Mutual exclusion scheduling with interval graphs or related classes. Part II.

Frédéric Gardi [1]

Bouygues SA / DGITN / e-lab,
32 avenue Hoche, 75008 Paris, France

Abstract

This paper is the second part of a study devoted to the mutual exclusion scheduling problem. Given a simple and undirected graph G and an integer k, the problem is to find a minimum coloring of G such that each color is used at most k times. The cardinality of such a coloring is denoted by $\chi(G, k)$. When restricted to interval graphs or related classes like circular-arc graphs and tolerance graphs, the problem has some applications in workforce planning. Unfortunately, the problem is shown to be $\mathcal{NP}$-hard for interval graphs, even if k is a constant greater than or equal to four [H.L. Bodlaender and K. Jansen (1995). Restrictions of graph partition problems. Part I. Theoretical Computer Science 148, pp. 93–109]. In this paper, the problem is approached from a different point of view by studying a non-trivial and practical sufficient condition for optimality. In particular, the following proposition is demonstrated: if an interval graph G admits a coloring such that each color appears at least k times, then $\chi(G, k) = \lceil n/k \rceil$. This proposition is extended to several classes of graphs related to interval graphs. Moreover, all our proofs are constructive and provide efficient algorithms to solve the MES problem for these graphs, given a coloring satisfying the condition in input.

Key words: chromatic scheduling, workforce planning, bounded coloring, interval graphs, graph algorithms

1991 MSC: 05C15, 90B35, 68R10, 68Q25

Email address: `fgardi@bouygues.com` (Frédéric Gardi).

[1] On leave from the firm Experian-Prologia SAS, Marseille, France.

1 Introduction

1.1 Presentation of the problem

Here is a fundamental problem in scheduling theory: n tasks must be completed on k processors in the minimum time, with the constraint that some tasks can not be executed at the same time because they share the same resources. When all tasks have the same processing time, the problem in question has an elegant formulation in graph-theoretical terms. In effect, having defined a simple and undirected graph where each vertex represents one task and two vertices are adjacent if the corresponding tasks are in conflict, an optimal schedule of the n tasks on k processors corresponds exactly to a minimum coloring of the graph such that each color appears at most k times. This is the reason why Baker and Coffman [2] called Mutual Exclusion Scheduling (shortly MES) the following problem:

Mutual Exclusion Scheduling
Input: a simple and undirected graph $G = (V, E)$, a positive integer k;
Output: a minimum coloration of G where each color appears at most k times.

When k is a fixed parameter (i.e., a constant of the problem), the abbreviation k-MES shall be used to name the problem.

In spite of few positive results, MES is $\mathcal{NP}$-hard for the majority of classes of graphs studied, even for small fixed values of k. The problem is $\mathcal{NP}$-hard for complements of line-graphs (even for fixed $k \geq 3$) [7], for bipartite graphs and cographs [5], for interval graphs (even for fixed $k \geq 4$) [5], for complements of comparability graphs (even for fixed $k \geq 3$) [26], and for permutation graphs (even for fixed $k \geq 6$) [24]. To the best of our knowledge, the sole classes of graphs for which MES problem was proved to be polynomial-time solvable are split graphs [26,5], forests and trees [2,25], collections of disjoint cliques [35], complements of strongly chordal graphs [10] and of interval graphs [26,5], and bounded treewidth graphs [4].

In a previous paper [16], we have begun a detailed study of the mutual exclusion scheduling problem for interval graphs as well as for two extensions, namely circular-arc graphs and tolerance graphs. When restricted to these classes of graphs, the problem has some applications to workforce planning. Linear-time and space algorithms are presented to solve the MES problem restricted to proper interval graphs and to threshold graphs, and the case $k = 2$ for interval graphs. Besides, the problem is shown to be solvable in quadratic time and linear space for proper circular-arc graphs, as well as in linear time and space when $k = 2$ for these graphs. On the other hand, the result of Bodlaender and Jansen [5] is completed by establishing the $\mathcal{NP}$-hardness of the 3-MES problem for bounded tolerance graphs, even if any cycle of length greater than or equal to five in the graph has two chords.

Unfortunately, restrictions based on subclasses of interval graphs, circular-arc graphs or tolerance graphs do not cover enough cases in practice. In this paper, the problem is approached from a different point of view by studying a non-trivial and practical sufficient condition for optimality in mutual exclusion scheduling.

1.2 Preliminaries

Formally, a graph $G = (V, E)$ is an *interval graph* if to each vertex $v \in V$ can be associated an open interval I_v of the real line, such that two distinct vertices $u, v \in V$ are adjacent if and only if $I_u \cap I_v \neq \emptyset$. The family $\{I_v\}_{v \in V}$ is an *interval representation* of G. The *left and right endpoints* of the interval I_v are respectively denoted $l(I_v)$ and $r(I_v)$. The class of interval graphs coincide with the intersection of the classes of chordal graphs and of complements of comparability graphs [17]. A graph is chordal if it contains no induced cycle of length greater than or equal to four; chordal graphs are also known as the intersection graphs of subtrees in a tree [17]. Comparability graphs are the transitively orientable graphs, they correspond to graphs of partial orders.

Circular-arc graphs and tolerance graphs are two natural extensions of interval graphs. *Circular-arc graphs* are the intersection graphs of collections of arcs on a circle. A circular-arc graph $G = (V, E)$ admits a *circular-arc representation* $\{A_v\}_{v \in V}$ in which each arc A_v is defined by its *counterclockwise endpoint* $ccw(A_v)$ and its *clockwise endpoint* $cw(A_v)$. Note that a circular-arc representation of a graph G which fails to cover some point p on the circle is topologically the same as an interval representation of G [17]. A graph $G = (V, E)$ is a *tolerance graph* if to each vertex $v \in V$ can be associated an interval I_v and a positive real number $t(v)$ referred to as its tolerance, such that each pair of distinct vertices $u, v \in V$ are adjacent if and only if $|I_u \cap I_v| > \min\{t(u), t(v)\}$. The family $\{I_v\}_{v \in V}$ is a *tolerance representation* of G. When G has a tolerance representation such that the tolerance associated to each vertex $v \in V$ is smaller than the length of I_v, G is a *bounded tolerance graph*. Every bounded tolerance graph is the complement of a comparability graph [18].

A graph G is a *proper interval graph* if there is an interval representation of G in which no interval properly contains another. A graph G is a *unit interval graph* if there is an interval representation of G in which all the intervals have the same length. The notion of proper or unit is defined similarly for circular-arc graphs and tolerance graphs. Proper interval graphs and proper circular-arc graphs are *claw-free graphs*, since they do not admit the claw $K_{1,3}$ as induced subgraph [17]. Another well-known subclass of claw-free graphs is formed by line-graphs. The *line-graph* of a graph G, denoted by $L(G)$, is the incidence graph of the edges of G: the edges of G are the vertices of $L(G)$ and two vertices of $L(G)$ are adjacent if their corresponding edges in G are incident to a same vertex.

The number of vertices and the number of edges of the graph $G = (V, E)$ are respectively denoted by n and m throughout the paper. A *complete set* or *clique* is a subset of pairwise adjacent vertices. The clique C is *maximum* if no other clique of the graph has a size strictly greater than the one of C; $\omega(G)$ denotes the size of a maximum clique in the graph G. On the other hand, an *independent set* or *stable* is a subset of pairwise non-adjacent vertices and the *stability* $\alpha(G)$ of a graph G denotes the size of a maximum stable in G. A *q-coloring* of the graph G corresponds to a partition of G into q stables. The number $\chi(G)$, which denotes the cardinality of a minimum coloring in G, is called the *chromatic number* of G. By analogy, the cardinality of a minimum coloring of G such that each color appears at most k times is denoted by $\chi(G, k)$. A trivial lower bound for the number $\chi(G, k)$ is given by the expression $\max\{\chi(G), \lceil n/k \rceil\}$.

All the graph-theoretical terms which are not defined here can be found in [6,17]. For more details on these graphs and their applications, the reader can consult the books of Roberts [28,29], Golumbic [17,18] or Fishburn [12].

1.3 The sufficient condition

In [16], the following property is implicitly demonstrated.

Proposition 1.1 *Let G be a proper circular-arc graph and k an integer. If G admits a coloring such that each color appears at least k times, then $\chi(G, k) = \lceil n/k \rceil$. Moreover, the MES problem is solved in linear time and space given such an initial coloring in input.*

This property, called *repartitioning property*, has a first practical interest. The parameter k, which concerns here the number of available processors or machines, is generally small in comparison with n, the number of tasks to schedule. Thus, in many cases, a simple coloring heuristic enables to obtain a partition of the conflict graph in stables of size at least k. In the context of workforce planning, some structural properties often guarantee the existence of such a partition. For example, this is the case when municipal bus drivers or airport employees are planned (such cases have been encountered during the development of the software BAMBOO, edited by the firm Experian-Prologia SAS [3]): the frequencies of buses or planes often induce some sets of consecutive tasks having a size greater than k (for reasonable values like $k \leq 5$).

The second practical interest of such a property appears when more resources are allocated to the completion of tasks. Indeed, having realized a perfect scheduling of the tasks to the k processors (that is, n/k tasks are assigned to each processor with n multiple of k), the addition of a new processor does not necessarily save completion time because of the structure of the subjacent conflict graph. According to the previous proposition, we obtain the following guarantee for proper circular-arc graphs.

Corollary 1.2 *Let G be a proper circular-arc graph. If G admits an exact partition into stables of size k, then G admits an optimal partition into $\lceil n/k' \rceil$ stables of size at most k', for all $k' < k$.*

In the following sections, we show that the repartitioning property is shared by claw-free graphs, by interval graphs and circular-arc graphs, by proper tolerance graphs for $k = 2$, and by chordal graphs for $k \leq 4$. Moreover, the proofs which are given are constructive, providing some efficient algorithms to solve the MES problem given an initial coloring satisfying the condition in input. These results are all the more unexpected as a very simple counterexample can be found not satisfying this property.

All the results presented here appear in the author's thesis [14], written in French, and have been announced in [13,15].

1.4 Counterexamples

Consider the complete bipartite graph $K_{k+1,k+1}$, with $k \geq 2$ (see Figure 1). One can observe that $\chi(K_{k+1,k+1}, k) = 4$ since two vertices which belong to different stables can not be matched. As the lower bound is $\lceil n/k \rceil = \lceil (2k+2)/k \rceil = 3$, the complete bipartite graph $K_{k+1,k+1}$ does not share the repartitioning property for any $k \geq 2$.

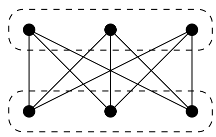

Fig. 1. The complete bipartite graph $K_{3,3}$.

Cographs, which are the graphs without a path of length four as induced subgraph, form a subclass of bounded tolerance graph; these are also known as the graphs of series-parallel orders. Clearly, the graph $K_{k+1,k+1}$ is a cograph for any positive value of k. The classes of weakly triangulated graphs and of Meyniel graphs, which are two extensions of chordal graphs, contain the graph

$K_{k+1,k+1}$ too. (The interested reader is referred to [6] for more details about these classes of graphs.)

From the complexity point of view, some strikes appear too. In their paper, Bodlaender and Jansen [5] investigate the complexity of the MES problem for bipartite graphs and show that the problem of partitioning a bipartite graph into three stables of size at most k is $\mathcal{NP}$-complete. Having observed that the bipartite graph used for the reduction is composed of two stables of size greater than k, we obtain the following result.

Proposition 1.3 (Bodlaender and Jansen, 1995) *The MES problem for bipartite graphs remains $\mathcal{NP}$-hard, even if a coloring of the graph G where each color appears at least k times is given in input.*

2 A characterization of claw-free graphs

The graph $K_{k+1,k+1}$ do not appear as an induced subgraph into claw-free graphs, for all $k \geq 2$. But do claw-free graphs share the repartitioning property? The next proposition shows that these graphs possess a property which is even stronger.

This proposition, which gives an algorithmic characterization of claw-free graphs, was established in other terms by De Werra [33] while he studied some timetabling problems. The *equitable coloring problem* consists in determining a coloring of this graph such that the number of vertices in each color class is the same except from one (see [4] for a survey of results on this subject). As pointed by Hansen et al. [21], there is a closed link between the equitable coloring problem and the mutual exclusion scheduling problem. In this section, we unify and complete the works of De Werra [33] and of Hansen et al. [21] on this subject. In particular, an algorithm coupled with special data structures is given to solve the MES problem and the equitable coloring problem for claw-free graphs, given a minimum coloring of the graph in input. In the same time, these results provide a generalization of Corollary 2.9 and Corollary 3.4 established in [16] for proper interval graphs and proper circular-arc graphs.

Curiously, the works of De Werra [33] and Hansen et al. [21] are not mentioned neither in the large survey recently proposed by Faudree et al. [11] on claw-free graphs, nor in those of Jansen [24] and Bodlaender and Fomin [4] on mutual exclusion scheduling and equitable coloring.

Proposition 2.1 (De Werra 1985, Hansen et al. 1993) *For a graph G, the following conditions are equivalent:*

(1) G is claw-free,
(2) any connected component of a subgraph induced by two disjoint stables in G is isomorphic to a chain or an even cycle,
(3) for any subgraph $G' \subseteq G$ induced by n' vertices, the equality

$$\chi(G', k) = \max\{\chi(G'), \lceil n'/k \rceil\}$$

is satisfied for all $k \geq 1$,
(4) for any subgraph $G' \subseteq G$ induced by n' vertices, G' admits an equitable q-coloring for all $q \geq \chi(G')$.

Proof. (1) ⇒ (2). Any subgraph induced by two disjoint stables is bipartite and contains no odd cycle. This bipartite graph is also claw-free here, which implies that all its vertices have a degree at most two. Consequently, each of its connected components is isomorphic to a chain or an even cycle.

The implications (2) ⇒ (3) and (2) ⇒ (4) are only established for the graph G, since they are immediately extended to any induced subgraph G' by heredity of the claw-free property.

(2) ⇒ (3). The proof relies on Lemma 3.2 established in [16] which remains valid for claw-free graphs according to (2): given a claw-free graph G and a positive integer k, a minimum coloring of G exists which satisfies either (a) each color appears at least k times, or (b) each color appears at most k times. The algorithm Refine-Coloring written below is used to obtain such a refined coloring of G.

Algorithm Refine-Coloring;
Input: a minimum coloring $\mathcal{S} = \{S_1, \ldots, S_{\chi(G)}\}$ of G, an integer k;
Output: a coloring $\mathcal{S}$ satisfying one of the two conditions (a) or (b);
Begin;
  while two disjoint stables $S_u, S_v \in \mathcal{S}$ exist such that $|S_u| > k$ and $|S_v| < k$ **do**
    $\mathcal{B} \leftarrow$ Connected-Components(S_u, S_v);
    while $|S_u| > k$ and $|S_v| < k$ **do**
      choose a connected component $B_r \in \mathcal{B}$ such that $|B_r^u| = |B_r^v| + 1$;
      exchange the vertices of S_u and S_v corresponding to B_r^u and B_r^v;
  return $\mathcal{S}$;
End;

Let $\mathcal{S} = \{S_1, \ldots, S_{\chi(G)}\}$ be a minimum coloring of G which has been refined through the algorithm Refine-Coloring. If $\mathcal{S}$ satisfies the condition (a), then we have immediately $\chi(G, k) = \chi(G)$. Otherwise, if $\mathcal{S}$ satisfies the condition (b), we show how to obtain a partition of G into $\lceil n/k \rceil$ stables of size at most k. Note that in this case, the inequality $n > k\chi(G)$ holds.

Let $|S_u| = \alpha_u k + \beta_u$ be the size of a stable S_u ($1 \leq u \leq \chi(G)$), with α_u a strictly positive integer and $0 \leq \beta_u \leq k-1$. First, extract from each stable S_u ($1 \leq u \leq \chi(G)$) $\alpha_u - 1$ stables of size exactly k, plus one if $\beta_u = 0$. After this operation, denote by $\bar{\chi}$ the number of stables which remains non empty and $\bar{n}$ the total number of vertices in these $\bar{\chi}$ stables. Clearly, each non empty stable S_u contains no more than $k+\beta_u$ with $\beta_u > 0$ and the inequality $\bar{n} > k\bar{\chi}$ is still verified. At the rate of one only per stable, extract now $\lfloor \bar{n}/k \rfloor - \bar{\chi}$ stables of size k, plus one of size $\bar{n} \bmod k$ if $\bar{n}$ is not a multiple of k. In this way, the number of vertices which remain in the $\bar{\chi}$ stables is exactly $k\bar{\chi}$ and at least two stables S_u and S_v of the partition are such that $|S_u| > k$ and $|S_v| < k$. Consequently, a new application of the algorithm Refine-Coloring on this partition (which remains claw-free) enables us to obtain $\bar{\chi}$ stables of size exactly k. To summary, only stables of size k have been extracted, except one of size $n \bmod k$ if n is not a multiple of k.

(2) $\Rightarrow$ (4). Let $S_1, \ldots, S_q$ be a q-coloring of G with $q \geq \chi(G)$. An equitable q-coloring is obtained by using the algorithm Refine-Coloring having replaced the conditions $|S_u| > k$ and $|S_v| < k$ in the two **while** loops by $|S_u| > \lceil n/q \rceil$ and $|S_v| < \lfloor n/q \rfloor$. The correctness of the algorithm is obtained by the same arguments than for the algorithm Refine-Coloring, except that here an equitable q-coloring is returned after q principal loops.

Since the implications (3) $\Rightarrow$ (1) and (4) $\Rightarrow$ (1) are straightforward (the graph $K_{1,3}$, which is 2-colorable, admits no partition into two stables of size two), the proof of the proposition is completed. $\square$

2.1 Computational issues

Some complexity questions are discussed which are related to the mutual exclusion scheduling problem and the equitable coloring problem for claw-free graphs. We show how to implement efficiently the procedure Refine-Coloring, which plays a central role in finding an optimal solution to the MES problem, by using a special data structure.

We assume that the input of the algorithm is composed of the claw-free graph $G = (V, E)$ represented by adjacency lists, of a minimum coloring $\mathcal{S} = \{S_1, \ldots, S_{\chi(G)}\}$ represented by $\chi(G)$ arrays (the array S_u containing the vertices of color u), and of the positive integer k. Note that the size of an array or of a list is considered to be obtainable in $O(1)$ time.

The special data structure, called F, is an array of size n. For each vertex $i \in V$, the cell F_i of the array has three attributes: an integer $F_i.color$ which represents the color of i (that is, the index of the stable which it belongs to), an integer $F_i.rank$ which represents the rank of i in this stable, and an array $F_i.S$ of size $\chi(G)$ which contains in each cell $F_i.S_u$ the list (of indices) of the vertices adjacent to i in the stable S_u (the size of this list is given by $F_i.|S_u|$). According to the assertion (2) of Proposition 2.1, the number of vertices stored in the list $F_i.S_u$ can not exceed two. Thus, the space required by the structure F is bounded by $O(\chi(G)n)$. Filling this structure can be done in $O(\chi(G)n)$ time, by first computing the attributes $F_i.color$ and $F_i.rank$ for each vertex $i \in V$ and then exploring the adjacency list associated to each vertex $i \in V$ in order to fill in the list $F_i.S$.

We are able to establish the complexity of the refinement algorithm. Having identified in $O(\chi(G))$ time two stables S_u and S_v such that $|S_u| > k$ and $|S_v| < k$, the algorithm calls the procedure Connected-Components which is described now. Having stored $t = \min\{|S_u| - k, k - |S_v|\}$ even chains having two extremities in S_u, the vertices of these two connected components are inserted into the lists $\mathcal{B}^u$ or $\mathcal{B}^v$, according to whether they come from S_u or S_v. Determining the t connected components is done as follows. The stable u is scanned to find some vertices having only one neighborhood in S_v (these vertices correspond to extremities of chains in the subgraph induced by S_u and S_v). When such a vertex is found, it is marked and the chain of which it is the extremity is traversed. If the length of this chain is even, then these vertices are stored into the structure $\mathcal{B} = \mathcal{B}^u \cup \mathcal{B}^v$. All this work takes $O(|S_u| + |S_v|)$ time and space thanks to the structure F which allows to obtain in $O(1)$ time the neighborhood of a vertex of the bipartite graph induced by $S_u \cup S_v$.

The exchange of vertices between stables is done as follows. First, the vertices to exchange are marked into the arrays S_u and S_v from the sets of vertices $\mathcal{B}^u \cup \mathcal{B}^v$ (this takes $O(|S_u| + |S_v|)$ time by using the field *rank* of the structure F). For each vertex $i \in V$, we can now proceed to the exchange of vertices in the lists $F_i.S_u$ and $F_i.S_v$. Since scanning these lists takes a constant time, the exchange is performed in $O(1)$ time (by using the field *rank* of the structure F and the marks on the vertices to exchange in S_u and S_v). Then, the arrays S_u and S_v can be updated in $O(|S_u| + |S_v|)$ time, as well as the fields $F_i.color$ and $F_i.rank$ of the exchanged vertices $i \in V$. Consequently, the complete update of the structure F is done in $O(n)$ time.

To summary, the use of the structure F, whose construction requires $O(\chi(G)n)$ time and space, allows to implement the principal loop of the algorithm Refine-Coloring so as to consume only $O(n)$ time and space at each iteration. Since $\chi(G)$ iterations suffice to refine a coloring, the algorithm Refine-Coloring runs in $O(\chi(G)n)$ time and space. By adjoining this result to the constructive proof of the implication (2) $\Rightarrow$ (3) of Proposition 2.1, we obtain the following results.

Proposition 2.2 *The MES problem is solved in $O(\chi(G)n)$ time and space for claw-free graphs, given a minimum coloring of the graph G in input.*

Corollary 2.3 *The MES problem is solved in $O(n^2/k)$ time and space for claw-free graphs, given a coloring of the graph G where each color appears at least k times in input.*

Corollary 2.4 *The equitable coloring problem is solved in $O(qn)$ time and space for claw-free graphs, given a q-coloring of the graph G in input.*

Remark 2.5 *Remind that the MES problem as well as the equitable coloring problem are $\mathcal{NP}$-hard for claw-free graphs, since finding a minimum coloring is $\mathcal{NP}$-hard for line-graphs [22].*

2.2 Applications

Here are some applications of the previous results. Since the minimum coloring problem is solved in $O(n^4)$ time for perfect claw-free graphs [23], we obtain the following corollary.

Corollary 2.6 *The MES problem is solved in $O(n^4)$ time and $O(n^2)$ space for perfect claw-free graphs.*

Now, some corollaries are given concerning the MES problem for line-graphs, a well-known subclass of claw-free graphs. The MES problem restricted to line-graphs can be viewed as the problem of determining a minimum coloring of the edges of a graph such that each color appears at most k times (two edges incident to a same vertex require different colors). Edge-coloring problems have important real-life applications in timetabling and scheduling [32–35]. The line-graph of G is denoted by $L(G)$, whereas n, m and $\Delta(G)$ denotes respectively the number of vertices, the number of edges and the maximum degree of G.

Corollary 2.7 *Let G be a graph and k an integer. If $\Delta(G) \geq \lceil m/k \rceil$, then $\Delta(G) \leq \chi(L(G), k) \leq \Delta(G) + 1$ and the MES problem for $L(G)$ is $\mathcal{NP}$-hard. Otherwise, $\chi(L(G), k) = \lceil m/k \rceil$ and the MES problem for $L(G)$ is solved in polynomial time and space.*

Proof. Transposing the assertion (3) of Proposition 2.1 to the line-graph of G gives $\chi(L(G), k) = \max\{\chi(L(G)), \lceil m/k \rceil\}$, whereas Vizing's Theorem (cf. [9, p. 103–105]) provides the inequalities $\Delta(G) \leq \chi(L(G)) \leq \Delta(G) + 1$. Then, when $\Delta(G) < \lceil m/k \rceil$, we obtain $\chi(L(G), k) = \lceil m/k \rceil$ and the constructive proof of Vizing's Theorem coupled with Proposition 2.2 gives a polynomial-time algorithm for determining a partition into $\lceil m/k \rceil$ stables of size at most k. Otherwise, we have $\chi(L(G), k) = \chi(L(G))$ (which implies immediately

$\Delta(G) \leq \chi(L(G), k) \leq \Delta(G) + 1)$ and the problem is $\mathcal{NP}$-hard according to Hoyler's Theorem [22]. $\square$

Remark 2.8 *When k is a constant of the problem, Alon [1] has shown that the MES problem is solvable in polynomial-time for line-graphs. On the other hand, Cohen and Tarsi [7] have established that the MES problem is $\mathcal{NP}$-hard for complements of line-graphs, even if k is a constant greater than or equal to three.*

Corollary 2.9 *If a graph G contains no induced cycle of length greater than or equal to five, then the MES problem for $L(G)$ is solved in polynomial time and space. Moreover, the equality $\chi(L(G), k) = \max\{\Delta(G), \lceil m/k \rceil\}$ holds.*

Proof. Trotter [30] has shown that the line-graph $L(G)$ of a graph G is perfect if and only if G contains no induced cycle of length greater than or equal to five (see also [27,31]). Since a minimum coloring can be found in polynomial time and space for perfect graphs [19], the result follows from Proposition 2.2. In addition, when $L(G)$ is perfect, we have $\Delta(G) = \chi(L(G))$. $\square$

A graph is weakly triangulated if and only if itself or its complement contain no induced cycle of length greater than or equal to five. Then, the following result holds.

Corollary 2.10 *The MES problem is solved in polynomial time and space for line-graphs of weakly triangulated graphs.*

Remark 2.11 *For example, bipartite graphs satisfy the conditions of the two previous corollaries. Note that for these ones, the result can be directly deduced from König's Theorem and its constructive proof (cf. [9, p. 103]).*

3 Sufficiency for interval and circular-arc graphs

Interval graphs contain no induced subgraph $K_{k+1,k+1}$ for any $k \geq 2$. Indeed, interval graphs are chordal and then admit no induced cycle of length greater than or equal to four. But do they share the repartitioning property?

In this section, we give a positive answer to this question, providing in addition a linear-time and space algorithm to solve the MES problem given a coloring where each color appears at least k times in input. Besides, the result is extended to circular-arc graphs. Remind that the MES problem remains $\mathcal{NP}$-hard for these two classes of graphs, even for fixed $k \geq 4$ [5]. To conclude the section, the extension of the repartitioning property is discussed for proper tolerance graphs.

3.1 The case of interval graphs

Lemma 3.1 (Repartitioning Lemma) *Let $S_1, \ldots, S_r$ be r disjoint stables each one containing at least t intervals ($1 \leq r \leq t$) and $\beta_1, \ldots, \beta_r$ r positive integers such that $\sum_{u=1}^{r} \beta_u = t$. Then, a stable S^* of size t exists such that for all $u = 1, \ldots, r$, exactly β_u intervals of S^* belong to S_u. Moreover, this stable S^* is extracted in $O(rt)$ time, given the intervals of each stable ordered.*

Proof. The proof is constructive: an algorithm is given which finds the stable S^* in $O(rt)$ time. The input of this algorithm consists of the r disjoint stables $S_1, \ldots, S_r$, each one of size at least t, and the r integers $\beta_1, \ldots, \beta_r$. We assume that the open intervals of each stable are ordered according to increasing left endpoints (the interval of rank j in S_u is denoted by $I_{u,j}$). The algorithm proceeds as follows: the interval of rank j in S^* is selected as the one having the smallest right endpoint among the intervals of rank j in the stables S_u (where β_u intervals have not been selected yet).

Algorithm Repartition-Intervals;
Input: the stables $S_1, \ldots, S_r$ of size at least t, the integers $\beta_1, \ldots, \beta_r$;
Output: the stable S^*;
Begin;
  $S^* \leftarrow \emptyset$;
  for j from 1 to t **do**
    $u^* \leftarrow 0$;
    for u from 1 to r **do**
      if $\beta_u > 0$ and ($u^* = 0$ or $r(I_{u,j}) < r(I_{u^*,j})$) **then** $u^* \leftarrow u$;
    $S^* \leftarrow S^* \cup \{I_{u^*,j}\}$, $\beta_{u^*} \leftarrow \beta_{u^*} - 1$;
  return S^*;
End;

Figure 2 illustrates the execution of the algorithm on three stables S_1, S_2, S_3 of size three ($r = 3$, $t = 3$ and $\beta_1 = \beta_2 = \beta_3 = 1$). The dark intervals correspond to intervals included in S^* at each rank $j = 1, 2, 3$ (intervals which are no more candidates to the selection at a given rank are hachured).

Let us establish the correctness of the algorithm. The output set S^* contains well t intervals, because one interval is extracted at each rank $j = 1, \ldots, t$ and the input stables are all of size at least t. Now, we show that for all $j = 1, \ldots, t-1$, the interval $I_{u,j} \in S_u$ included in S^* at rank j and the interval $I_{v,j+1} \in S_v$ included in S^* at rank $j+1$ are such that $r(I_{u,j}) \leq l(I_{v,j+1})$. If

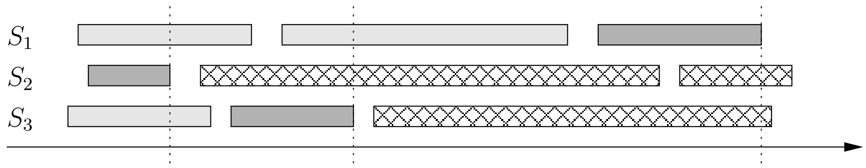

Fig. 2. The algorithm Repartition-Intervals.

$u = v$, then the assertion trivially holds. Otherwise, suppose that $l(I_{v,j+1}) < r(I_{u,j})$. As $r(I_{v,j}) \leq l(I_{v,j+1})$, we obtain that $r(I_{v,j}) < r(I_{u,j})$. Since $I_{v,j+1}$ have been selected at rank $j + 1$, the interval $I_{v,j}$ was necessarily a candidate to the selection at rank j. Consequently, $I_{u,j}$ was not, among the intervals candidates at rank j, the one having the smallest right endpoint, which is a contradiction. $\square$

Proposition 3.2 *Let G be an interval graph and k an integer. If G admits a coloring such that each color appears at least k times, then $\chi(G, k) = \lceil n/k \rceil$. Moreover, the MES problem for G is solved in linear time and space, given such an initial coloring in input.*

Proof. Let $S_1, \ldots, S_q$ be a partition of G into stables of size at least k. Assume that the intervals of each stable are ordered and denote the size of the stable S_u by $|S_u| = \alpha_u k + \beta_u$, with α_u a strictly positive integer and $0 \leq \beta_u \leq k-1$. First, from each stable S_u are extracted $\alpha_u - 1$ stables of size k, plus one if $\beta_u = 0$. Then, while r stables exist whose β_u's sum is greater than k, Repartitioning Lemma (3.1) is applied to extract some stables of size exactly k (the conditions of lemma are satisfied because each stable contains at least k intervals). At the end of the process, one stable of size $n \bmod k$ remains to extract if n is not a multiple of k.

Having computed an ordered interval representation of G in linear time and space [20] (see also [6,17] for more details on interval graph recognition), the intervals of each stable of $S_1, \ldots, S_q$ are ordered in $O(n)$ time and space. Thanks to careful applications of the procedure Repartition-Intervals, the extraction of stables of size k can be done in $O(qk)$ time on the whole, that is, $O(n)$ time since $n > qk$. $\square$

3.2 The case of circular-arc graphs

In the present form, Repartitioning Lemma is not extendable to circular-arcs; the following example shows that, even restricted to unit circular-arcs, an infinity of cases exists for which Repartitioning Lemma does not hold.

Let $S_1, \ldots, S_r$ be a set of disjoint stables containing each one t open unit circular-arcs with $\beta_u = 1$ for all $u = 1, \ldots, r$ $(r, t \geq 1)$. In order to define the position of these arcs on the circle, divide the circle into t sections $\theta_0, \ldots, \theta_{t-1}$, each one of length ℓ. An arc has rank j if its counterclockwise endpoint belongs to the section θ_j of the circle. The r arcs of rank j are disposed in section j, each arc being of length ℓ and shifted of $\epsilon < \ell/t$ from the previous so as to the arc of S_u overlaps on one side the arcs of rank $(j+1) \bmod t$ which belong to stables with index lower than u and on the other side the arcs of rank $(j-1) \bmod t$ which belong to stables with index greater than u (see Figure 3).

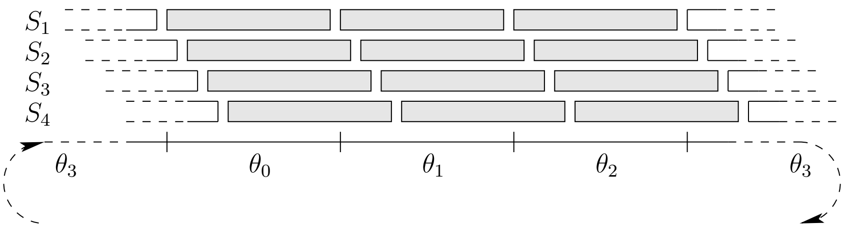

Fig. 3. An example of construction with $r = 4$ and $t = 4$.

Now, suppose that a stable S^* of size t exists having one and only one arc in each stable S_u for all $u = 1, \ldots, r$. Since none of these arcs can have the same rank, S^* contains one arc of rank j for all $j = 1, \ldots, t$. Assume that the arc $A_1^* \in S^*$, coming from S_1, is of rank j. According to the previous observation, S^* contains necessarily one arc of rank $(j-1) \bmod t$ from $S_2, \ldots, S_r$. But, by definition, all arcs of rank $(j-1) \bmod t$ overlap A_1^*. Consequently, we obtain a contradiction and no stable of size t exists in $S_1, \ldots, S_r$ with the desired property. In fact, one can prove that no stable of size t exists except $S_1, \ldots, S_r$ themselves.

Despite this negative result, one can observe that in the constructive proof of Proposition 3.2, the stables to which Repartitioning Lemma is applied are all of size strictly greater than k. Thus, a weaker version of Repartitioning Lemma could be employed in this proof, where each stable would be of size $t+1$, and not of size t.

Lemma 3.3 (Weak Repartitioning Lemma) *Let $S_1, \ldots, S_r$ be r disjoint stables each one containing at least $t+1$ circular-arcs $(1 \leq r \leq t)$ and $\beta_1, \ldots, \beta_r$ r positive integers such that $\sum_{u=1}^{r} \beta_u = t$. Then, a stable S^* of size t exists such that for all $u = 1, \ldots, r$, exactly β_u arcs of S^* belong to S_u. Moreover, this stable S^* is extracted in $O(rt)$ time, given the arcs of each stable ordered.*

Proof. The proof relies on Repartitioning Lemma (3.1). Let p be a point on the circle. For all $u = 1, \ldots, r$, remove from each stable S_u the arc which contains p. Having removed these arcs, no arc of $S_1, \ldots, S_r$ covers the point

p any more. Consequently, the graph induced by these stables is an interval graph and Repartitioning Lemma can be applied (each stable having a size at most t).

Clearly, removing the arcs which cover p takes $O(rt)$ time. Then, the procedure Repartition-Intervals can be applied on the new set of arcs, provided that the arcs of each stable are renumbered clockwise from the point p. Therefore, the total time necessary to the extraction of S^* remains in $O(rt)$. $\square$

According to the previous discussion, the following proposition is established.

Proposition 3.4 *Let G be a circular-arc graph and k an integer. If G admits a coloring such that each color appears at least k times, then $\chi(G, k) = \lceil n/k \rceil$. Moreover, the MES problem for G is solved in linear time and space, given such an initial coloring in input.*

Remark 3.5 *A corollary of Weak Repartitioning Lemma is that circular-arc graphs admit no induced subgraph $K_{k+1,k+1}$ for all $k \geq 2$. Unlike interval graphs, they admit induced copies of $K_{2,2}$, also known as the chordless cycle of length four C_4.*

3.3 The case of proper tolerance graphs

As noticed in introduction, the graph $K_{k+1,k+1}$ belongs to the class of bounded tolerance graphs, for any positive value of k (see Figure 4 below). However, a question remains: does the graph $K_{k+1,k+1}$ admit a proper tolerance representation? In this last part, we answer to this question for $k = 2$, by demonstrating that proper tolerance graphs admit no induced copy of $K_{3,3}$. On the other hand, Repartitioning Lemma is shown to be not extendable to unit tolerance graphs for $k \geq 3$, even in its weak version.

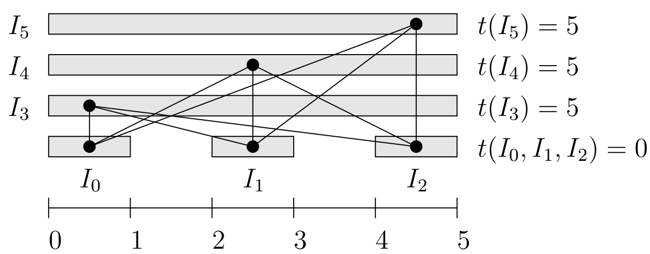

Fig. 4. A bounded tolerance representation of the graph $K_{3,3}$.

Lemma 3.6 *Let G be a proper tolerance graph and A, B two disjoint stables of G each one containing three vertices. Then, a vertex in A and a vertex in B exist which are not connected by an edge.*

Proof. Consider a proper tolerance representation of G where all the intervals are open and write $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$ with (i) $g(a_1) \le g(a_2) \le g(a_3)$ and $g(b_1) \le g(b_2) \le g(b_3)$ (because the intervals are proper, the right endpoints are in the same order than the left endpoints). Without loss of generality, we assume that $g(a_2) \le g(b_2)$. Then, we claim that the vertices a_1 and b_3 are not connected by an edge. First, the inequality $d(a_1) - g(b_3) \le \min\{t(a_1), t(b_3)\}$ is shown. Since the pairs of vertices a_1, a_2 and b_2, b_3 belong to the same stable, we have that (ii) $d(a_1) - g(a_2) \le \min\{t(a_1), t(a_2)\}$ and $d(b_2) - g(b_3) \le \min\{t(b_2), t(b_3)\}$. The intervals being proper and $g(a_2) \le g(b_2)$, we also have that (iii) $d(a_1) - g(a_2) \ge d(a_1) - g(b_2)$ and $d(b_2) - g(b_3) \ge d(a_2) - g(b_3)$. By combining the inequalities (i), (ii) and (iii), we obtain:

$$d(a_1) - g(b_3) \le d(a_1) - g(b_2) \le d(a_1) - g(a_2) \le \min\{t(a_1), t(a_2)\} \le t(a_1)$$

$$d(a_1) - g(b_3) \le d(a_2) - g(b_3) \le d(b_2) - g(b_3) \le \min\{t(b_2), t(b_3)\} \le t(b_3)$$

Thus, the inequality $d(a_1) - g(b_3) \le \min\{t(a_1), t(b_3)\}$ is valid and our claim is demonstrated. □

Proposition 3.7 *Let G be a proper tolerance graph. If G admits a coloring such that each color appears at least two times, then $\chi(G, 2) = \lceil n/2 \rceil$. Moreover, the 2-MES problem for G is solved in linear time and space, given such an initial coloring in input.*

Remark 3.8 *Such a result might be applied to solve in linear time and space the 2-MES problem for proper tolerance graphs, as it was done for interval graphs in [16].*

Extending the previous proposition for $k \ge 3$ seems to be difficult. Indeed, the following example shows that Repartitioning Lemma is not extendable to bounded unit tolerance graphs, even in its weak version.

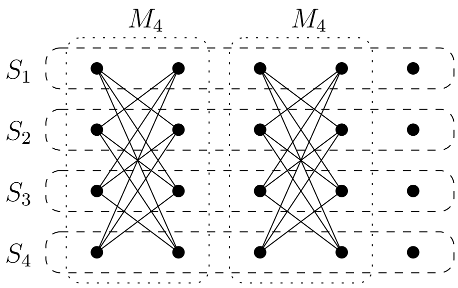

Fig. 5. The graph N_4.

Let M_k be the graph defined as the union of two sets of vertices numbered from 1 to k such that the only non-connected pairs of vertices have identical numbers. Now, the graph N_k is defined, for $k \ge 3$, as the union of two copies of M_k and one stable S of size $k^2 - 3k$ (see Figure 5). The graph N_k admits a partition $S_1, \ldots, S_k$ into k stables of size $k+1$, where each stable S_k is composed of the four vertices with number k in $M_k \cup M_k$ and of $k-3$ vertices in S. We claim that the graph N_k admits no stable S^* of size k with one vertex in each stable S_u for all $u = 1, \ldots, k$. Indeed, if a vertex i of S^* belongs to the subgraph M_k, then no other vertex of M_k can belong to S^* (since the sole vertex to which i is not connected in M_k belongs to the same stable than i). Consequently, the stable S^* must contain at least $k-2$ vertices from the subgraph $N_k \setminus (M_k \cup M_k)$ belonging to distinct stables, which is impossible since each stable S_u $(1 \leq u \leq k)$ contains no more than $k-3$ vertices in this subgraph.

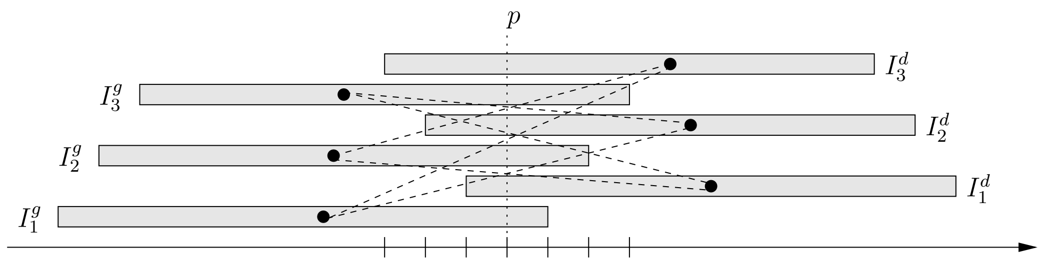

Fig. 6. A bounded unit tolerance representation of the graph M_3.

Now is described a bounded unit tolerance representation of M_k. Let p be an integer point on the real axis and ℓ an integer greater than $4k$. For all $i = 1, \ldots, k$, the two vertices with number i are respectively represented by the intervals $I_i^g =]p+i-\ell, p+i[$ and $I_i^d =]p-i, p-i+\ell[$ with tolerances $t(I_i^g) = t(I_i^d) = 2i$ (see Figure 6). The set composed of intervals I_i^g (resp. I_i^d) induces a clique, because these intervals share a portion of the axis having a length greater than $2k$. Then, for any pair of intervals I_i^g and $I_{i'}^d$, we have that $|I_i^g \cap I_{i'}^d| = i + i'$; since their tolerances are respectively $2i$ and $2i'$, these ones are connected by an edge only if $i + i' \leq 2i$ and $i + i' \leq 2i'$, that is, if $i = i'$. Consequently, this representation of M_k is correct and a bounded unit tolerance representation of N_k can be easily deduced from it.

Thus, the weak version of the repartitioning lemma holds for proper tolerance graphs when $t = 2$, but not for bounded unit tolerance graphs when $t \geq 3$. By observing that the graph M_2 is isomorphic to the induced cycle C_4, we deduce that the strong version of the repartitioning lemma does not hold any more for bounded unit tolerance graphs when $t = 2$.

Remark 3.9 *The weak version of the repartitioning lemma holds for planar graphs when $t = 2$, since any planar graph admits no induced subgraph $K_{3,3}$ according to Kuratowski's Theorem (cf. [9, pp. 80–84]). Consequently, planar graphs share the repartitioning property for $k = 2$. On the other hand, the graph N_3 is planar, which stops any attempt of extension for $t = 3$. As for proper tolerance graphs, the following question is asked: do planar graphs share the repartitioning property for $k \geq 3$?*

4 Sufficiency for chordal graphs

In this last section, the repartitioning property is discussed for chordal graphs. Despite many efforts, we succeed in extending Proposition 3.2 for $k \leq 4$ only.

Proposition 4.1 *Let G be an circular-arc graph and $k \leq 4$ an integer. If G admits a coloring such that each color appears at least k times, then $\chi(G, k) = \lceil n/k \rceil$.*

As for interval or circular-arc graphs, the proof of this proposition is based on a repartitioning lemma. But before detailing this, here is a lemma on which all the proofs of the section rely.

Lemma 4.2 (Local Sparsity Lemma) *Let A, B be two disjoint stables of a chordal graph. Then the subgraph induced by A and B is a forest which contains at most $|A| + |B| - 1$ edges.*

Proof. The subgraph induced by A and B is chordal and then contains no induced cycle of length greater than or equal to four. Since it is bipartite, it also contains no induced cycle of length three. Consequently, this one is acyclic and contains no more than $|A| + |B| - 1$ edges. □

A repartitioning lemma, in weak form and ad hoc to $k \leq 4$, is established in two parts. The first part, described below, allows to extract some stables of size two or three when applying the repartitioning process employed in the proof of Proposition 3.2. A vertex is d-connected (resp. exactly d-connected) to a stable when it is connected to at least (resp. exactly) d vertices of this stable.

Lemma 4.3 (Ad hoc Repartitioning Lemma) *Let A, B, C be three disjoint stables of a chordal graph.*

(1) If A and B are each one of size two, then a stable of size two exists with one vertex in A and one vertex in B.
(2) If A and B are each one of size three, then a stable of size three exists with two vertices in A and one vertex in B.
(3) If A, B and C are each one of size four, then a stable of size three exists with one vertex in each stable A, B and C.

Proof. The proof of assertion (1) is immediate according to Local Sparsity Lemma. Indeed, consider two vertices of A and two vertices of B. If assertion (1) is not verified, then the two vertices of A are both connected to the two vertices of B. This implies that the subgraph induced by A and B must contain four edges, whereas Local Sparsity Lemma imposes some less than three, which is a contradiction. More directly, assertion (1) results from the fact that the bipartite graph induced by A and B contains no induced copy of $K_{2,2}$, alias the chordless cycle C_4.

The proof of assertion (2) is quite as direct. If (2) is not satisfied, the three vertices of B are connected to at least two of the three vertices of A. Then, the subgraph induced by these six vertices contains six edges. On the other hand, Local Sparsity Lemma imposes some less than five, which is a contradiction.

Now suppose that no stable exists as described in assertion (3). First, we claim that any vertex $i \in A \cup B \cup C$ is 3-connected to at least one stable among A, B, C. Without loss of generality, let us assume that $i \in A$ contradicts this claim. Then, at least two vertices of B and two vertices of C exist which are not connected to i and according to Local Sparsity Lemma, one of the two vertices of B is not connected to one the two vertices of C. Since these last ones are not connected to i, they induce with i a stable of size three having the desired property. This is a contradiction and the claim is demonstrated. Now consider three vertices of A. According to the previous claim, these three vertices are each one 3-connected to stables B or C. Consequently, at least two of these three vertices are 3-connected to the same stable, which is impossible without violating Local Sparsity Lemma. □

Thereafter, we shall see how assertion (3) can be reinforced thanks to some additional efforts (see Lemma 4.10). The second part of the repartitioning lemma, described below, rules the extraction of stables of size four. Except the proof of assertion (4), the demonstration of this second part requires more efforts.

Lemma 4.4 (Ad hoc Repartitioning Lemma) *Let A, B, C, D be four disjoint stables of a chordal graph.*

(4) If A and B are each one of size four, then a stable of size four exists with three vertices in A and one vertex in B.
(5) If A and B are each one of size four, then a stable of size four exists with two vertices in A and two vertices in B.
(6) If A, B and C are respectively of size six, four and four, then a stable of size four exists with two vertices in A, one vertex in B and one in C.
(7) If A, B, C and D are each one of size five, then a stable of size four exists with one vertex in each stable A, B, C and D.

Assertion (4) follows immediately from Local Sparsity Lemma. Indeed, if no stable of size four exists with the desired property, then each vertex of B is connected to at least two vertices to A. Thus, the subgraph induced by A and

B must contain at least eight edges, whereas Local Sparsity Lemma imposes some less than seven.

To make the next proofs more readable, we need some specific definitions. Let A, B, C, D be four disjoint stables of a chordal graph, each one of size at least two. We call *doublet* in (A, B) a stable of size two with one vertex in A and one vertex in B; by analogy, a *triplet* in (A, B, C) (resp. a *quadruplet* in (A, B, C, D)) is a stable of size three (resp. four) with one vertex in each stable A, B and C (resp. A, B, C and D). A *square* in (A, B) is a stable of size four with two vertices in A and two vertices in B; by analogy, a *quasi-square* $\{a, b, a', b'\}$ in (A, B) corresponds to the succession of the three doublets $\{a, b\}$, $\{b, a'\}$ and $\{a', b'\}$, and becomes a square if the vertices a and b' are not connected (see Figure 7). Remind that the degree of a vertex i is denoted by $d(i)$.

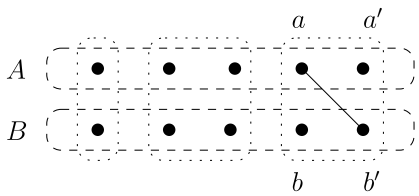

Fig. 7. A doublet, a square and a quasi-square.

Lemma 4.5 (Doublets Lemma) *Let A, B be two disjoint stables of a chordal graph. If A and B are of size at least $t \geq 2$, then $t - 1$ disjoint doublets exist in (A, B).*

Proof. When A and B are of size $t = 2$, the assertion follows immediately from Local Sparsity Lemma. For $t > 2$, we proceed as follows. While $t \geq 2$, we apply Local Sparsity Lemma on two stables $A' \subseteq A$ and $B' \subseteq B$ to extract one doublet. Then, the number of extracted doublets is $t - 1$. $\square$

4.1 Proofs of assertions (5) and (6) of Ad hoc Repartitioning Lemma

Here is the lemma on which relies the proof of assertions (5) and (6). In addition, it provides another simple proof of assertion (3) of Ad hoc Repartitioning Lemma (4.3).

Lemma 4.6 (Square Lemma) *Let A, B be two disjoint stables of a chordal graph. If A and B are each one of size at least four, then a square exists in (A, B).*

Proof. We write $A = \{a_1, a_2, a_3, a_4\}$ and $B = \{b_1, b_2, b_3, b_4\}$. Suppose that no square exists in (A, B). This implies that for each pair of vertices $a_i, a_j \in A$, the inequality $d(a_i)+d(a_j) \geq 3$ is satisfied (otherwise, at least two vertices of B are not connected to a_i and a_j and a square exists in (A, B)). Then, we deduce that three vertices of A are of degree at least two. On the other hand, Local Sparsity Lemma imposes that the sum of degrees of vertices in A is lower than or equal to seven. Consequently, we set without loss of generality $d(a_1) \leq 1$, $d(a_2) \geq 2$, $d(a_3) \geq 2$ and $d(a_4) \geq 2$. By applying Local Sparsity Lemma to the subgraph induced by $A \setminus \{a_1\}$ and B, we also obtain the inequality $d(a_2) + d(a_3) + d(a_4) \leq 6$ which, combined to the previous ones, gives (i) $d(a_1) \leq 1$ and $d(a_2) = d(a_3) = d(a_4) = 2$.

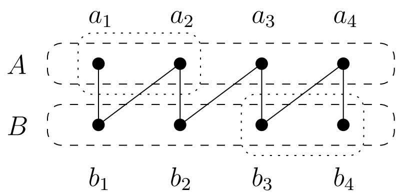

Fig. 8. The chain and the square.

By symmetric arguments, we obtain that (ii) $d(b_1) = d(b_2) = d(b_3) = 2$ and $d(b_4) \leq 1$ for the vertices of the set B. Once joined together, conditions (i) and (ii) force the bipartite graph induced by a and B to be isomorphic to a chain where $\{a_1, a_2, b_3, b_4\}$ forms a square (see Figure 8), which contradicts our first hypothesis. $\square$

Assertion (5) of Ad hoc Repartitioning Lemma (4.4) follows immediately from Square lemma. Here is the proof of assertion (6). Let A, B, C be three disjoint stables of a chordal graph, respectively of size six, four and four. According to Square Lemma, a square $\{b_1, c_1, b_2, c_2\}$ exists in (B, C). Then, consider the bipartite graph induced by this square and the stable A and suppose that no stable of size four exists with one vertex in $\{b_1, b_2\}$, one vertex in $\{c_1, c_2\}$ and two vertices in A. Clearly, this implies that for any pair b_i, c_j of vertices ($1 \leq i, j \leq 2$), the inequality $d(b_i)+d(c_j) \geq 5$ is satisfied. By summing the four inequalities in question, we obtain that the number of edges in the considered bipartite graph is greater than ten. However, Local Sparsity Lemma imposes less than nine edges, which is a contradiction.

A simple proof of assertion (3) is obtained by similar arguments. In fact, this proof technique can be generalized into the following lemma.

Lemma 4.7 *Let $X = X_1 \cup \cdots \cup X_t$ and Y be two disjoint stables of size $2t$ in a chordal graph, where $X_1, \ldots, X_t$ induce t disjoint pairs of vertices ($t \geq 1$). Then a stable of size $t + 1$ exists with one vertex in each set X_i ($1 \leq i \leq t$) and one vertex in Y.*

Proof. Assume that such a stable does not exists. Then, the inequality $d(x_1) + \cdots + d(x_t) \geq 2t$ is satisfied for any t-tuple $x_1 \in X_1, \ldots, x_t \in X_t$. By summing these inequalities for all possible t-tuples, we obtain that $d(X_1) + \cdots + d(X_t) \geq 4t$ where $d(X_i)$ represents the sum of degrees of vertices in X_i. On the other hand, Local Sparsity Lemma imposes that $d(X_1) + \cdots + d(X_t) \leq 4t - 1$, which contradicts the previous inequality. $\square$

Thus, the proof of assertion (3) corresponds to the combination of Square Lemma and Lemma 4.7 with $t = 2$ (set $X_1 = \{b_1, b_2\}$, $X_2 = \{c_1, c_2\}$ and $Y = A$ with $\{b_1, b_2, c_1, c_2\}$ a square in (B, C)).

Remark 4.8 *Assertion (6) can be reinforced in such a way that the stable A has only size five and stables B and C remain of size four. Indeed, the bipartite graph induced by the stable A and the square $\{b_1, c_1, b_2, c_2\}$ induces a tree. After an exhaustive search based on the maximum degree (or the diameter), a few non-isomorphic trees are found which have all the desired property.*

4.2 Proof of assertion (7) of Ad hoc Repartitioning Lemma

The proof of assertion (7), in the same spirit than the one given for assertion (3), relies on the fact that a triplet can be extracted from three disjoint stables of size only three. This last claim is proved according to the following lemma.

Lemma 4.9 (Quasi-square Lemma) *Let A, B be two disjoint stables of a chordal graph. If A and B are each one of size at least three, then a quasi-square exists in (A, B).*

Proof. Set $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$. The two following cases are distinguishable.

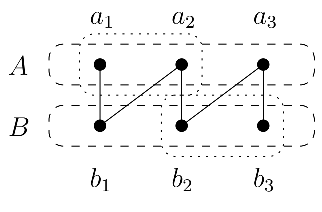

Fig. 9. Quasi-square Lemma: case (a).

Case (a): the maximum degree of the subgraph induced by A and B is at most two. In this case, there is only one non-isomorphic configuration which is maximal according to the number of edges: the chain. Without loss of generality, consider the chain $\{a_1, b_1, a_2, b_2, a_3, b_3\}$. Then, observe that the set $\{a_1, a_2, b_2, b_3\}$ induces a quasi-square (see Figure 9).

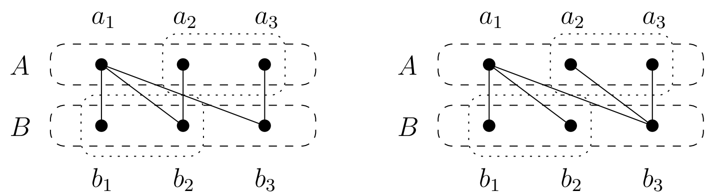

Fig. 10. Quasi-square Lemma: case (b).

Case (b): the maximum degree of the subgraph induced by A and B is three. W.l.o.g., assume that the vertex a_1 has a degree equal to three. According to Local Sparsity Lemma, we obtain that the vertices a_2 and a_3 have each one a degree lower than one. In this case, there are only two non-isomorphic configurations, which are illustrated on Figure 10. Having numbered the vertices of these two configurations as on this figure, we can observe that the set $\{a_2, a_3, b_1, b_2\}$ induces a quasi-square. $\square$

Lemma 4.10 (Triplet Lemma) *Let A, B, C be three disjoint stables of a chordal graph. If A, B and C are each one of size at least three, then a triplet exists in (A, B, C).*

Proof. Set $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2, b_3\}$ and $C = \{c_1, c_2, c_3\}$. According to the previous lemma, a quasi-square exists in (A, B). W.l.o.g., let $\{a_1, b_1, a_2, b_2\}$ be this quasi-square, with $\{a_1, b_2\}$ the pair of vertices which may be connected by an edge, and assume that no triplet exists in (A, B, C). We have necessarily that (i) $d(a_1)+d(b_1) \geq 3$, $d(b_1)+d(a_2) \geq 3$ and $d(a_2)+d(b_2) \geq 3$. On the other hand, Local Sparsity Lemma imposes that (ii) $d(a_1) + d(b_1) + d(a_2) \leq 5$ and $d(b_1)+d(a_2)+d(b_2) \leq 5$ (since the sets $\{a_1, b_1, a_2\}$ and $\{b_1, a_2, b_2\}$ induce each one a stable). From inequalities (i) and (ii), we can deduce that $1 \leq d(a_i) \leq 2$ and $1 \leq d(b_i) \leq 2$ for $i = 1, 2$. In this case, there are only two non-isomorphic configurations which are minimal according to the number of edges.

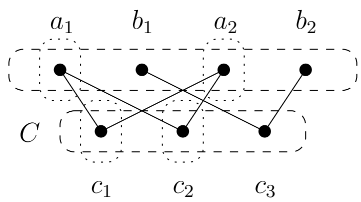

Fig. 11. Triplet Lemma: case (a).

Case (a): $d(a_1) = 2, d(b_1) = 1, d(a_2) = 2, d(b_2) = 1$. Assume w.l.o.g. that the vertex a_1 is connected to c_1 and c_2. By applying inequalities (i) in cascade, we obtain that the pairs $\{b_1, c_3\}$, $\{a_2, c_1\}$, $\{a_2, c_2\}$ of vertices must be connected by an edge (see Figure 11). Then, the set $\{b_1, c_2, b_2, c_3\}$ induces a chordless cycle of length four in (B, C), which is a contradiction.

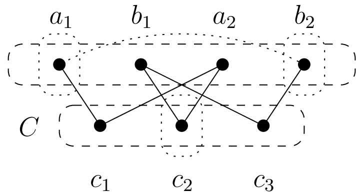

Fig. 12. Triplet Lemma: case (b).

Case (b): $d(a_1) = 1, d(b_1) = 2, d(a_2) = 2, d(b_2) = 1$. Assume w.l.o.g. that the vertex a_1 is connected to c_1. By applying inequalities (i) in cascade while respecting Local Sparsity Lemma, we obtain w.l.o.g. that the pairs $\{b_1, c_2\}$, $\{b_1, c_3\}$, $\{a_2, c_1\}$, $\{a_2, c_2\}$, $\{b_2, c_3\}$ of vertices must be connected by an edge (see Figure 12). If the vertex a_1 is connected to b_2, then the subgraph induced by $\{a_1, b_1, a_2, b_2\}$ and C is isomorphic to a chordless cycle of length seven, which is a contradiction. Otherwise, the set $\{a_1, b_2, c_2\}$ induces the desired triplet, which contradicts our initial hypothesis. $\square$

Finally, assertion (7) of Ad hoc Repartitioning Lemma can be established. Let A, B, C, D be four disjoint stables of a chordal graph, each one of size five. First, we show that if no quadruplet exists in (A, B, C, D), then any vertex of the subgraph induced by these four stables must be exactly 3-connected to one of the three stables to which it does not belong.

Assume w.l.o.g. that $a_1 \in A$ is 3-connected to none of the three stables B, C or D. In this case, a subgraph with three vertices in each stable B, C and D exists such that none of these nine vertices is connected to a_1. According to Triplet Lemma, a triplet exists in this subgraph and then a quadruplet in (A, B, C, D) (since each vertex of this triplet is not connected to a_1), which is a contradiction. On the other hand, assume w.l.o.g. that the vertex $a_1 \in A$ is 4-connected to the stable B. Clearly, no other vertex of A can be 3-connected to B without violating Local Sparsity Lemma. Assume w.l.o.g. that the vertices $a_2, a_3 \in A$ (resp. $a_4, a_5 \in A$) are 3-connected to C (resp. D). According to the previous discussion, each vertex of D must be 3-connected to A, B or C. Thus, at least one vertex of D is 3-connected to the stable B and assume w.l.o.g. that this one is $d_1 \in D$. As a_1 is 4-connected to B, two vertices of B exist which are both connected to a_1 and to d_1. According to Local Sparsity Lemma, these two last vertices must be connected by an edge. According similar arguments, another vertex $d_2 \in D$ is shown to be connected to a_1. Indeed, a vertex of D exists which is 3-connected to B like d_1 or 3-connected to A and connected to a_1 (see Figure 13). Hence, the vertex $a_1 \in A$ is 2-connected to the stable D. Since the vertices $a_4, a_5 \in A$ are 3-connected to the stable D, we obtain that the subgraph induced by $\{a_1, a_4, a_5\}$ and D violates Local Sparsity Lemma, which is a contradiction.

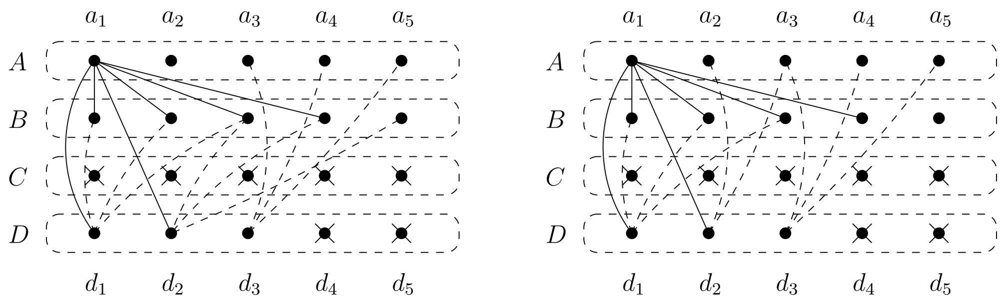

Fig. 13. The proof of assertion (7): a_1 is 4-connected to B. According to hypothesis, $d_4, d_5 \in D$ are assumed to be 3-connected to the stable C. Then, two cases are possible: two vertices of D of which d_1 are 3-connected to B (on the left) or two vertices of D different from d_1 are 3-connected to A (on the right).

We have just shown that any vertex of the subgraph induced by (A, B, C, D) must be exactly 3-connected to one of the three stables to which it does not belong. By using the same arguments, we can show that if no quadruplet exists in (A, B, C, D), then any vertex of this subgraph can not both 3-connected to a stable and 2-connected to another one. Assume w.l.o.g. that the vertex $a_1 \in A$ is connected to the vertices $b_1, b_2, b_3 \in B$ and to the vertices $c_1, c_2 \in C$. A second vertex of A must be 3-connected to the stable B, another one to the stable C and the two last ones to the stable D. Assume w.l.o.g. that $a_2 \in A$ is 3-connected to B, $a_3 \in A$ is 3-connected to C and $a_4, a_5 \in A$ are 3-connected to D. According to the previous discussion, each vertex of D must 3-connected to A, B or C. Then, we claim that at least two vertices of D satisfy one of the three following conditions: (i) the vertex is 3-connected to A and connected to a_1, (ii) the vertex is 3-connected to B and connected to at least two vertices among $\{b_1, b_2, b_3\}$, or (iii) the vertex is 3-connected to C and connected to c_1 and c_2. In order to avoid too heavy technical details, the proof of this claim is left to the reader (see Figure 14).

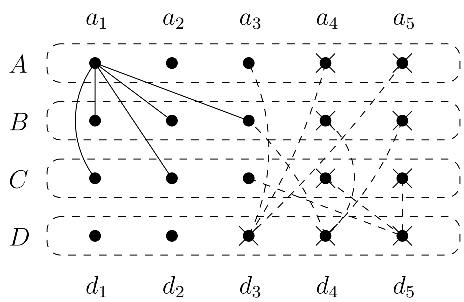

Fig. 14. The proof of assertion (7): a_1 is 3-connected to B and 2-connected to C.

Denote w.l.o.g. by d_1 and d_2 the two vertices of D satisfying one of these three conditions. Thus, we have that d_1 and d_2 are both connected to a_1 (in case (i), this is immediate and in cases (ii) and (iii), Local Sparsity Lemma imposes it). Since the vertices $a_4, a_5 \in A$ are 3-connected to the stable D, we obtain that the subgraph induced by $\{a_1, a_4, a_5\}$ and D violates Local Sparsity Lemma, which is a contradiction.

To conclude, assertion (7) of Ad hoc Repartitioning Lemma is established. For this, suppose that no quadruplet exists in (A, B, C, D). According to the previous discussion, any vertex of the subgraph induced by these four stables is exactly 3-connected to a stable and not 2-connected to another stable. In each stable A, B, C or D, two pairs of vertices are 3-connected to the same stable. Assume w.l.o.g. that the two vertices $a_1, a_5 \in A$ are 3-connected to the stable B with b_1, b_2, b_3 connected to a_1 and b_3, b_4, b_5 connected to a_5. As a_1 and a_5 can not be 2-connected to C or D, at least three vertices of C and three vertices of D are not connected to a_1 and a_5. Assume w.l.o.g. that these vertices are respectively c_2, c_3, c_4 and d_2, d_3, d_4 (see Figure 15). According to Triplet Lemma, a triplet exists with a vertex in $\{b_1, b_2, b_4, b_5\}$, a vertex in $\{c_2, c_3, c_4\}$ and a vertex in $\{d_2, d_3, d_4\}$. If the triplet in question is composed of one of the two vertices b_1 or b_2 (resp. b_4 or b_5), then it induces a quadruplet with the vertex a_5 (resp. a_1). In both cases, (A, B, C, D) contains a quadruplet, which contradicts our first hypothesis and complete the proof of assertion (7).

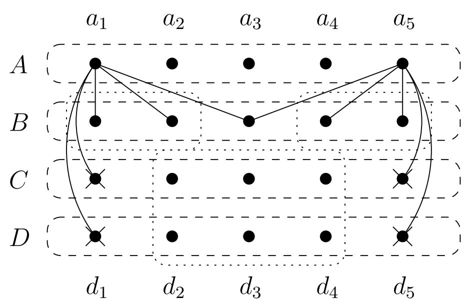

Fig. 15. The proof of assertion (7): the epilogue.

4.3 Discussion and conjectures

An interesting question remains open: does the repartitioning property hold for chordal graphs when $k \geq 5$? Having found no counterexample going against a positive answer, we emit the following conjecture.

Conjecture 4.11 *Let G be a chordal graph and k an integer. If G admits a coloring such that each color appears at least k times, then $\chi(G, k) = \lceil n/k \rceil$.*

Besides, we have verified the validity of this assertion for two subclasses of chordal graphs: forests and split graphs. A forest is an acyclic graph and a split graph is a graph whose vertices admit a partition into a stable and a clique (see [6,17] for more details). The result about forests is due to Baker and Coffman [2]; the polynomial-time algorithm proposed by the two authors for solving MES problem restricted to forests or trees relies on this result. The proof given here is more direct than the original.

Proposition 4.12 (Baker and Coffman, 1996) *Let G be a forest and k*

an integer. If G admits a coloring such that each color appears at least k times, then $\chi(G,k) = \lceil n/k \rceil$.

Proof. Let $F = (X, Y, E)$ be a forest given as a bipartite graph with $|X| \geq k$ and $|Y| \geq k$. We can consider that $|X| = k + \beta_x$ and $|Y| = k + \beta_y$ with $1 \leq \beta_x, \beta_y \leq k - 1$ (otherwise, extract stables of size exactly k in X and Y while this condition is not satisfied). If $\beta_x + \beta_y > k$, then F admits a trivial minimum partition into four stables of size at most k. Otherwise, we show that we can always extract a stable of size exactly $\beta_x + \beta_y$ from F and then obtain a partition into three stables of size at most k.

When $\beta_x + \beta_y \leq k$, we have that $\min\{\beta_x, \beta_y\} \leq k/2$. Assume w.l.o.g. that $\beta_x \leq k/2$ and consider three disjoint sets $X_1 \cup X_2 \cup X_3 \subseteq X$, each one of size β_x. If no stable exists in F composed of β_x vertices in X_i $(1 \leq i \leq 3)$ and β_y vertices in Y, then the sum of degrees of vertices of X_i must be greater than $k+1$. Indeed, if this is not the case, β_y vertices of Y are connected to no vertex of X_i (because $Y = k+\beta_y$) and the desired stable exists. Hence, we obtain the following contradiction: the number of edges in F must be greater than $3k+3$, whereas Local Sparsity Lemma imposes no more than $2k+\beta_x+\beta_y-1 \leq 3k-1$ edges. $\square$

Proposition 4.13 *Let G be a split graph and k an integer. If G admits a coloring such that each color appears at least k times, then $\chi(G,k) = \lceil n/k \rceil$.*

Proof. Consider a partition of a split graph such that all stables have a size at least k. Since the graph is split, all the vertices which do not belong to a maximum clique induce necessarily a stable. Then, for each vertex of the maximum clique, we can extract a stable of k which contains this vertex. The remaining vertices, inducing a stable, can be partitioned in an optimal way. $\square$

Corollary 4.14 *The MES problem for forests and split graphs is solved in linear time and space, given a coloring of the graph where each color appears at least k times in input.*

To confirm the previous conjecture requires to extend assertions (3) and (7) of Ad hoc Repartitioning Lemma to the case $k = 5$, which corresponds to answer to the following question: can we extract from five disjoint stables of size six one stable having a vertex in each stable? The demonstration of assertion (7), which is long and fastidious, does not not seem to be extendable.

5 Conclusion

The following tables summarize all the results presented throughout the paper about the repartitioning property.

	Claw-free graphs	Interval graphs	Circular-arc graphs
$k \geq 2$	$O(n^2/k)$	$O(n+m)$	$O(n+m)$

	Proper tolerance graphs	Tolerance graphs
$k = 2$	$O(n+m)$	counterexample
$k \geq 3$	*open*	counterexample

	Forests	Split graphs	Chordal graphs
$k \leq 4$	$O(n+m)$	$O(n+m)$	$O(n+m)$
$k \geq 5$	$O(n+m)$	$O(n+m)$	*open*

Contrary to the case of chordal graphs, we think that some bounded unit tolerance graphs exist for which the repartitioning property does not hold. To conclude, a stronger conjecture is proposed, which tends to unify all the results of the paper. Indeed, the condition enounced in this second conjecture, derived from Local Sparsity Lemma, holds for claw-free graphs, circular-arc graphs and chordal graphs.

Conjecture 5.1 *The repartitioning property holds for graphs such that any subgraph induced by two disjoint stables A and B contains no more than $|A| + |B|$ edges.*

Acknowledgements

This paper is dedicated to Prof. Michel Van Caneghem from Laboratoire d'Informatique Fondamentale, Marseille, France, and the BAMBOO team of the firm Experian-Prologia SAS, Marseille, France, in memory of the great work done on workforce planning, at the root of my PhD thesis. We also express our gratitude to the two anonymous referees for their reviewing, which has resulted in an improved paper.

References

[1] N. Alon (1983). A note on decomposition of graphs into isomorphic matchings. *Acta Mathematica Hungarica* 42, pp. 221–223.

[2] B.S. Baker and E.G. Coffman, Jr. (1996). Mutual exclusion scheduling. *Theoretical Computer Science* 162, pp. 225–243.

[3] Bamboo, edited by Experian-Prologia SAS, Marseille, France. `http://www.experian-prologia.fr/bamboo/bamboo.html`

[4] H.L. Bodlaender and F.V. Fomin (2004). Equitable colorings of bounded treewidth graphs. In *Proceedings of MFCS 2003, the 29th International Symposium on Mathematical Foundations of Computer Science* (J. Fiala, V. Koubek and J. Kratochvíl, eds) . *Lecture Notes in Computer Science* 3153, pp. 180–190. Springer-Verlag, Berlin, Germany.

[5] H.L. Bodlaender and K. Jansen (1995). Restrictions of graph partition problems. Part I. *Theoretical Computer Science* 148, pp. 93–109.

[6] A. Brandstädt, V.B. Le and J.P. Spinrad (1999). *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics and Applications 3, SIAM Publications, Philadelphia, PA.

[7] E. Cohen and M Tarsi (1991). $\mathcal{NP}$-completeness of graph decomposition problem. *Journal of Complexity* 7, pp. 200–212.

[8] D.G. Corneil, H. Kim, S. Natarajan, S. Olariu and A. Sprague (1995). Simple linear time recognition of unit interval graphs. *Information Processing Letters* 55, pp. 99–104.

[9] R. Diestel (1991). *Graph Theory.* Graduate Texts in Mathematics 173, Springer-Verlag, New York, NY. (2nd edition)

[10] E. Dahlhaus and M. Karpinski (1998). Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Applied Mathematics* 84, pp. 79–91.

[11] R. Faudree, E. Flandrin and Z. Ryjàček (1997). Claw-free graphs – A survey. In *Proceedings of the 2nd Krakow Conference of Graph Theory* (sous la direction de A. Rycerz, A.P. Wojda and M. Wozniak), *Discrete Mathematics* 164, pp. 87–147.

[12] P.C. Fishburn (1985). *Interval Orders and Interval Graphs.* Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons, New York, NY.

[13] F. Gardi (2004). A sufficient condition for optimality in mutual exclusion scheduling for interval graphs and related classes. In *Abstracts of the 9th International Workshop on Project Management and Scheduling* (A. Oulamara and M.-C. Portmann, eds), pp. 154–157. Nancy, France.

[14] F. Gardi (2005). Ordonnancement avec exclusion mutuelle par un graphe d'intervalles ou d'une classe apparentée: complexité et algorithmes. PhD Thesis. Université de la Méditerranée – Aix-Marseille II, Marseille, France. `http://www.lif-sud.univ-mrs.fr/~gardi/downloads/These_Gardi.pdf`

[15] F. Gardi (2006). Mutual exclusion scheduling with interval graphs or related classes: complexity and algorithms. *4OR, A Quarterly Journal of Operations Research* 4(1), pp. 87–90. (PhD Abstract)

[16] F. Gardi (2006). Mutual exclusion scheduling with interval graphs or related classes. Part I. (submitted to *Discrete Applied Mathematics*)

[17] M.C. Golumbic (1980). *Algorithmic Graph Theory and Perfect Graphs.* Computer Science and Applied Mathematics Series, Academic Press, New York, NY.

[18] M.C. Golumbic and A.N. Trenk (2004). *Tolerance Graphs.* Cambridge Studies in Advanced Mathematics 89, Cambridge University Press, Cambridge, UK.

[19] M. Grötschel, L. Lovász and A. Schrijver (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1, pp. 169–197.

[20] M. Habib, R.M. McConnel, C. Paul and L. Viennot (2000). Lex-BSF and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science* 234, pp. 59–84.

[21] P. Hansen, A. Hertz and J. Kuplinski (1993). Bounded vertex colorings of graphs. *Discrete Mathematics* 111, pp. 305–312.

[22] I.J. Holyer (1981). The NP-completeness of edge colorings. *SIAM Journal on Computing* 10, pp. 718–720.

[23] W.-L. Hsu (1981). How to color claw-free perfect graphs. *Annals of Discrete Mathematics* 11, pp. 189–197.

[24] K. Jansen (2003). The mutual exclusion scheduling problem for permutation and comparability graphs. *Information and Computation* 180(2), pp. 71–81.

[25] J. Jarvis and B. Zhou (2001). Bounded vertex coloring of trees. *Discrete Mathematics* 232, pp. 145–151.

[26] Z. Lonc (1991). On complexity of some chain and antichain partition problems. In *Proceedings of WG'91, the 17th International Workshop on Graph–Theoretic Concepts in Computer Science* (G. Schmidt and R. Berghammer, eds) . *Lecture Notes in Computer Science* 570, pp. 97–104. Springer-Verlag, Berlin, Germany.

[27] F. Maffray (1992). Kernels in perfect line-graphs. *Journal of Combinatorial Theory Series B* 55, pp. 1–8.

[28] F.S. Roberts (1976). *Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems.* Prentice-Hall, Englewood Cliffs, NJ.

[29] F.S. Roberts (1978). *Graph Theory and its Application to the Problems of Society*. CBMS-NSF Regional Conference Series in Applied Mathematics 29, SIAM Publications, Philadelphia, PA.

[30] L.E. Trotter (1977). Line perfect graphs. *Mathematical Programming* 12, pp. 255–259.

[31] D. De Werra (1978). On line perfect graphs. *Mathematical Programming* 15, pp. 236–238.

[32] D. De Werra (1985). An introduction to timetabling. *European Journal of Operational Research* 19, pp. 151–162.

[33] D. De Werra (1985). Some uses of hypergraphs in timetabling. *Asia-Pacific Journal of Operational Research* 2(1), pp. 2–12.

[34] D. De Werra (1996). Extensions of coloring models for scheduling purposes. *European Journal of Operational Research* 92, pp. 474–492.

[35] D. De Werra (1997). Restricted coloring models for timetabling. *Discrete Mathematics* 165/166, pp. 161–170.