión, V. Neumann-Lara and M.A. Pizaña. *Clique divergent clockwork graphs and partial orders.* Discrete Appl. Math. **141** (2004) 195–207.

[15] F. Larrión, V. Neumann-Lara and M.A. Pizaña. *Graph relations, clique divergence and surface triangulations.* J. Graph Theory **51** (2006) 110–122.

[16] F. Larrión, M.A. Pizaña and R. Villarroel-Flores. *Contractibility and the clique graph operator.* Submitted.

[17] C.L. Lucchesi, C.P. de Mello and J.L. Szwarcfiter. *On clique-complete graphs.* Discrete Math. **183** (1998) 247–254.

[18] V. Neumann-Lara. *On clique-divergent graphs.* In *Problèmes combinatories et Théorie de Graphes*, number 260, pages 313–315, Orsay, France, 1978. Colloques Internationaux C.N.R.S.

[19] E. Prisner. *Convergence of iterated clique graphs.* Discrete Math. **103** (1992) 199–207.

[20] D. Quillen. *Homotopy properties of the poset of nontrivial p-subgroups of a group.* Adv. in Math. **28** (1978) 101–128.

[21] I. Rival. *A fixed point theorem for partially ordered sets.* J. Combin. Theory (A) **21** (1976) 309–318.

[22] B.S.W. Schröder. *Ordered sets. An introduction.* Birkhäuser Boston Inc., Boston, MA, 2003.

A central approach to bound the number of crossings in a generalized configuration.

Bernardo M. Ábrego
Silvia Fernández–Merchant

Department of Mathematics, California State University at Northridge

Jesús Leaños
Gelasio Salazar

Instituto de Física, Universidad Autónoma de San Luis Potosí, México

Abstract

A generalized configuration is a set of n points and $\binom{n}{2}$ pseudolines such that each pseudoline passes through exactly two points, two pseudolines intersect exactly once, and no three pseudolines are concurrent. Following the approach of allowable sequences we prove a recursive inequality for the number of $(\leq k)$-sets for generalized configurations. As a consequence we improve the previously best known lower bound on the pseudolinear and rectilinear crossing numbers from $0.37968\binom{n}{4} + \Theta\left(n^3\right)$ to $0.379972\binom{n}{4} + \Theta\left(n^3\right)$.

Keywords: k-sets, $\leq k$-sets, rectilinear crossing number, pseudolinear crossing number, complete graphs

1 Introduction

A *pseudoline* is a curve in the projective plane whose removal does not disconnect it, or alternatively, a simple curve in the plane that extends infinitely on both directions. A *generalized configuration* consists of $\binom{n}{2}$ pseudolines and n points such that each pseudoline passes through exactly two points, two pseudolines intersect exactly once, and no three pseudolines are concurrent. A *simple allowable sequence* on n points is a doubly infinite sequence $\Pi = (\ldots, \pi_{-1}, \pi_0, \pi_1, \ldots)$ of permutations on n elements, such that any two consecutive permutations differ by a transposition of neighboring elements, and such that for every j, π_j is the reversed permutation of $\pi_{j+\binom{n}{2}}$. Then Π is determined by $\pi_0, \ldots, \pi_{\binom{n}{2}}$ and any given transposition occurs exactly once on this interval.

Allowable sequences were introduced by Goodman and Pollack [8], who proved a correspondence between the set of simple allowable sequences and the set of generalized configuration of points. When all the pseudolines are straight lines, the generalized configuration is completely determined by the set of points. In [2] two well-known concepts for configurations of points in general position in the plane were extended to generalized configurations: *k-sets* and *rectilinear crossing number* of K_n (see [6] for more references and related problems). Given an n-point set P and $k \leq n/2$, a k-set is a subset of k points in P that can be separated from the rest by a straight line. The rectilinear crossing number of K_n, denoted $\overline{\mathrm{cr}}(n)$, is the maximum number of crossings determined by a complete geometric graph on n vertices. In the more general setting, the *k-sets* of Π are the subsets of $\{1, 2, \ldots, n\}$ of size k that occupy the first or last k positions in a permutation of Π. These are determined by *k-critical transpositions*, that is, those that occur between elements in positions k and $k+1$ or $n-k$ and $n-k+1$. Denote by $\mathcal{N}_k(\Pi)$ the number of k-critical transpositions of Π. Also let $\mathcal{N}_{\leq k}(\Pi) = \sum_{j=1}^{k} \mathcal{N}_j(\Pi)$ and $\mathcal{N}_{>k}(\Pi) = \binom{n}{2} - \mathcal{N}_{\leq k}(\Pi)$, called the number of $(\leq k)$*-sets* and $(> k)$*-sets* of Π, respectively. Let $\widetilde{\mathrm{cr}}(\Pi)$ denote the number of pseudoline crossings in Π. Then the *pseudolinear crossing number* of K_n is defined as $\widetilde{\mathrm{cr}}(n) = \min_{|\Pi|=n} \widetilde{\mathrm{cr}}(\Pi)$. Clearly $\widetilde{\mathrm{cr}}(n) \leq \overline{\mathrm{cr}}(n)$. The next relationship between k-sets and crossing numbers was proved in [2] and [9],

$$\widetilde{\mathrm{cr}}(\Pi) = \sum_{k=1}^{\lfloor n/2 \rfloor} (n - 2k - 1)\, \mathcal{N}_{\leq k}(\Pi) + \Theta(n^3). \tag{1}$$

The problem of determining the rectilinear crossing number was proposed by Erdős and Guy [7] and is equivalent to finding the minimum number of convex quadrilaterals in a set of n points in general position. The best known upper bound, $\widetilde{\mathrm{cr}}(n) \leq \overline{\mathrm{cr}}(n) \leq 0.380548\binom{n}{4} + \Theta(n^3)$, is attained by a recursive construction in [3] using as a starting point a suitable 90-point set. On the other hand, $\mathcal{N}_{\leq k}(\Pi)$ was bounded in [1] (and in [5] in the rectilinear case) by

$$\mathcal{N}_{\leq k}(\Pi) \geq 3\tbinom{k+1}{2} + 3\tbinom{k+1-\lfloor n/3 \rfloor}{2} - \max\{0, (k - \lfloor n/3 \rfloor)(n - 3\lfloor n/3 \rfloor)\}. \quad (2)$$

This in turn implies the previously best known lower bound [1]: $\overline{\text{cr}}(n) \geq \widetilde{\text{cr}}(n) \geq 0.37968\binom{n}{4} + \Theta(n^3)$.

In this work we first bound the $(> k)$-sets in terms of the k-sets in Theorem 2.1. Using this theorem with (1) and (2), we obtain $\overline{\text{cr}}(n) \geq \widetilde{\text{cr}}(n) \geq \frac{277}{729}\binom{n}{4} + \Theta(n^3) > 0.379972\binom{n}{4} + \Theta(n^3)$. It is important to note that this is the first time that a lower bound on $\overline{\text{cr}}(n)$ and $\widetilde{\text{cr}}(n)$ follows from the *central behavior* of a generalized configuration (a bound on the $(> k)$-sets, i.e., all k-sets with k close to $n/2$) rather than on its *boundary behavior* (a lower bound on the $(\leq k)$-sets, i.e. all k-sets with k far from $n/2$).

2 The Central Theorem

Given a permutation π_j of Π and $k \leq n/2$, we denote by $C(k, \pi_j)$, the set of elements in the middle $n - 2k$ positions of the permutation π_j. Let

$$s_k = \min\left\{ |C(k, \pi_0) \cap C(k, \pi_i)| : 0 \leq i \leq \binom{n}{2} \right\}.$$

Theorem 2.1 *Let Π be a generalized configuration of n points. Then*

$$\mathcal{N}_{>k}(\Pi) \leq (n - 2k - 1)\mathcal{N}_k(\Pi) - \frac{s_k}{2}(\mathcal{N}_k(\Pi) - (n-1)).$$

In [4] we prove Theorem 2.1 for $k = \lfloor n/2 \rfloor - 1$. In fact, for such k we prove $\mathcal{N}_{>k}(\Pi) \leq (n - 2k - 1)\mathcal{N}_k(\Pi)$ which does not depend on s_k but is *not true* for all k. Intuitively, to bound the number of transpositions in the middle $n - 2k - 1$ positions of Π it is important to know how many elements remain in the middle positions at any given time. This is why we must involve s_k. Given Π, we first modify it in such a way that the transpositions within the first or last k positions remain intact. This transformation also guarantees that $\mathcal{N}_{>k}(\Pi)$, $\mathcal{N}_k(\Pi)$, and s_k do not change, and allows us to control the $(> k)$-critical transpositions to count them depending on s_k.

Because of the space limitations we cannot include the proof of the theorem here. It will appear in the full version of this paper. We choose instead to show the consequences of this result to the number of $(\leq k)$-sets and to the pseudolinear and rectilinear crossing numbers.

3 New Lower Bound for $\mathcal{N}_{\leq k}(\Pi)$

Let $m = \lceil (4n-2)/9 \rceil$. Define for every n the following recursive sequence:

$$u_{m-1} = 3\binom{m}{2} + 3\binom{m - \lfloor n/3 \rfloor}{2} - 3\left(m - 1 - \left\lfloor \frac{n}{3} \right\rfloor\right)\left(\frac{n}{3} - \left\lfloor \frac{n}{3} \right\rfloor\right) \text{ and}$$

$$u_k = \left\lceil \frac{1}{n-2k}\left(\binom{n}{2} + (n-2k-1)u_{k-1}\right)\right\rceil \text{ for } k \geq m.$$

The next lemmas give useful estimates for the sequence u_k and are needed for the proofs of Theorem 3.3 and Corollary 3.4. They can be proved by elementary inductive arguments.

Lemma 3.1 *For any k, $m-1 \leq k \leq (n-3)/2$:*

$$3\sqrt{1 - \frac{2k+5/2}{n}} \leq \frac{\binom{n}{2} - u_k}{\binom{n}{2} - u_{m-1}} \leq 3\sqrt{1 - \frac{2k}{n}}.$$

Lemma 3.2 *For any k, $m \leq k \leq (n-3)/2$:*

$$3\sqrt{1 - \frac{2k+5/2}{n}}\left(\binom{n}{2} - u_{m-1}\right) \geq (n-1)(n-2k-1).$$

This is the new lower bound that follows once Theorem 2.1 is established:

Theorem 3.3 *For any generalized configuration Π and any k, $m-1 \leq k \leq (n-2)/2$ we have $\mathcal{N}_{\leq k}(\Pi) \geq u_k$.*

Proof of Theorem 3.3. We proceed by induction on k. If $k = m-1$ the result is true by the bound in (2). Assume $k \geq m$ and $\mathcal{N}_{\leq k-1}(\Pi) \geq u_{k-1}$. From Theorem 2.1, if $s = 0$ or $\mathcal{N}_k(\Pi) \geq n-1$ then $\mathcal{N}_{>k}(\Pi) \leq (n-2k-1)\mathcal{N}_k(\Pi)$. Thus

$$\binom{n}{2} - \mathcal{N}_{\leq k}(\Pi) \leq (n-2k-1)\left(\mathcal{N}_{\leq k}(\Pi) - \mathcal{N}_{\leq k-1}(\Pi)\right),$$

and then by induction

$$\mathcal{N}_{\leq k}(\Pi) \geq \frac{1}{n-2k}\left(\binom{n}{2} + (n-2k-1)u_{k-1}\right).$$

This implies that $\mathcal{N}_{\leq k}(\Pi) \geq u_k$ by definition of the sequence u_k.

Now assume $s_k > 0$ and $\mathcal{N}_k(\Pi) < n-1$. It is easy to see that (c.f. [9]) $\mathcal{N}_k(\Pi) \geq 2k+1$, thus $k \leq (n-3)/2$. From Theorem 2.1 we have that

$$\mathcal{N}_{>k}(\Pi) \leq (n-2k-1-\frac{s_k}{2})\mathcal{N}_k(\Pi) + \frac{s_k}{2}(n-1)$$

and since $n-2k-1-s_k/2 > n-2k-1-s_k \geq 0$ and $\mathcal{N}_k(\Pi) < n-1$, then

$$\mathcal{N}_{>k}(\Pi) < (n-2k-1-\frac{s_k}{2})(n-1) + \frac{s_k}{2}(n-1) = (n-1)(n-2k-1)\,.$$

Therefore

$$\mathcal{N}_{\leq k}(\Pi) > \binom{n}{2} - (n-1)(n-2k-1)\,.$$

Then by Lemmas 3.1 and 3.2 we have that $\mathcal{N}_{\leq k}(\Pi) \geq u_k$. □

Corollary 3.4 *For any generalized configuration* Π *and any* k, $m-1 \leq k \leq (n-2)/2$ *we have*

$$\mathcal{N}_{\leq k}(\Pi) \geq \binom{n}{2} - \frac{1}{9}\sqrt{1-\frac{2k}{n}}\left(5n^2-25n+4\right).$$

Proof. Follows directly from Theorem 3.3, Lemma 3.1, and the fact that $u_{m-1} \leq 3\binom{m}{2} + 3\binom{m-\lfloor n/3 \rfloor}{2} \leq 3\binom{(4n+6)/9}{2} + 3\binom{(n+10)/9}{2}$. □

Corollary 3.5 $\overline{\mathrm{cr}}(n) \geq \widetilde{\mathrm{cr}}(n) \geq \frac{277}{729}\binom{n}{4} + \Theta(n^3) > 0.379972\binom{n}{4} + \Theta(n^3)$.

Proof. According to (1), if Π is a generalized configuration on n points then

$$\widetilde{\mathrm{cr}}(\Pi) = \binom{n}{4}\left(24\sum_{k=1}^{\lfloor n/2 \rfloor -1}\frac{1}{n}\left(1-\frac{2k}{n}\right)\frac{\mathcal{N}_{\leq k}(\Pi)}{n^2}\right) + \Theta\left(n^3\right).$$

Now, using (2) we know that for $1 \leq k \leq m-1$,

$$\frac{\mathcal{N}_{\leq k}(\Pi)}{n^2} \geq \frac{3}{2}\left(\frac{k}{n}\right)^2 + \frac{3}{2}\max\left(0, \frac{k}{n}-\frac{1}{3}\right)^2 - \Theta\left(\frac{1}{n}\right).$$

Similarly, if $m \leq k \leq (n-2)/2$ then by Corollary 3.4,

$$\frac{\mathcal{N}_{\leq k}(\Pi)}{n^2} \geq \frac{1}{2} - \frac{5}{9}\sqrt{1-\frac{2k}{n}} + \Theta\left(\frac{1}{n}\right).$$

Therefore

$$\begin{aligned}\widetilde{\operatorname{cr}}(\Pi) &\geq \binom{n}{4}\left(24\int_0^{4/9}\frac{3}{2}(1-2x)\left(x^2+\max\left(0,x-\frac{1}{3}\right)^2\right)dx\right)\\ &+\binom{n}{4}\left(24\int_{4/9}^{1/2}(1-2x)\left(\frac{1}{2}-\frac{5}{9}\sqrt{1-2x}\right)dx\right)+\Theta(n^3)\\ &\geq \binom{n}{4}\left(\frac{86}{243}+\frac{19}{729}\right)+\Theta(n^3)=\frac{277}{729}\binom{n}{4}+\Theta(n^3).\end{aligned}$$

□

References

[1] B. M. Ábrego, J. Balogh, S. Fernández-Merchant, J. Leaños, and G. Salazar. An extended lower bound on the number of ($\leq k$)-edges to generalized configurations of points and the pseudolinear crossing number of K_n. Submitted (2006).

[2] B. M. Ábrego and S. Fernández-Merchant, A lower bound for the rectilinear crossing number, *Graphs and Comb.* **21** (2005), 293–300.

[3] B. M. Ábrego and S. Fernández-Merchant. Geometric drawings of K_n with few crossings. *J. Combin. Theory Ser. A* **114** (2007), 373-379.

[4] B. M. Ábrego, S. Fernández-Merchant, J. Leaños, and G. Salazar. The maximum number of halving lines and the rectilinear crossing number of K_n for $n \leq 27$. Preprint (2007).

[5] O. Aichholzer, J. García, D. Orden, and P. Ramos, New lower bounds for the number of ($\leq k$)-edges and the rectilinear crossing number of K_n. *Discrete and Comp. Geom.*, to appear.

[6] P. Brass, W. Moser and J. Pach, *Research Problems in Discrete Geometry*, Springer-Verlag, New York, 2005.

[7] P. Erdős and R. K. Guy, Crossing number problems, *Amer. Math. Monthly* **80** (1973), 52–58.

[8] J. E. Goodman, R. Pollack, On the combinatorial classification of nondegenerate configurations in the plane, *J. Combin. Theory Ser. A* **29** (1980), 220-235.

[9] L. Lovász, K. Vesztergombi, U. Wagner, E. Welzl, Convex quadrilaterals and k-sets. In: Pach, J. editor: *Towards a theory of geometric graphs*, Contemporary Mathematics Series, **342**, AMS 2004, 139-148.

Nowhere-zero 5-flows and $(1, 2)$-factors

Martín Matamala [a,b] and José Zamora [a,1]

[a] *Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Casilla 170-3, Correo 3, Santiago, Chile.*

[b] *Centro de Modelamiento Matemático Universidad de Chile and UMR 2071-CNRS, Casilla 170-3, Correo 3, Santiago, Chile*

Abstract

A graph G has a nowhere-zero k-flow if there exists an orientation D of the edges and an integer flow ϕ such that for all $e \in D(G), 0 < |\phi(e)| < k$. A $(1, 2)$-factor is a subset of the edges $F \subseteq E(G)$ such that the degree of any vertex in the subgraph induced by F is 1 or 2. It is known that cubic graphs having a nowhere-zero k-flow with $k = 3, 4$ are characterized by properties of the cycles of the graph. We extend these results by giving a characterization of cubic graphs having a nowhere-zero 5-flow based on the existence of a $(1, 2)$-factor of the graph such that the cycles of the graph satisfies an algebraic property.

Keywords: nowhere-zero flow factor

1 Introduction

Let $G = (V, E)$ be a simple directed graph without bridges. A flow in G is a nowhere-zero $k-$flow if its values are in $\{-(k-1), \ldots, k-1\} \setminus \{0\}$.

It is easy to see that if G has a nowhere-zero $k-$flow then every directed graph obtained from G by reversing the orientations of some of its arcs also

[1] Partially supported by CONICYT grant.

has a nowhere-zero $k-$flow. This shows that the existence of a nowhere-zero $k-$flow of G is a property of the underlying undirected graph.

The concept of nowhere-zero k-flow was introduced by Tutte [6] as a refinement and a generalization for the face coloring problem in planar graphs. In terms of the four-color Theorem it says that every bridgeless planar graph has a nowhere-zero 4-flow. This result can not be extended to any bridgeless graph since the Petersen graph has no a nowhere-zero 4-flow. However, in [7] Tutte formulated his famous 5-flow conjecture which still is open:

Conjecture 1.1 *Every bridgeless graph admits a nowhere-zero 5-flow*

Work about Conjecture 1.1 have focused in properties of a minimal counterexample (see [3],[2],[1]) and into the study of structural properties of graphs having a nowhere-zero 5-flow (see [5]).

The best approximation for Conjecture 1.1 is a result of Seymour [4] where he proved that every bridgeless graph has a nowhere-zero 6-flow.

The motivation for studying this conjecture in cubic graphs has two sources. First, it is known that for the Conjecture 1.1 to be true, it is enough to prove it for cubic graphs. Second, for cubic graphs there exist well known characterizations of the existence of nowhere-zero k-flow for $k = 3, 4$. These characterizations provide some intuition about the structural properties of cubic graph admitting nowhere-zero $k-$flow.

On one hand, Tutte gave the following characterization.

Theorem 1.2 *[6] Let G be a cubic graph. G admits a nowhere-zero 3-flow if and only if G is bipartite*

This result can be seen as a parity condition that must satisfy all cycles of a cubic graph to admit a nowhere-zero $3-$flow. The parity condition being that each cycle has even length.

On the other hand, for nowhere-zero $4-$flow we only need to check this parity property in the complement of a perfect matching.

Theorem 1.3 *Let $G = (V, E)$ be a cubic graph. G has a nowhere-zero 4-flow if and only if G has a perfect matching M such that all the cycles in $G - M$ have even length.*

This condition can also be formulated in a more algebraic way: for each cycle C of G, $l(C)2^{|M \cap E(C)|} \equiv 0$ modulo 2, where $l(C)$ is the length of C and $E(C)$ is the set of edges of C. Notice that in this formulation the condition depends on M and must be satisfied for all the cycles.

In Section 2 we state Theorem 2.2 which is our main result. It is somehow similar to Theorem 1.3 since it characterizes the existence of nowhere-zero $5-$flow in terms of an algebraic condition that must satisfy all cycles. In our case, the algebraic condition is stated in terms of a subset of edges F such that each vertex of G is incident with 1 or 2 edges of F. This kind of sets is known as $(1,2)$-factors. As a by product, we obtain in Section 3 a relation between nowhere-zero 5-flow and 4-edge-colorings.

It is known by ([7]) that a graph admits a nowhere-zero k-flow if and only if it admits a Γ-flow where Γ is an abelian group of cardinality k.

In the following, we will work with nowhere-zero $\mathbb{Z}_k$-flows instead of nowhere-zero k-flows and we will consider only cubic graphs.

2 Nowhere-zero $\mathbb{Z}_5$-flow and $(1,2)-$factors

In this section we give a new characterization of cubic graphs admitting a nowhere-zero $\mathbb{Z}_5$-flow.

We note that in the group $\mathbb{Z}_5$ the flow equation for cubic graphs have the following property:

Lemma 2.1 *In the group $\mathbb{Z}_5$, $x+y+z=0$ if and only if $\{x,y,z\} \in \{\{1,1,3\},\{4,4,2\},\{2,2,1\},\{3,3,4\}\}$*

In order to state our main result, we need some definitions.

Let us assume that G has a nowhere-zero $\mathbb{Z}_5$-flow ϕ. By Lemma 2.1, it is easy to see that at each vertex at least one edge has flow 1 or 4 and at least one edge has flow 2 or 3. Let us color the edges having flow 1 or 4 with color red and the remaining edges with color blue. This splits the set of edges into two $(1,2)-$factors corresponding to the sets of edges with color red and the set of edges with color blue, which we call the $(1,2)-$factors *induced by the nowhere-zero $\mathbb{Z}_5-$flow ϕ.*

Now let F be a $(1,2)-$factor of the graph $G=(V,E)$. We say that two adjacent edges (sharing a vertex) are $F-$related if both belong to F or both belong to $E \setminus F$. Let v be a vertex and let e, e' and e'' be its three incident edges. The $F-$parity of the pair (e,e'), $p_F(e,e')$, is 2 when e, e' are $F-$related. It is 1 when e and e'' are $F-$related and it is 3 when e' and e'' are $F-$related. Notice that the $F-$parity of (e',e) is the inverse in $\mathbb{Z}_4$ of the $F-$parity of (e,e') (see Figure 1 where the continuous edges are in F).

Let $\overrightarrow{C}=(e_1,e_2,\ldots,e_n,e_1)$ be a cycle of G. We define the $F-$parity of $\overrightarrow{C}$ as the sum of the $F-$parities of (e_i,e_{i+1}), for $i=1,\ldots,n-1$, plus the $F-$parity of (e_n,e_0). Notice that the F-parity of $\overrightarrow{C}$ is the additive inverse in

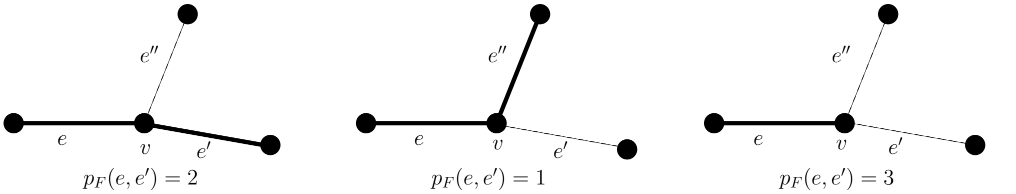


Fig. 1.

$\mathbb{Z}_4$ of the F-parity of the $\overleftarrow{C} = (e_n, e_{n-1}, \dots e_1, e_n)$, the cycle with the reverse order. We say that C is $F-$null if its $F-$parity of $\overrightarrow{C}$ is zero in $\mathbb{Z}_4$.

Now we can state our main result which is a characterization of cubic graphs admitting a nowhere-zero $\mathbb{Z}_5-$flow.

Theorem 2.2 *Let $G = (V, E)$ be an undirected cubic graph. G admits a nowhere-zero $\mathbb{Z}_5$-flow if and only if there exists a $(1, 2)$-factor F such that every cycle C of G is F-null*

Proof (Sketch)

To proof the forward direction we define F as a $(1, 2)-$factor induced by a nowhere-zero $\mathbb{Z}_5-$flow ϕ of G. It is not hard to see that the sum of the $F-$parities over the vertices in a path P of G is zero modulo 4 if and only if the sum of $\phi(e)$ over the edges not in P incident with the inner vertices of P is zero modulo 5. In order to prove that each cycle is $F-$null we use the fact that for each cycle C the sum of $\phi(e)$ over the cut defined by C is zero modulo 5.

To prove the backward implication we take a spanning tree $T \subseteq E$ of G with root r and $D(E)$ an orientation of G. We will define a nowhere-zero $\mathbb{Z}_5$-flow such that an arc has value 1 or 4 if it is in F and value 2 or 3 if not. We note that under the previous assumption and by Lemma 2.1, for any vertex v there are only two possibles solutions for the flow equation of its incident arcs. Let us call $\varphi_{v,i}$, with $i = 1, 2$ these solutions. We define a function $\pi : V \to \{1, 2\}$ such that $\pi(r) = 1$ and for all arcs $e = (u, v)$ in T, $\varphi_{u,\pi(u)}(e) = \varphi_{v,\pi(v)}(e)$. If at some arc $f = (x, y)$ not in T, $\varphi_{x,\pi(x)}(f) \neq \varphi_{y,\pi(y)}(f)$, it can be shown that the cycle formed by f in T is not $F-$null. Since for all the arcs its endvertices define the same solution, then the function $\varphi : D(E) \to \mathbb{Z}_5$ such that for all $e = (u, v) \in D(E), \varphi(e) = \varphi_{u,\pi(u)}(e)$ is a nowhere-zero $\mathbb{Z}_5$-flow of G. □

Note 1 *We note that as we only need to check the property for the cycles defined by edges not in T, given the $(1, 2)-$factor F we can check in polynomial time whether it is induced by a nowhere-zero $\mathbb{Z}_5-$flow.*

3 Nowhere-zero 5-flow and 4-edge coloring

In this section we show that Theorem 2.2 can be thought as a characterization of $4-$edge-colorings which defines nowhere-zero 5-flow.

On one hand, the following Theorem completely characterizes the existence of nowhere-zero 4-flow in terms of edge-colorings.

Theorem 3.1 *[7] Let G be a cubic graph. G has a nowhere-zero 4-flow if and only if G is 3-edge coloreable.*

On the other hand, the following result shows that if Conjecture 1.1 is true then, for all cubic bridgeless graphs there exists a nowhere-zero 5-flow whose values defines or induces a 4-edge-coloring, it means that for any vertex v the values of the flow in the edges incident with v are all different. We say that a vertex v is bad for a nowhere-zero $\mathbb{Z}_5$-flow (φ, D) if there exists two edge in its incident edges with the same flow. A path of G is directed if all the interior vertices of the path have one ingoing and one outgoing arc. A directed path is $(x, -x)$-alternating with $x \in \mathbb{Z}_5$ if the value of the flow in its edges alternate between x and $-x$.

Proposition 3.2 *Let $G = (V, E)$ be a cubic graph having a nowhere-zero 5-flow ϕ. Then there exists a nowhere-zero 5-flow ϕ' that induces a 4-edge-coloring of G.*

Proof. Let (φ', D') be a nowhere zero 5-flow that minimize the number of bad vertices. We assume that there exist a bad vertex u with two incidents arcs with the same flow x. We note that both start in u or both finish in u because the third arc can not be zero. In both cases, let $e_1 = (u, v)$ one of them and let $P = e_1 e_2 \dots e_l$ the longest directed alternating $(x, -x)$-path starting with e_1. Note that all the interior vertex of P are not bad. Then for all $e \in P$ we define $D''(e) = -D'(e)$ and $\varphi''(e) = -x$. Note that in $\mathbb{Z}_5, x \neq -x$ and the third arc can not be neither x or $-x$. For the rest of the arcs e we define $D''(e) = D'(e)$ and $\varphi''(e) = \varphi'(e)$. Note that for all $v \in V$ the flow equation do not change. Then (φ'', D'') is a nowhere-zero 5-flow with less bad vertices than (φ', D'). This contradict the minimality of (φ', D'), then no such u can exist, so (φ', D') induce a 4-edge coloring. □

Unfortunately, there are 4-edge-colorings not induced by a nowhere-zero 5-flow. In the Figure 2 we show a 4-edge coloring of the Petersen graph which is not induced by a nowhere-zero $\mathbb{Z}_5$-flow.

Nevertheless, Theorem 2.2 gives a condition for $4-$edge colorings so as they are induced by nowhere-zero 5-flow. In fact, each $4-$edge-coloring defines six

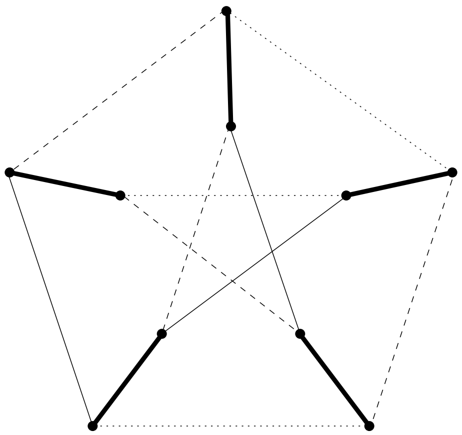


Fig. 2.

possibles $(1,2)-$factors. If this $4-$edge coloring is induced by a nowhere-zero 5-flow then, at least one of its associated $(1,2)-$factors must satisfies the condition stated in Theorem 2.2. Then by Note 1 we can check in polynomial time if a 4-edge coloring is induced by a nowhere-zero 5-flow.

References

[1] Celmins, U., *On cubic graphs that do not have an edge-3-coloring*, Ph.D Thesis, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada (1884).

[2] Kochol, M., *Reduction of the 5-flow conjecture to cyclically 6-edge-connected snarks*, J. Combin. Theory Ser. B **90** (2004), pp. 139–145, dedicated to Adrian Bondy and U. S. R. Murty.

[3] Kochol, M., *Restrictions on smallest counterexamples to the 5-flow conjecture*, Combinatorica **26** (2006), pp. 83–89.

[4] Seymour, P. D., *Nowhere-zero* 6-*flows*, J. Combin. Theory Ser. B **30** (1981), pp. 130–135.

[5] Steffen, E., *Tutte's* 5-*flow conjecture for graphs of nonorientable genus* 5, J. Graph Theory **22** (1996), pp. 309–319.

[6] Tutte, W. T., *On the imbedding of linear graphs in surfaces*, Proc. London Math. Soc. (2) **51** (1949), pp. 474–483.

[7] Tutte, W. T., *A contribution to the theory of chromatic polynomials*, Canadian J. Math. **6** (1954), pp. 80–91.

Random Graphs, Retractions and Clique Graphs

F. Larrión[1]

Instituto de Matemáticas
Universidad Nacional Autónoma de México
Mexico City, Mexico

M.A. Pizaña[2]

Departamento de Ingeniería Eléctrica
Universidad Autónoma Metropolitana
Mexico City, Mexico

R. Villarroel-Flores[3]

Centro de Investigación en Matemáticas
Universidad Autónoma del Estado de Hidalgo
Pachuca, Mexico

Abstract

We found $d_1 = d_1(n, p)$ and $d_2 = d_2(n, p)$ such that almost every (random) graph $G \in G(n, p)$ has retractions to d-dimensional octahedra $\mathcal{O}_d$ for every integer d satisfying $d_1 < d < d_2$. This result has several important consequences: (the clique complex of) the random graph has several non trivial homology/homotopy groups, the random graph is not contractible, the random graph is not homotopy equivalent to its clique graph and the random graph is clique divergent.

Keywords: random graphs, graph retractions, clique graphs, clique divergence, homotopy type

For every fixed n and p, $0 < p < 1$, $G(n,p)$ is the probability space over the set of graphs on the vertex set $\{1,\ldots,n\}$ with edge probability p: $\Pr[\{i,j\} \in E(G)] = p$. According to standard practice [4], we say that "Almost every graph $G \in G(n,p)$ has property Q" whenever $\Pr[G(n,p) \models Q] \to 1$ as $n \to \infty$, moreover, following Alon and Spencer [3], we shall also abuse notation and say "The random graph has property Q". For an extremely well written introduction to the topic see [21].

A *morphism* of graphs $f : G \to H$ is a function between their vertex sets $f : V(G) \to V(H)$ such that $x \simeq y$ implies $f(x) \simeq f(y)$. Note that it is possible for adjacent vertices in G, $x \sim y$ to be mapped to the same vertex $f(x) = f(y)$ in H. As usual, a *retraction* is a morphism $\rho : G \to H$ such that there is another morphism $\sigma : H \to G$ satisfying $\rho \circ \sigma = 1_H$.

We investigated the question of whether the random graph has retractions to some other graph. Specially, we are interested in retractions from the random graph to d-dimensional octahedra $\mathcal{O}_d$. By definition $\mathcal{O}_d$ is the only $(2d-2)-$regular graph on $2d$ vertices. It should be clear that $\mathcal{O}_3$ is precisely (the underlying graph of) the octahedron. More specifically, we are interested in determining, for every constant p, the values of $d = d(n,p)$ such that almost every graph $G \in G(n,p)$ has a retraction to $\mathcal{O}_d$. Our investigations yield the following theorem:

Theorem 0.1 *There are real numbers $d_1 = d_1(n,p)$ and $d_2 = d_2(n,p)$, such that almost every graph $G \in G(n,p)$ has retractions to d-dimensional octahedra $\mathcal{O}_d$ for every integer d satisfying $d_1 < d < d_2$. Moreover d_1 and d_2 are given by:*

$$d_1 = \frac{1}{\ln(1/p)}\left(\ln n - \ln\ln n\right)$$

$$d_2 = \frac{1}{\ln(1/p)}\left(\ln n - \frac{1}{2}\ln\ln n\right)$$

We mention here that Matthew Kahle [10] has independently investigated the issue under a different approach: For a fixed d, which values of $p = p(n,d)$ guarantee a retraction from the random graph to $\mathcal{O}_d$. As far as we know, there is no way for proving his results using ours or viceversa. We also mention here that the retractions stated in the Theorem imply that the

[1] Email:paco@math.unam.mx

[2] Email:map@xanum.uam.mx

[3] Email:rafaelv@uaeh.edu.mx

random graph has (maximal) cliques of small sizes d (indeed we need this fact in the proof), compare to Bollobás' result [4] that, for every $\varepsilon > 0$, only guarantees the existence of cliques of sizes $(1+\varepsilon)\frac{1}{2}r_0 < d < (1-\varepsilon)r_0$ for $\frac{1}{2}r_0 = \left(\frac{1}{\ln(1/p)}\right)(\ln n - \ln\ln n + \Theta(1))$ which was not enough for our purposes.

Now, the *clique graph* $K(G)$ of G is the intersection graph of its (maximal) cliques [8,19]. *Iterated clique graphs* [9], are defined recursively by $K^0(G) = G$ and $K^m(G) = K(K^{m-1}(G))$. For extensive bibliography on the topic see [15,22], recent work may be found in [1,2,5,6,12]. Iterated clique graphs have also been used in Quantum Gravity to explain the quantum space-time as an emergent property of an underlying, sub-Planck scale, discrete reality [16,17] and to tackle renormalization problems in Quantum Gravity [18].

The discovery of *clique divergent* graphs (i.e. the sequence of iterated clique graphs of G grows without bound) by Víctor Neumann-Lara originally reported by Escalante in [7] was soon followed by the celebrated Neumann-Lara's Retraction Theorem [13] which states that if $\rho : G \to H$ is a retraction and H is clique divergent, then so is G. The first known examples of clique divergent graphs where precisely the d-dimensional octahedra (indeed $K(\mathcal{O}_d) \cong \mathcal{O}_{2^{d-1}}$), hence, our Theorem yields immediately the following:

Corollary 0.2 *The random graph is clique divergent.*

There is also a natural way to assign topological properties to graphs G using its clique complex $\Delta(G)$ which is the simplicial complex whose facets are precisely the cliques of G [14,11] (we refer to [20] for undefined terms on Algebraic Topology). Then we say, for instance, that G and H are homotopy equivalent $G \simeq H$ if the geometric realizations of their clique complexes are homotopy equivalent $|\Delta(G)| \simeq |\Delta(H)|$. It is easily seen (since all the transformations involved are functors) that a retraction of graphs $\rho : G \to H$ induces a retraction of simplicial complexes $\hat{\rho} : \Delta(G) \to \Delta(H)$, which then induces a retraction of topological spaces $\bar{\rho} : |\Delta(G)| \to |\Delta(H)|$ which finally induces a retraction of d-dimensional homology groups $\tilde{\rho} : H_d(|\Delta(G)|) \to H_d(|\Delta(H)|)$. Now, homology groups are always abelian and retractions between abelian groups are the same as projections onto direct summands. Hence, we conclude that if $\rho : G \to H$ is a retraction and $H_d(H)$ is non trivial, then $H_d(G)$ is also non trivial.

Since $|\Delta(\mathcal{O}_m)|$ is (homeomorphic to) the $(m-1)$-sphere, we know that $H_{m-1}(\mathcal{O}_m) = \mathbb{Z} \neq 0$. On the other hand, the fact that almost every graph $G \in G(n,p)$ has a retraction to $\mathcal{O}_d$, together with Neumann-Lara's Retraction Lemma [13] imply that $K(G)$ has a retraction to $K(\mathcal{O}_d) \cong \mathcal{O}_{2^{d-1}}$

which then implies that $H_{2^{d-1}-1}(K(G)) \neq 0$. Since the random graph does not have cliques of sizes greater than $\frac{2}{\ln(1/p)} \ln n$ [4] it follows that whenever $m > \frac{2}{\ln(1/p)} \ln n$, $H_m(G) = 0$ for almost every graph $G \in G(n,p)$. In particular $H_{2^{d-1}-1}(G) = 0$. Hence:

Corollary 0.3 *The random graph is not homotopy equivalent to its clique graph. That is, for almost every $G \in G(n,p)$, we have $|\Delta(G)| \not\simeq |\Delta(K(G))|$.*

Matthew Kahle started the study of random topological spaces [10] by converting the standard probability spaces of random graphs $G(n,p)$ into the probability spaces of simplicial complexes $\Delta(n,p)$ which are naturally induced by the mapping $G \mapsto \Delta(G)$. The fact that the barycentric subdivision of any simplicial complex is a clique complex, tells us that every simplicial complex has a representative (up to homeomorphism) in $\Delta(n,p)$. In this setting, our Theorem has the following immediate consequences:

Corollary 0.4 *The random complex $\Delta(n,p)$ have non vanishing homology groups in several dimensions. In particular, $\Delta(n,p)$ is not contractible.*

References

[1] Alcón, L., *Clique-critical graphs: maximum size and recognition*, Discrete Appl. Math. **154** (2006), pp. 1799–1802.

[2] Alcón, L., L. Faria, C. M. H. de Figueiredo and M. Gutierrez, *Clique graph recognition is NP-complete*, in: *Graph-theoretic concepts in computer science*, Lecture Notes in Comput. Sci. **4271**, Springer, Berlin, 2006 pp. 269–277.

[3] Alon, N. and J. H. Spencer, "The probabilistic method," Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience [John Wiley & Sons], New York, 2000, second edition, xviii+301 pp., with an appendix on the life and work of Paul Erdős.

[4] Bollobás, B., "Random graphs," Cambridge Studies in Advanced Mathematics **73**, Cambridge University Press, Cambridge, 2001, second edition, xviii+498 pp.

[5] Bonomo, F., G. Durán, M. Groshaus and J. L. Szwarcfiter, *On clique-perfect and K-perfect graphs*, Ars Combin. **80** (2006), pp. 97–112.

[6] de Mello, C. P., A. Morgana and M. Liverani, *The clique operator on graphs with few P_4's*, Discrete Appl. Math. **154** (2006), pp. 485–492.

[7] Escalante, F., *Über iterierte Clique-Graphen*, Abh. Math. Sem. Univ. Hamburg **39** (1973), pp. 59–68.

[8] Hamelink, R. C., *A partial characterization of clique graphs*, J. Combinatorial Theory **5** (1968), pp. 192–197.

[9] Hedetniemi, S. T. and P. J. Slater, *Line graphs of triangleless graphs and iterated clique graphs*, Springer Lecture Notes in Math. **303** (1972), pp. 139–147.

[10] Kahle, M., *Topology of random clique complexes*, preprint, arXiv:math/0605536v2 [math.CO].

[11] Larrión, F., V. Neumann-Lara and M. A. Pizaña, *On the homotopy type of clique graphs*, J. of the Brazilian Comp. Soc. **7** (2002), pp. 69–73.

[12] Larrión, F., V. Neumann-Lara and M. A. Pizaña, *Graph relations, clique divergence and surface triangulations*, J. Graph Theory **51** (2006), pp. 110–122.

[13] Neumann-Lara, V., *On clique-divergent graphs*, Problèmes Combinatoires et Théorie des Graphes. (Colloques internationaux du C.N.R.S., Paris) **260** (1978), pp. 313–315.

[14] Prisner, E., *Convergence of iterated clique graphs*, Discrete Math. **103** (1992), pp. 199–207.

[15] Prisner, E., "Graph dynamics," Longman, Harlow, 1995, xii+233 pp.

[16] Requardt, M., *(Quantum) spacetime as a statistical geometry of lumps in random networks*, Classical Quantum Gravity **17** (2000), pp. 2029–2057.

[17] Requardt, M., *Space-time as an order-parameter manifold in random networks and the emergence of physical points*, in: *Quantum theory and symmetries (Goslar, 1999)*, World Sci. Publ., River Edge, NJ, 2000 pp. 555–561.

[18] Requardt, M., *A geometric renormalization group in discrete quantum spacetime*, J. Math. Phys. **44** (2003), pp. 5588–5615.

[19] Roberts, F. S. and J. H. Spencer, *A characterization of clique graphs*, J. Combinatorial Theory Ser. B **10** (1971), pp. 102–108.

[20] Spanier, E. H., "Algebraic topology," Springer-Verlag, New York, 1981, xvi+528 pp., corrected reprint.

[21] Spencer, J., "The strange logic of random graphs," Algorithms and Combinatorics **22**, Springer-Verlag, Berlin, 2001, x+168 pp.

[22] Szwarcfiter, J. L., *A survey on clique graphs*, in: B. A. Reed and C. Linhares-Sales, editors, *Recent advances in algorithms and combinatorics*, CMS Books Math./Ouvrages Math. SMC **11**, Springer, New York, 2003 pp. 109–136.

$2K_2$ vertex-set partition into nonempty parts [1]

Simone Dantas [a,2], Elaine M. Eschen [b,3], Luerbio Faria [c,2],
Celina M. H. de Figueiredo [d,2], Sulamita Klein [e,2]

[a] *IM, Universidade Federal Fluminense, Brazil.*

[b] *Lane Department of Computer Science and Electrical Engineering, West Virginia University, USA.*

[c] *FFP, Universidade do Estado do Rio de Janeiro, Brazil.*

[d] *COPPE, Universidade Federal do Rio de Janeiro, Brazil.*

[e] *IM and COPPE, Universidade Federal do Rio de Janeiro, Brazil.*

Abstract

A graph is $2K_2$-partitionable if its vertex set can be partitioned into four nonempty parts A, B, C, D such that each vertex of A is adjacent to each vertex of B, and each vertex of C is adjacent to each vertex of D. Determining whether an arbitrary graph is $2K_2$-partitionable is the only vertex-set partition problem into four nonempty parts according to external constraints whose computational complexity is open. We show that for C_4-free graphs, circular-arc graphs, spiders, P_4-sparse graphs, and bipartite graphs the $2K_2$-partition problem can be solved in polynomial time.

Keywords: Structural graph theory, Computational difficulty of problems, Analysis of algorithms and problem complexity.

[1] Partially supported by Brazilian agencies CNPq 470420/2004-9 and FAPERJ.

[2] Email:sdantas,luerbio,celina,sula@cos.ufrj.br

[3] Acknowledges support from WV EPSCoR and WVU Office of International Programs; Email:Elaine.Eschen@mail.wvu.edu

1 Introduction

The problem of partitioning the vertex set of a graph subject to a given set of constraints on adjacencies between vertices in two distinct parts, or among vertices within a part, is a fundamental problem in algorithmic graph theory. For example, k-colorability and finding a skew partition, clique or stable cutset, or homogeneous set. A *partition decision problem* asks if a given graph admits a specific partition.

For a given symmetric $k \times k$ matrix M over $\{0, 1, *\}$, an *M-partition* [7] is a partition of the vertex set of a graph into k parts (*empty parts are permitted*), corresponding to the rows and columns of M, such that: for $i \neq j$, if $M[i, j] = 0$ (resp., 1, $\star$), then 'no edges' (resp., 'all edges', 'no restriction') are required between vertices in part i and vertices in part j; if $M[i, i] = 0$ (resp., 1, $\star$), then part i is required to induce a stable set (resp., clique, arbitrary subgraph). Thus, for any specific matrix M, we have an M-PARTITION decision problem. In a LIST M-PARTITION problem, in addition, the input includes for each vertex v a nonempty list $\mathcal{L}(v) \subseteq \{1, 2, \ldots, k\}$. The problem asks: "Does G admit an M-partition in which each vertex v is assigned to a part in $\mathcal{L}(v)$?". For a given matrix M, a polynomial-time algorithm for LIST M-PARTITION implies a polynomial-time algorithm for M-PARTITION with the additional constraint that all parts be nonempty, via solving $\mathrm{O}(n^4)$ instances of LIST M-PARTITION.

A *$2K_2$-partition* is a partition of the vertex set of graph into four *nonempty* parts A, B, C, D such that each vertex of A is adjacent to each vertex of B, and each vertex of C is adjacent to each vertex of D. The $2K_2$-PARTITION problem can be formulated as an M-partition problem with the additional requirement that all parts be nonempty.

Every list M-partition problem with M of dimension 4 was classified by Feder et al. [7] as either solvable in quasi-polynomial time or NP-complete. In particular, LIST M-PARTITION, where M is the $2K_2$-partition matrix, was classified as NP-complete [4]. Later, Cameron et al. [1] showed that all the quasi-polynomial-time cases, except one problem and its complement, of the Feder et al. [7] quasi-dichotomy result are actually polynomial-time solvable.

Dantas et al. [3] studied *H-partitions*, where the matrix M has dimension 4 and only $*$s on the main diagonal (i.e., no internal constraints are imposed), and all parts must be nonempty. All the H-partition problems have been shown to be polynomial-time solvable [1,3,9,7], except $2K_2$-PARTITION.

The NONEMPTY-PART LIST $2K_2$-PARTITION problem takes as input a graph G and four vertices x_a, x_b, x_c, $x_d \in V(G)$, and asks whether G admits a $2K_2$-partition such that $x_a \in A$, $x_b \in B$, $x_c \in C$ and $x_d \in D$. Campos et al. [2] proved that this case of LIST M-PARTITION, where M is the $2K_2$-partition matrix, and in which input lists have restricted form remains NP-complete. Therefore, a polynomial-time algorithm for $2K_2$-PARTITION is not immediately obtained by solving $\mathrm{O}(n^4)$ instances of LIST M-PARTITION.

Feder et al. [6] have shown that all list M-partition problems on cographs admit polynomial-time algorithms. Further recent work on list M-partitions of chordal graphs [8] and perfect graphs [5] leaves the complexity of LIST M-PARTITION, where M is the $2K_2$-partition matrix, open for these special classes.

We solve $2K_2$-PARTITION efficiently for the graph classes: C_4-free, spiders (which can have arbitrary induced subgraphs), P_4-sparse (which properly contain cographs), circular-arc, and bipartite.

2 Universal pairs

Let $G = (V, E)$ be a simple graph. An *edge universal* vertex u is such that every edge in E is incident to u. A graph is not $2K_2$-partitionable if it contains an edge universal vertex. Also, a graph with isolated vertices, $|V| \leq 3$, or $|E| \leq 1$ is not $2K_2$-partitionable. So, we assume that none of these cases hold.

A *universal pair* is pair of vertices u and v such that $N(u) \cup N(v) \supseteq V \setminus \{u, v\}$ and there exists distinct vertices u' and v' in $V \setminus \{u, v\}$ such that u is adjacent to u' and v is adjacent to v'. Here, $N(u) = \{w \in V | uw \in E\}$.

Lemma 2.1 *Let G be a graph. If G has a universal pair of vertices, then G is $2K_2$-partitionable.*

Indeed, the existence of a universal pair is a special case of the definition of $2K_2$-partitionable in which two parts are singleton sets. In the sequel, we identify several classes of graphs (C_4-free, circular-arc, spiders, connected P_4-sparse) for which the Yes answer to $2K_2$-PARTITION is equivalent to the existence of a universal pair. Consequently, $2K_2$-PARTITION is polynomial-time solvable for these classes of graphs. In addition, we establish the polynomial-time solvability of $2K_2$-PARTITION for bipartite graphs, for which this equivalency does not hold.

3 C_4-free graphs and circular-arc graphs

Theorem 3.1 *Let G be a C_4-free graph. G is $2K_2$-partitionable if and only if G has a universal pair of vertices.*

Proof. For any $2K_2$-partition of G one part of each pair $\{A, B\}$ and $\{C, D\}$ is a clique. Thus, G has a universal pair. □

A graph G is a *proper circular-arc graph* if G is a circular-arc graph and G has a circular-arc model where no arc properly contains another arc.

Theorem 3.2 *Let G be a circular-arc graph. G is $2K_2$-partitionable if and only if G has a universal pair.*

Proof. We shall prove by induction that if G is a $2K_2$-partitionable circular-arc graph, then G has a universal pair. Let G be a circular-arc graph, and assume a fixed $2K_2$-partition $\mathcal{P}$ of G. We have the following base cases:
Case 1. G is a proper circular-arc graph and C_4-free.
In this case, G has a universal pair by Theorem 3.1.
Case 2. G is a proper circular-arc graph and not C_4-free.
Consider a proper arc model for G. It is easy to see that any pair of nonadjacent vertices in a C_4 of G form a universal pair.
Case 3. The $2K_2$-partition $\mathcal{P}$ of G has a singleton part.
This is the case for any graph with $4 \leq |V| \leq 7$. Suppose part $A = \{v\}$. If part C or D is a clique, then G has a universal pair. If neither C nor D is a clique, then G has a C_4 with 2 vertices in D and 2 vertices in C. One vertex w of this C_4 must see v; place w in part B. Repeat until either C or D is a clique. Then, G has a universal pair.

Now suppose $|V| \geq 8$ and none of Cases 1, 2, and 3 apply to G and the given $2K_2$-partition $\mathcal{P}$. In this case, in any circular-arc model for G there is an arc v that properly contains an arc u. The graph $G - v$ has a $2K_2$-partition formed by removing v from $\mathcal{P}$. Thus, by the induction hypothesis, $G - v$ has a universal pair, and this pair is a universal pair in G as well. □

4 Spiders and P_4-sparse graphs

A graph is *P_4-sparse* if every set of five vertices induces at most one P_4 [10]. This class strictly contains the class of *cographs*, graphs that do not contain a P_4. A graph G is a *spider* if the vertex set V admits a partition into sets S, K and R such that: S is a stable set, K is a clique and $|S| = |K| \geq 2$; and each vertex in R is adjacent to each vertex in K and nonadjacent to each vertex in S. In addition, in a *thin spider* every vertex of K has exactly one private (with respect to K) neighbor in S, and in a *thick spider* every vertex of K has exactly one private (with respect to K) non-neighbor in S. A spider with $|S| = |K| = 2$, is considered to be thick. Every spider contains a *special* P_4 with endpoints in S and midpoints in K.

Theorem 4.1 *[11] A graph G is P_4-sparse if and only if for every induced subgraph H of G with at least two vertices, exactly one of the following con-*

ditions is satisfied: H is disconnected, $\overline{H}$ is disconnected, or H is isomorphic to a spider.

Theorem 4.2 *Let G be a disconnected graph. G is $2K_2$-partitionable if and only if G has exactly two connected components and each component of G is complement disconnected.*

Note in this case, it is possible that G is $2K_2$-partitionable and there is no universal pair; for example, take each connected component of G to be a C_4.

Theorem 4.3 *Let G be a complement disconnected graph. G is $2K_2$-partitionable if and only if G has a universal pair.*

Theorem 4.4 *Let G be a spider. G is $2K_2$-partitionable if and only if G is a thick spider.*

Proof. If G is a thick spider, any two vertices in K form a universal pair. Suppose G is a thin spider. Consider all the possible ways of placing the vertices of a special P_4 with endpoints in S and midpoints in K. In every case, a $2K_2$-partition cannot be completed. □

Corollary 4.5 *Let G be a spider. G is $2K_2$-partitionable if and only if G has a universal pair.*

It is easy to distinguish thin and thick spiders and find K in a thick spider by examining vertex degrees.

Corollary 4.6 *Given a P_4-sparse graph G, it can be decided in polynomial time whether G has a $2K_2$-partition. If the answer is Yes, the partition can also be found in polynomial time.*

Proof. Consider the three possible conditions given by Theorem 4.1 for G.□

5 Bipartite graphs

Two edges of a graph G are *separable* if they induce a $2K_2$ in G.

Theorem 5.1 *Given a bipartite graph $G = (X, Y, E)$, it can be decided in polynomial time whether G has a $2K_2$-partition. If the answer is Yes, the partition can also be found in polynomial time.*

Proof. When G is disconnected we refer to Theorem 4.2. $|X| \geq 2$ and $|Y| \geq 2$, else G is not $2K_2$-partitionable.
Case 1. G contains a separable pair of edges xy, $x'y'$.

Place x in part A and x' in part D, y in part B and y' in part C. Now the problem reduces to an instance of 2-SAT.
Case 2. G does not contain a separable pair of edges.
In this case, for any pair of vertices x_i, x_j, $N(x_i) \subseteq N(x_j)$ or $N(x_j) \subseteq N(x_i)$. Order the vertices of X by $|N(x_i)|$ from largest to smallest: $|N(x_1)| \geq |N(x_2)| \geq \ldots \geq |N(x_s)|$. If G is complete bipartite, G is $2K_2$-partitionable. Otherwise, note that $N(x_s) \neq Y$, and a $2K_2$-partition is obtained by setting: $A = \{x_1\}$, $B = Y \setminus N(x_s)$, $C = N(x_s)$, $D = X \setminus \{x_1\}$. □

Note a bipartite graph G may be $2K_2$-partitionable and there is no universal pair; for example, the case where G is two C_4s joined by a single edge.

References

[1] K. Cameron, E.M. Eschen, C.T. Hoàng, R. Sritharan. The list partition problem for graphs In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 391–399, 2004.

[2] C.N. Campos, S. Dantas, L. Faria, and S. Gravier, $2K_2$-partition problem, In 7th International Colloquium on Graph Theory. Electronic Notes on Discrete Mathematics 22 (2005), 217–221. Submitted to Discrete Applied Mathematics.

[3] S. Dantas, C.M.H. de Figueiredo, S. Gravier, and S. Klein, Finding H-partition problems, RAIRO-Theoretical Informatics and Applications 39 (2005), 133–144.

[4] T. Feder, P. Hell, List homomorphisms to reflexive graphs, J. Combinatorial Theory B 72 (1998), 236–250.

[5] T. Feder, P. Hell, Matrix partitions of perfect graphs, Discrete Math. 306 (2006), 2450–2460.

[6] T. Feder, P. Hell, W. Hochstattler, Generalized colourings (matrix partitions) of cographs, In Graph theory in Paris, 149–167, Trends Math., Birkhuser, Basel, 2007.

[7] T. Feder, P. Hell, S. Klein, and R. Motwani, List partitions, SIAM J. Discrete Math. 16 (2003), 449–478.

[8] T. Feder, P. Hell, S. Klein, L.T. Nogueira, and F. Protti, List matrix partitions of chordal graphs, Theoretical Computer Science 349 (2005) 52–66.

[9] C.M.H. de Figueiredo, S. Klein, Y. Kohayakawa, and B.A. Reed, Finding skew partitions eficiently, Journal of Algorithms, 37 (2000), 505–521.

[10] C.T. Hoàng, A class of perfect graphs, M.Sc. Thesis, McGill University, Montreal, (1983).

[11] B. Jamison and S. Olariu, A tree representation for P_4-sparse graphs, Discrete Appl. Math. 35 (1992), 115–129.

A Priori and A Posteriori Aggregation Procedures to Reduce Model Size in MIP Mine Planning Models

Andres Weintraub[1] Marianela Pereira[2] Ximena Schultz[3]

Department of Industrial Engineering
University of Chile
Santiago, Chile

Abstract

Mine planning models have proved to be very effective in supporting decisions on sequencing the extraction of material in copper mines. These models have been developed for CODELCO, the Chilean state copper mine and used successfully. Here, we wish to develop a corporate model, including all mines of CODELCO. The original models are extremely large MIPs. In order to run a global model, the original models need to be reduced significantly. We develop an approach to aggregate the models. The aggregation is done both on the original data of the mine as well as on the MIP original models. The aggregation is based on clustering analysis. Promising results were obtained with data of a large underground mine.

Keywords: Mining, Planning Models, Aggregation Procedures.

[1] Email: aweintra@dii.uchile.cl

[2] Email: marianela.pereira@coe.ubc.ca

[3] Email: xschultz@ing.uchile.cl

1 Introduction

Models have been developed for planning in copper mines for CODELCO, the state owned mine corporation, one of the main world producers of copper. CODELCO owns seven mines, both open pit and underground. In the planning process, detailed models are used. Consider the largest underground mine, El Teniente. The mine is divided into sectors and sub-sectors. Each sector is divided into columns of extraction points, we call PEXTs. These columns are composed of blocks, usually of 20 by 20 by 20 meters, called UBCs, which are the basic units of extraction. The planning process considers a 25 year horizon, and the sequence to follow points of extraction is given. The decisions to be made are the period (yearly) in which each PEXT (or column) is started, and how many UBCs will be extracted. Typically the % of copper decreases as the extraction goes up the column, so it often is more convenient to leave some UBCs not extracted and move to the next PEXT. The extraction is carried out through gravity, using explosives to loosen the material, which is then carried through a transportation system to downstream plants. In each PEXT then, the lower UBC is extracted first and the extraction follows upward. A MIP model was developed to support decisions for planning extraction of UBCs, and leads to decisions of exact timing for each UBC. Using the model to support decisions increased the value of the mine over 100 million dollars. Similar models have been developed for open pit mines, mostly in northern Chile. Again, these are large scale MIP models (Epstein et al. 2003 [3], Caro et al. 2007 [2]).

We are interested in developing a model which integrates all mines for corporate decisions, to determine extraction from each sector, in each mine, each period. In particular, given price uncertainties the corporation might want to develop a stochastic model to incorporate these uncertainties. In this case, we will consider a 5 year horizon model.

The original models are too large, too slow to run to include them directly in such a corporate model. Also, at corporate level there is no need to define in such detail the decisions. While at one level, planning for each mine is carried out in detail, and given the very large amounts involved, this planning is a vital issue by itself, there is also need to consider the whole enterprise. CODELCO owns seven mines, between underground and open pit. Some, very large like the one we are presenting in this study. At the corporation level there are global constraints that are not present at mine level. CODELCO produces a significant percentage of the world copper, thus its actions influence the market. So, the corporation will need a strategy on how much copper it wants to produce as a sum of all its mines. It is also important that the model corresponding to each mine runs fast. We are considering developing an approach to incorporate prize uncertainty into the decision process. We want to develop approaches, like robust programming (Berktsekas et al. 2003 [1]) or coordinated branching (Alonso et al. 2003 [4]) to incorporate prize uncertainty. Any of these approaches requires running each of the mine problems multiple times, hence the need for fast solutions.

For this reason it is convenient to consider aggregate models. We will use cluster analysis to do the aggregation. The advantage of rigorous aggregation processes is in the preservation of feasibility once the aggregate solution is disaggregated to detailed level. Two types of aggregations will be considered. One, *a priori*, which will aggregate the UBCs, and *a posteriori*, which will aggregate columns of the original model.

2 The *A Priori* Aggregation

We consider first the a priori aggregation. We need to aggregate UBCs that are connected to each other and in such a way that it is possible to extract each cluster independently from the others. In this way we can assign one extraction variable to each cluster. We need to group the UBCs according to similarities. We chose basic characteristics that describe UBCs to establish a way to measure the dissimilarity between two UBCs. These are Tonnage, Percentage of Copper, Percentage of Molybdenum and Speed of Extraction, which best describe a UBC. Each characteristic has a different importance, so a set of weights associated to the characteristics was defined. This was done with support of a knowledgeable source.

The dissimilarity measurement between two UBCs is defined as the sum of the difference between the numerical values of the characteristics of both UBCs divided by the average value taken by the characteristic through all UBCs, multiplied by the weight of each characteristic. Then, the dissimilarity measurement of a cluster is defined as the sum of the dissimilarity measure between every pair of UBCs in the same cluster.

In order to group UBCs into clusters we developed a K-means type algorithm. To use this type of algorithm an initial partition was created, based on a greedy selection of clusters according to similarities and spatial location. Note the importance of the spatial location, as clusters defined must be feasible to be extracted independently.

As an output a parameter, delta, is given, corresponding to the total dissimilarity of the partition. This is the sum of the total dissimilarity of each cluster.

After the initial step a the K-means algorithm is applied. It is based on exchanging UBCs between clusters in order to reduce the value of delta. Combinations of exchanges are tested to find those that reduce the value of delta. If no improving exchange is found, the algorithm stops.

Note that only exchanges that create feasible clusters are considered.

The value of each characteristic in every cluster . These are calculated as a weighted average proportional to the tonnage of each UBC.

Once the clusterization process was finished, the aggregate model was created.

The main variables relate to extraction of clusters from subsectors in given time periods (0-1) and the tonnages of copper and molybdenum extracted from each sector and period.

The constraints that need to be satisfied are: Each cluster can be extracted only once, satisfy the defined sequence of extractions, the allowable speed and capacity of extraction, as well as conservation of flows and logical relationships between variables. The objective function is to maximize the profit. The costs are those of extraction, activation of a PEXT, the cost due to increase or decrease of production between periods, and the transportation costs. The benefits are related to sales of copper and molybdenum.

We then solved the aggregate model. The solution was fed into the original model by indicating productions by sector, so we could compare the solution of the original model with the solution of the detailed model where productions by sector and period follow the aggregated model solution. This was implemented in the underground mine of El Teniente. A five year model was created, that considered the eleven sectors that could be extracted in the next 5 years.

When comparing the original model with the disaggregation of the aggregate model, the percentage error in the value of the objective function was 3,62%. The reduction of execution time was of 73,68%. The model dimension was reduced by 90%.

So the solutions found are coherent with the original model and the reduction of time is significant. The reduction of time will be more significant when the model is larger (more periods). Note that when the number of periods goes to 25 years, as in the case of the models presently used, solving the original model takes two hours of CPU.

3 The *A Posteriori* Approach

This approach consists in aggregating the original mine planning model through clustering techniques.

In order to develop this aggregated model we use *a posteriori* aggregation. This type of aggregation procedures is based on the original standard linear problem expressed in matrix form $Max\{Z = cx : Ax \leq b,\ x \geq 0\}$, where A is a $m \times n$ matrix; c, $x \in \Re^n$, and $b \in \Re^m$.

Again cluster analysis is used for the aggregation process based on similarities. Each column is associated to a variable of the problem. We used a column aggregation procedure proposed by Zipkin, 1980 [6]:

- We consider a partition in subsets of K columns. Therefore $\sigma = \{S_k : k = 1, 2, ..., K\}$, is a *partition* of $\{1, 2, 3, ..., n\}$ where S_k has n_k elements.
- To obtain the model parameters we use a method called fixed-weight combination. It involves a convex weighting of a cluster of elements of cardinality n_k by an n_k-vector g_k, whose components are nonnegative and sum to unity.
- Let $A'_k = A_k g_k$, $c'_k = c_k g_k$, $x'_k = x_k g_k$, $\quad k = 1, ..., K$

  $c' = (c'_1, c'_2, ..., c'_k)$, $A' = (A'_1, A'_2, ..., A'_k)$, and $x' = (x'_1, x'_2, ..., x'_k)$.

  Then, the aggregate problem AP is $Max\{Z' = c'x' : A'x' \leq b,\ x' \geq 0\}$.

In order to give the same importance to the values of the columns these were normalized and then components were normalized again within columns, to give an adequate importance to its values, for example, the component of the speed of extraction constraint is less important for the aggregation criteria than the X position of the UBC.

The similarity criteria defined was: $1 - cosin(B, C)$, where B y C are two modified vectors of the A matrix corresponding to the extraction variables and $cosin(X, Y) = \frac{\langle X,Y \rangle}{\|X\| \cdot \|Y\|}$.

The clustering method used was a modification of the Hartigan's Leader Type Method [5]:. This procedure was chosen due to the low computational effort required and reasonable quality of solution.

After the aggregation process is carried out, we can solve the smaller aggregate problem, using a commercial code.

This type of aggregation is highly sensitive to the weights g_k used. The aggregation and weights used considered the spatial characteristics of the UBCs to insure feasible solutions to the aggregate problem.

The disaggregation process uses the vector g_k to obtain results of the disaggregate problem, using the fixed weights. Feasibility is assured by this process

Finally, Zipkin provides for this method an a posteriori error bound for the objective function using data from the original problem and the solution of the aggregate problem [6], but does not need information from the solution of the original problem so, once the aggregate problem is solved, it is easy to obtain an a posteriori bound. Results show that the amount of variables was reduced to around 15% of the number of variables of the original problem. The model was solved in 27 seconds, a 88% solve time reduction.

The error bound obtained was 3% which is very similar to the real error of 2.93%.

This aggregation procedure had difficulties in defining aggregations and weights that insured feasible solutions, but finally led to reasonable good solutions.

In conclusion, it appears that the proposed cluster analysis approach , in the a priori and a posteriori cases, can lead to reduced corporate models, which approximate reasonably well the original large scale, detailed models. In this form, we can determine smaller models, which run in reduced CPU times, as needed for corporate models, where each mine model needs to be run several times. Further research is needed to integrate both types of aggregation.

References

[1] Bertsimas and Sim, *"Robust Discrete Optimization and Network Flows"*, Mathematical Programming, **98(1-3):49–71**, (2003).

[2] Caro, R., R. Epstein, P. Santibañez, and A. Weintraub, *"Handbook of Operation Research in Natural Resources,"* Chapter **32**, Springer Science (2007).

[3] Epstein, R., S. Gaete, F. Caro, A. Weintraub, P. Santibañez, and J. Catalan *Optimizing Long Term Planning for Underground Copper Mines*, International Copper Conference (2003).

[4] Alonso-Ayuso A, Escudero L.F., Teresa Ortuno M., *"Branch-and-Fix Coordination algorithmic framework for solving some types of stochastic pure and mixed 0-1 programs"* European Journal of Operational Research, **151:503-519**, (2003).

[5] Hartigan, J. *Clustering Algorithms*, Wiley New York **686 p-**.

[6] Zipkin, P. *Bounds on the Effect of Aggregating Variables in Linear Programs*, Operations Research **28** (1980), 1299–1309.

Combinatorial flexibility problems and their computational complexity[1]

Néstor E. Aguilera

CONICET and UNL. IMAL/INTEC, Güemes 3450, 3000 Santa Fe, Argentina.

Valeria A. Leoni[2], Graciela L. Nasini

CONICET and UNR. Depto de Matemática, Fac. de Cs. Exactas, Ingeniería y Agrimensura, Av. Pellegrini 250, 2000 Rosario, Argentina.

Abstract

The concept of flexibility—originated in the context of heat exchanger networks—is associated with a substructure which guarantees the *performance* of the original structure, in a given range of possible states. We extend this concept to combinatorial optimization problems, and prove several computational complexity results in this new framework.

Under some monotonicity conditions, we prove that a combinatorial optimization problem polynomially transforms to its associated flexibility problem, but that the converse need not be true.

In order to obtain polynomial flexibility problems, we have to restrict ourselves to combinatorial optimization problems on matroids. We also prove that, when relaxing in different ways the matroid structure, the flexibility problems become *NP*-complete. This fact is shown by proving the *NP*-completeness of the flexibility problems associated with the Shortest Path, Minimum Cut and Weighted Matching problems.

Keywords: combinatorial problems, flexibility, computational complexity

1 Flexibility and combinatorial optimization problems

The concept of flexibility arose from chemical engineering problems in the design of heat exchanger networks (see, for example [2]). This concept is associated with a substructure which guarantees the *performance* of the original structure in a given range of possible states. In the context of heat exchanger networks, this performance is defined by the value of a Maximum Flow-Minimum Cut. This problem motivated us to extend the flexibility concept to general combinatorial optimization problems.

Following [5], in a combinatorial optimization problem we deal with a finite set A, a vector of costs $\mathbf{c} \in \mathbb{R}^A$ and a family $\mathcal{F}(A)$ of subsets of A. The cost of $F \subseteq A$ will be indicated by $\mathbf{c}\,(F) = \sum_{a \in F} c_a$, and the optimal value by

$$\xi_A^{\mathcal{F}}(\mathbf{c}) = \min(\max)\left\{\mathbf{c}(F) : F \in \mathcal{F}(A)\right\}.$$

If $\mathcal{F}(A) = \emptyset$, we set $\xi_A^{\mathcal{F}}(\mathbf{c}) = +\infty$ and $\xi_A^{\mathcal{F}}(\mathbf{c}) = -\infty$, respectively. We say that $F \in \mathcal{F}(A)$ is $\mathbf{c}$-*optimal* if $\mathbf{c}(F) = \xi_A^{\mathcal{F}}(\mathbf{c})$.

We will identify a particular combinatorial optimization problem by the oracle algorithm $\mathcal{F}$ which decides, in constant time, whether a given subset of A belongs to the family $\mathcal{F}(A)$.

Working on flexibility problems, we consider a family of instances determined by a given set A and vectors $\mathbf{c}^-, \mathbf{c}^+ \in \mathbb{Z}_+^A$, defining the *state set* $S = \left\{\mathbf{c} \in \mathbb{R}^A : \mathbf{c}^- \leq \mathbf{c} \leq \mathbf{c}^+\right\}$. We also consider a *substructure* given by $B \subseteq A$. For simplicity we will indicate by $\xi_B^{\mathcal{F}}(\mathbf{c})$, the optimal value corresponding to the restriction of $\mathbf{c}$ to $\mathbb{R}^B$.

Given a state set S and $W \subseteq S$, we will say that B is $\mathcal{F}$-*flexible* in W if $\xi_B^{\mathcal{F}}(\mathbf{c}) = \xi_A^{\mathcal{F}}(\mathbf{c})$, for all $\mathbf{c} \in W$. When $W = S$, we just say that B is $\mathcal{F}$-*flexible*.

The $\mathcal{F}$-*flexibility problem* ($\mathcal{F}$-flex) is formulated as follows:

INSTANCE: A finite set A; $B \subseteq A$; $\mathbf{c}^+, \mathbf{c}^- \in \mathbb{Z}_+^A$.
QUESTION: Is there $\mathbf{c}$ with $\mathbf{c}^- \leq \mathbf{c} \leq \mathbf{c}^+$ and $\xi_A^{\mathcal{F}}(\mathbf{c}) \neq \xi_B^{\mathcal{F}}(\mathbf{c})$?

Notice that $\mathcal{F}$-flex consists of answering whether B is not $\mathcal{F}$-flexible.

From now on, we restrict ourselves to combinatorial optimization problems verifying some kind of monotonicity under inclusion on the optimal value. We say that $\mathcal{F}$ is *monotone increasing (decreasing)* if for all $\mathbf{c} \in \mathbb{R}^A$ and $B \subseteq A$ we have $\xi_B^{\mathcal{F}}(\mathbf{c}) \leq \xi_A^{\mathcal{F}}(\mathbf{c})$ ($\xi_B^{\mathcal{F}}(\mathbf{c}) \geq \xi_A^{\mathcal{F}}(\mathbf{c})$).

In fact, most of the combinatorial optimization problems of interest are monotone. In particular, between the problems considered here, the *Weighted*

[1] Partially supported by grants of CONICET-PIP 5810 and ANPCyT-PICT2005 38036

[2] Email: valeoni@fceia.unr.edu.ar

Matching problem and the *Maximum Weight Forest problem* are monotone increasing maximization problems, whereas the *Shortest Path problem* is a monotone increasing minimization problem and the *Minimum Cut problem* is monotone decreasing.

The monotonicity imposed on the optimal value leads us to prove that solving $\mathcal{F}$-flex is equivalent to asking whether a given element $a \in A$ is *useful* for S, following the terminology introduced in [1] in the context of the Maximum Flow problem. Formally, we prove:

Lemma 1.1 *$\mathcal{F}$-flex may be reduced to the family of instances given by $B = A \setminus \{a\}$, for $a \in A$.*

The key of the proof is the fact that the answer corresponding to an instance of $\mathcal{F}$-flex given by $(A, B, \mathbf{c}^+, \mathbf{c}^-)$ is YES if and only if, for any B' with $B \subseteq B' \subseteq A$, at least one of the instances $(A, B', \mathbf{c}^+, \mathbf{c}^-)$ and $(B', B, \mathbf{c}^+, \mathbf{c}^-)$ has also an affirmative answer.

In the next section we show that solving $\mathcal{F}$-flex is always at least as hard as solving $\mathcal{F}$, and try to find families of optimization problems with polynomial complexity associated flexibility problems.

The proofs of all computational complexity results are strongly based on the fact that, when checking flexibility, it is enough to do so on a *finite* subset of states, called *test set*. In particular, for $F \subseteq A$ we define the *F-state* $\mathbf{c}^F \in \mathbb{Z}_+^A$ by:

- if $\mathcal{F}$ is a minimization problem, $c_a^F = c_a^-$ if $a \in F$ and $c_a^F = c_a^+$ if $a \notin F$;
- if $\mathcal{F}$ is a maximization problem, $c_a^F = c_a^+$ if $a \in F$ and $c_a^F = c_a^-$ if $a \notin F$.

We prove that, for a given $F \in \mathcal{F}(A)$, the F-state is that state for which F has the greatest possibility of being optimal in the sense that, if F is a $\mathbf{c}$-optimal element for some $\mathbf{c} \in S$, then F is $\mathbf{c}^F$-optimal.

Moreover:

Lemma 1.2 *Let A, $B \subseteq A$, $\mathbf{c}^-$ and $\mathbf{c}^+$ defining an instance of $\mathcal{F}$-flex. If $\mathcal{F}$ is a monotone increasing minimization problem or a monotone decreasing maximization problem, then $\{\mathbf{c}^F : F \in \mathcal{F}(A)\}$ is a test set. If $\mathcal{F}$ is a monotone increasing maximization problem or a monotone decreasing minimization problem, then $\{\mathbf{c}^F : F \in \mathcal{F}(B)\}$ is a test set.*

Let us point out that, for a general combinatorial optimization problem $\mathcal{F}$, the cardinalities of the test sets given by Lemma 1.2 are non polynomial in the size of A.

2 Looking for polynomial flexibility problems

In order to compare the computational complexities of $\mathcal{F}$ and $\mathcal{F}$-flex, we need to take into account the relationship between $\mathcal{F}(B)$ and $\mathcal{F}(A)$, when $B \subseteq A$. Let us impose the following monotonicity conditions on the set of feasible solutions

(P1) for every $E \in \mathcal{F}(B)$, there exists $F \in \mathcal{F}(A)$ such that $E \subseteq F$,

(P2) if $E \in \mathcal{F}(A)$ and $E \subseteq B$ then $E \in \mathcal{F}(B)$.

Under these conditions we can prove:

Theorem 2.1 *$\mathcal{F}$ may be polynomially reduced to its corresponding flexibility problem, $\mathcal{F}$-flex.*

Let us observe that, for the Weighted Matching, the Maximum Weight Forest and the Shortest Path problems, it holds that

(P3) $\mathcal{F}(B) = \{F \in \mathcal{F}(A) : F \subseteq B\}$,

whereas, for the Minimum Cut problem it holds that

(P4) $\mathcal{F}(B) = \{F \cap B : F \in \mathcal{F}(A)\}$.

Each of the properties (P3) and (P4) imply (P1) and (P2). Conditions (P1) and (P2) are the weakest we may impose for proving Theorem 2.1. In fact, when finding the optimal value of $\mathcal{F}$ we just use (P1), and then (P2) is necessary in order to find an optimal element.

From Theorem 2.1, if we want to find polynomial flexibility problems, we should reduce our search to the family of polynomial optimization problems.

An *exchanger network* is a digraph $D = (V_1 \cup V_2, E)$ with $V_1 \cap V_2 = \emptyset$, $E \subseteq V_1 \times V_2$ and a vector $\mathbf{c} \in \mathbb{R}^{V_1 \cup V_2}$ (for $i \in V_1$, c_i is the *supply* of i and for $j \in V_2$, c_j is the *demand* of j). The maximum *exchange* in this class of networks can be modeled as a Maximum Flow-Minimum Cut problem in a certain st-network.

In [1] and [4] we may find two independent proofs of the *NP*-completeness of the Minimum Cut flexibility problem (*FF*), even on instances with $\mathbf{c}^- = \mathbf{0}$. However, considering those instances of *FF* corresponding to exchanger networks (*FT*), the problem becomes polynomial when $\mathbf{c}^- = \mathbf{0}$.

Nevertheless,

Theorem 2.2 *FT is NP-complete.*

The proof is based on the reduction of the Balanced Complete Bipartite Graph problem (*BCBG*). *BCBG* consists in deciding whether there is a complete bipartite balanced subgraph of certain size in a given bipartite graph. The proof of its *NP*-completeness may be found for example in [3, p. 196].

We wonder if tightening conditions (P1) and (P2)—imposing, for example (P3)—we may establish the converse of Theorem 2.1. However, we prove:

Theorem 2.3 *The Shortest Path flexibility problem (FP) is NP-complete.*

In this case, we reduce *DVDP2* to the Shortest Path problem. Given a digraph G and nodes s, r, t, w of G, *DVDP2* consists in deciding whether there exist vertex disjoint st- and rw-paths.

A combinatorial optimization problem $\mathcal{F}$ is *hereditary* if for all A, $F \in \mathcal{F}(A)$ and $F' \subseteq F$ it holds that $F' \in \mathcal{F}(A)$. Since we deal with non negative states, hereditary combinatorial optimization problems become relevant when they are maximization problems.

Once again, instances with $\mathbf{c}^- = \mathbf{0}$ lead us to guess that hereditary optimization problems could have "easy" associated flexibility problems.

Lemma 2.4 *Let $\mathcal{F}$ be a hereditary problem satisfying condition (P2), $\mathbf{c}^- = \mathbf{0}$ and $\mathbf{c}^+$ with $c_a^+ > 0$ for some $a \in A$. Then, deciding if a is useful can be done in constant time.*

The key of the proof is to show that $a \in A$ is useful if and only if $\{a\} \subseteq \mathcal{F}(A)$.

We now consider the Weighted Matching problem, which satisfies (P3) and is also *hereditary*:

Theorem 2.5 *The Weighted Matching flexibility problem (FBM) is NP-complete, even on instances corresponding to bipartite graphs.*

The proof is based on the reduction of *FP* to *FBM* by using some known transformation between shortest paths and maximum weighted matchings in bipartite graphs.

Finally, hereditary optimization problems defined on matroids seem to be the "best candidates" when looking for polynomial flexibility problems, because of the characterization of matroids through greedy algorithms. In this case, this assertion can be confirmed by the following result:

Theorem 2.6 *The Maximum Independent Set (in a matroid) flexibility problem is polynomial.*

For the proof we use some previous results which allow us to show that, for an instance given by a matroid A, $\{\mathbf{c}^{\{a\}} : a \in A\}$ is a test set.

Finally, let us observe that the family of matchings in a bipartite graph is the intersection of two matroids. Hence, Theorem 2.5 implies that:

Theorem 2.7 *The Two Matroid Intersection flexibility problem is NP-complete.*

This last result leads us to guess that a "matroid structure" is the "weakest" from which we can obtain polynomial flexibility problems.

References

[1] Biedl, T. C., B. Brejová and T. Viñar, *Simplifying Flow Networks. M. Nielsen and B. Rovan (Eds.)*, MFCS 2000, LNCS 1893. (2000), 192–201.

[2] Cerdá, J., M. R. Galli, N. Camussi and M. A. Isla, *Synthesis of Flexible Heat Exchanger Networks-I. Convex Networks*, Comp. Chem. Engng. **14** (1990), 197–211.

[3] Garey, M. R. and D. Johnson, "Computers and Intractability: A Guide to the Theory of *NP*-Completeness", W. H. Freeman & Co., 1979.

[4] Nasini, G. L., "Sobre la Flexibilidad en Problemas de Optimización", Tesis Doctoral, Universidad Nacional de Rosario, Argentina, 1997.

[5] Nemhauser G. and L. Wolsey, "Integer and Combinatorial Optimization", Wiley, 1988.