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

Frédéric Gardi [1]

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

Abstract

In this paper, the mutual exclusion scheduling problem is addressed. 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. 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]. Several polynomial-time solvable cases significant in practice are exhibited here, for which we took care to devise simple and efficient algorithms (in particular linear-time and space algorithms). On the other hand, by reinforcing the $\mathcal{NP}$-hardness result of Bodlaender and Jansen, we obtain a more precise cartography of the complexity of the problem for the classes of graphs studied.

Key words: chromatic scheduling, bounded coloring, workforce planning, 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. Due to their numerous industrial applications, many variants of this problem have been extensively studied; the vast literature dedicated to combinatorial optimization and operations research contains a lot of references about them. The interested reader is referred to the paper of Krarup and De Werra [35] and the recent survey of Blazewicz et al. [5].

When all tasks have the same processing time, the problem in question has an elegant formulation in graph-theoretical terms. In effect, a natural way to express mutual exclusion between tasks is to define a simple and undirected graph, where each vertex represents one task and two vertices are connected by an edge if the corresponding tasks are in conflict. Then, an optimal schedule of the n tasks on k processors corresponds exactly to a minimum coloring of the conflict graph such that each color appears at most k times. In this way, Baker and Coffman [2] have called Mutual Exclusion Scheduling (shortly MES) the following combinatorial optimization 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.

Finding a minimum coloring of a graph is a celebrated $\mathcal{NP}$-hard problem [34]. Consequently, the mutual exclusion scheduling problem is $\mathcal{NP}$-hard too and the quest for a polynomial-time algorithm to solve this problem may be vain. Nevertheless, if the problem is restricted to graphs for which a minimum coloring is computed in polynomial time (for example perfect graphs), then the MES problem is not necessarily $\mathcal{NP}$-hard. Unfortunately, only few positive results have been published in this way. The problem is $\mathcal{NP}$-hard for complements of line-graphs (even for fixed $k \geq 3$) [9], for bipartite graphs and cographs [7], for interval graphs (even for fixed $k \geq 4$) [7], for complements of comparability graphs (even for fixed $k \geq 3$) [37], and for permutation graphs (even for fixed $k \geq 6$) [31]. 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 [37,7], forests and trees [2,32], collections of disjoint cliques [44], complements of strongly chordal graphs [12] and complements of interval graphs [37,7], and bounded treewidth graphs [6]. Note that Jost [33, pp. 154–164] has recently claimed that MES for complements of triangulated graphs is polynomial-time solvable.

1.2 Motivations

Among perfect graphs, interval graphs can be distinguished by their field of applications large and varied: genetic, scheduling, psychology, archaeology, etc. An interval graph is the intersection graph of a set of intervals of the real line, that is, a graph whose vertices correspond to intervals such that two vertices connected by an edge are associated to intersecting intervals (see Fig. 1).

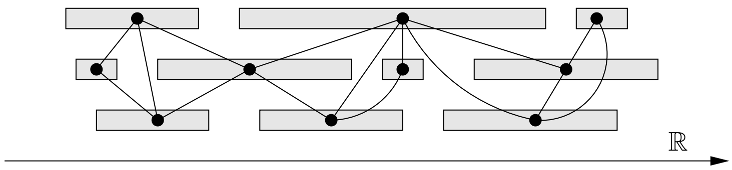

Fig. 1. An interval graph and its representation.

Here is an application related to interval graphs met during the development of the software Bamboo for workforce planning, edited by the firm Experian-Prologia SAS [3]. Let $\{T_i\}_{i=1,\dots,n}$ be a set of daily tasks to assign to employees, each task having a starting date l_i and an ending date r_i. An employee is able to execute correctly a set of tasks if they do not overlap during the day. For several reasons (regulation of work, security, maintenance of machines), an employee must not execute more than k tasks in a day (generally $k \leq 5$). Then, the question is how employees have to be mobilized to complete all the tasks? Obviously, a planning describing which tasks have to be assigned to each employee is required. Since each task is only an interval of time, the problem amounts to coloring the underlying interval graph such that each color appears no more than k times, which corresponds exactly to the *mutual exclusion scheduling problem for interval graphs*. When the planning is cyclic (the same tasks recur each day and some of them spread out over two consecutive days), we obtain the same problem but for *circular-arc graphs*. When there are not enough employees to execute all the tasks (e.g., because some employees are absent), it is interesting to allow overlaps between certain tasks during the assignment. In this case, the problem relates to *tolerance graphs*. Unfortunately, the following result of Bodlaender and Jansen [7] is a serious strike against the resolution of these workforce planning problems.

Theorem 1.1 (Bodlaender and Jansen, 1995) *For each fixed $k \geq 4$, the k-MES problem is $\mathcal{NP}$-hard for interval graphs (and also for circular-arc graphs and bounded tolerance graphs).*

The objective of this paper is to detail the complexity of mutual exclusion scheduling problem for interval graphs and related classes, in particular circular-arc graphs and tolerance graphs. Although the question of the complexity of 3-MES for interval graphs raised by Bodlaender and Jansen [7] is not answered here, the study that we have led on the subject provides several positive results. Some polynomial cases significant in practice are exhibited, for which we have been careful to devise some simple and efficient algorithms (in particular linear-time and space algorithms). On the other hand, by reinforcing the $\mathcal{NP}$-hardness result of Bodlaender and Jansen [7], we obtain a more precise cartography of the complexity of the problem for the classes of graphs studied.

First, the complexity of MES is approached for interval graphs. A new algorithm, much simpler than the previous one of [1], is proposed to solve in linear time and space the 2-MES problem for interval graphs. In addition, the problem is shown to be linear-time and space solvable for two well-known subclasses of interval graphs, namely proper interval graphs and threshold graphs. Then, the problem is investigated for the two extensions of interval graphs which are circular-arc graphs and tolerance graphs. An algorithm is proposed to solve in $O(n^2)$ time and linear space the problem restricted to proper circular-arc graphs, as well as a linear-time and space algorithm for the same problem when $k = 2$. Finally, the 3-MES problem is shown to be $\mathcal{NP}$-hard for bounded tolerance graphs, even if any cycle of length greater than or equal to five has two chords. This result has for corollary that the 3-MES problem is $\mathcal{NP}$-hard for Meyniel graphs and weakly triangulated graphs, even if their complement is transitively orientable.

All the results presented here appear in the author's thesis [20], written in French, and have been announced in [21]. A preliminary version of these results also appears in [18,19].

1.3 Interval graphs and related classes

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 (see Fig. 1). 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. 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. Comparability graphs are the transitively orientable graphs, they correspond to graphs of partial orders.

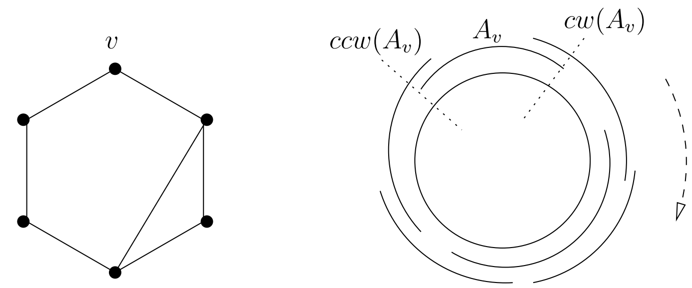

Fig. 2. A circular-arc graph and its representation around the circle.

Circular-arc graphs and tolerance graphs are two natural extensions of interval graphs. Circular-arc graphs are the intersection graphs of a set 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* $cw(A_v)$ (see Fig. 2). 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. 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*.

Proper interval graphs and threshold graphs are two subclasses of interval graphs. A graph G is a *proper interval graph* if there is an interval representation of G in which no interval properly contains another. The notion of properness is defined similarly for circular-arc graphs and tolerance graphs. A graph $G = (V, E)$ is a *threshold graph* if to each vertex $v \in V$ can be associated a positive integer a_v such that $X \subseteq V$ is an independent set if and only if $\sum_{x \in X} a_x \leq t$ with t an integer constant (called the threshold). The vertices of a threshold graph can be partitioned into a clique $C = C_1 \cup \cdots \cup C_r$ and an independent set $S = S_1 \cup \cdots \cup S_r$ ($r \leq n$ and C_i, S_i not empty for all $i = 1, \ldots, r$) such that a vertex of S_i is adjacent to a vertex of $C_{i'}$ if and only if $i' > i$ for any $i, i' \in \{1, \ldots, r\}$ (see Fig. 3).

Interval graphs and tolerance graphs are perfect, which is not true for (proper) circular-arc graphs (see [8,24]). Interval graphs, proper interval graphs, threshold graphs, circular-arc graphs and proper circular-arc graphs are recognized in linear time and space (see [8,11,13,24,29]); the complexity of recognition for tolerance graphs remains an open question. Computing a minimum coloring is done in linear time and space for interval graphs [27,28] (see also [20, pp. 42–47]) and in $O(n^2)$ time for tolerance graphs if a tolerance representation is given in input [25]. The minimum coloring problem is $\mathcal{NP}$-hard for

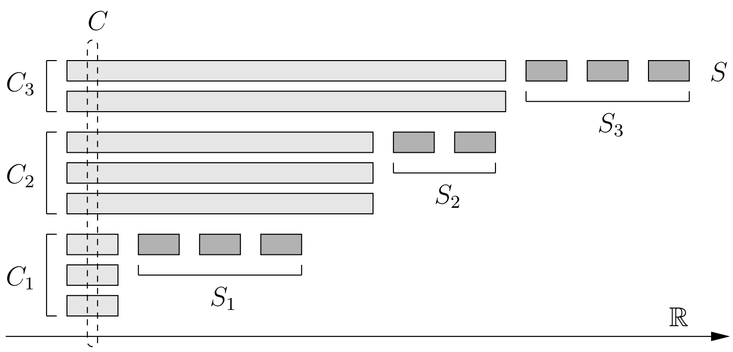

Fig. 3. An interval representation of a threshold graph.

circular-arc graphs [17]. Restricted to proper circular-arc graphs, the minimum coloring problem becomes solvable in $O(n^{1.5})$ time [41]. For more details on these graphs and their applications, the reader can consult the books of Roberts [39,40], Golumbic [24,26], Fishburn [15] and Brandstädt et al. [8].

1.4 Basic definitions and notations

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. All the graph-theoretical terms which are not defined here can be found in [8,24].

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

Finally, a *matching* in a graph is a subset of edges such that no two of them share a vertex in common. A *maximum matching* (resp. *perfect matching*) is a matching whose cardinality is as large as possible (resp. equals to $n/2$). A maximum matching corresponds in fact to a minimum partition into cliques of size at most two or a minimum partition into stables of size at most two in the complement graph.

2 Mutual exclusion scheduling with interval graphs

In this section, the linear order induced by non-decreasing left (resp. right) endpoints of a set $\mathcal{I}$ of intervals is denoted by $<_l$ (resp. $<_r$).

2.1 A new linear-time algorithm for $k = 2$

The 2-MES problem is solved in polynomial time by reducing it to the maximum matching problem in the complement graph. However, algorithms for maximum matching in general graphs are not easy to implement and their execution time is more than quadratic in the number of vertices and edges of the graph [23]. That is why designing simple and efficient matching algorithms, dedicated to certain classes of graphs, remains topical.

To our knowledge, only two algorithms have been proposed to solve the 2-MES problem restricted to the class of interval graphs. The first appears in an unpublished manuscript of M.G. Andrews and D.T. Lee. This algorithm, briefly evoked in [1], considers an interval representation as input and performs plane sweepings to build in $O(n \log n)$ time an optimal solution (even if the endpoints of the intervals are given sorted in input). The second, of a geometric nature too, is presented in the paper of Andrews et al. [1]. The authors give a parallel recursive algorithm which requires $O(\log^3 n)$ time on an EREW PRAM architecture with $O(n/\log^2 n)$ processors (see [10, pp. 675–715] for an introduction to parallel algorithms). The serial version of their algorithm runs in $O(n \log n)$ time and the authors claim that this complexity can be lowered to $O(n)$ if the endpoints of intervals are given sorted in input. Despite that, their algorithm remains complicated and the proof of its correctness is long.

In this section, a new algorithm is presented which relies on graph-theoretical concepts. This algorithm is simple, incremental and the proof of its validity is short. We show that this algorithm runs in $O(n)$ time and space if an ordered interval representation is given as input. Similar to Andrews et al. [1], our algorithm uses as subroutine an algorithm for maximum matching in convex bipartite graphs. A bipartite graph $B = (X, Y, E)$ is Y*-convex* if the vertices of Y can be ordered such that the vertices adjacent to any vertex of X appear consecutively in this order. Convex bipartite graphs have been introduced by Glover [22]; Steiner and Yeomans [42] have shown how computing a maximum matching in a convex bipartite graph in $O(|X|)$ time and $O(|Y|)$ space, if the interval of vertices of Y adjacent to each vertex of X is given in input.

2.1.1 Ingredients and correctness of the algorithm

Although we work on an open interval representation rather than on the interval graph itself, we shall keep a graph-theoretical vocabulary to describe the algorithm. Thus, a stable is defined as a set of pairwise disjoint intervals and a clique as a set of pairwise intersecting intervals. In this context, coloring a set of intervals consists of partitioning this set into subsets of disjoint intervals. By analogy with matching, we shall say that two intervals can be matched when they are disjoint.

Consider a set $\mathcal{I} = \{I_1, \ldots, I_n\}$ of intervals and a minimum partition $\mathcal{S} = \{S_1, \ldots, S_{\chi(\mathcal{I})}\}$ of $\mathcal{I}$ into stables. Here are the few assertions on which the validity of the algorithm relies.

Lemma 2.1 *If a stable $S_u \in \mathcal{S}$ contains only one interval, then this one belongs to each maximum clique of $\mathcal{I}$.*

Proof. Since interval graphs are perfect, the cardinality of any maximum clique in $\mathcal{I}$ equals the cardinality of any minimum coloring of $\mathcal{I}$ (cf. [8,24]). Then, any maximum clique must contain one and only one interval from each stable of $\mathcal{S}$. If the unique interval of S_u does not belong to a maximum clique of $\mathcal{I}$, we obtain a contradiction. $\square$

Lemma 2.2 *If all stables of $\mathcal{S}$ contains more than two intervals and that the number n of intervals in $\mathcal{I}$ is even, then $\chi(\mathcal{I}, 2) = n/2$.*

Proof. Let $S_u, S_v \in \mathcal{S}$ be two stables of size odd and greater than or equal to three. We show that it is always possible to match two intervals, the one from S_u and the other from S_v, in order to redefine two new stables of size even and greater than two. Let $I_a, I_b \in S_u$ and $I_c, I_d \in S_v$ be four intervals such that $r(I_a) \leq l(I_b)$ and $r(I_c) \leq l(I_d)$. If I_a and I_d are disjoint, then these ones forms the desired pair for matching. Otherwise, we claim that I_b and I_c are such ones. Since I_a and I_d are intersecting, we have $l(I_d) \leq r(I_a)$. Then, by using the inequalities $r(I_c) \leq l(I_d)$ and $r(I_a) \leq l(I_b)$, we obtain that $r(I_c) \leq l(I_b)$.

To conclude, the following construction establishes the lemma. Since n is even, the number of stables of odd size is necessarily even too. According to the previous property, we can redefine two by two the stables of odd size in stables of even size, while exhibiting pairs of disjoint intervals. Finally, the remaining stables, all of even size, admits a trivial partition into disjoint pairs of intervals. $\square$

Proposition 2.3 *If the number $\vartheta(\mathcal{I})$ of stables of $\mathcal{S}$ containing only one interval is as small as possible, then $\chi(\mathcal{I}, 2) = \lceil (n + \vartheta(\mathcal{I}))/2 \rceil$.*

Proof. The proposition is established thanks to the two previous lemmas. The first lemma imposes that $\chi(\mathcal{I}, 2) \geq \lceil (n - \vartheta(\mathcal{I}))/2 \rceil + \vartheta(\mathcal{I})$, since at most $\vartheta(\mathcal{I})$ intervals of $\mathcal{I}$ can not be matched. Having extracted these ones, the second lemma enables us to obtain a perfect matching among the $n - \vartheta(\mathcal{I})$ remaining intervals (minus one if this number is odd). $\square$

According to this proposition, the 2-MES problem is reduced to finding a coloring of the set $\mathcal{I}$ such that the number of stables of size one is as small as possible. By Lemma 2.1, this new problem is solved by computing a maximum disjoint matching $\mathcal{M}_c$ between the intervals of a maximum clique and the rest of the intervals. In effect, having computed this matching, a procedure COMPLETE-STABLES is used to minimize $\vartheta(\mathcal{I})$ by adding to each stable $S_u = \{I_i\}$ of size one the interval $I_j \in S_v$ if the pair (I_i, I_j) belongs to $\mathcal{M}_c$. Hence, an optimal solution to the 2-MES problem is obtained by applying the constructive proof of Lemma 2.2. Here we describe the complete algorithm.

Algorithm 2-MES-INTERVALS;
Input: a set $\mathcal{I} = \{I_1, \ldots, I_n\}$ of intervals;
Output: an optimal solution $\mathcal{M}$ to the 2-MES problem for $\mathcal{I}$;
Begin;
stage 1:
  compute a minimum coloring $\mathcal{S} = \{S_1, \ldots, S_{\chi(\mathcal{I})}\}$ of $\mathcal{I}$;
  if all the stables of $\mathcal{S}$ have a size at most two **then goto** stage 3;
  if all the stables of $\mathcal{S}$ have a size at least two **then goto** stage 3;
stage 2:
  compute a maximum clique $C = \{c_1, \ldots, c_{\chi(\mathcal{I})}\}$ of $\mathcal{I}$;
  build the bipartite graph $B_c = (X, Y, E)$ such that:
    . $X = C$ and $Y = \mathcal{I} \setminus C$;
    . $E = \{(I_i, I_j) \mid I_i \in C, I_j \in \mathcal{I} \setminus C \text{ and } I_i \cap I_j = \emptyset\}$;
  compute a maximum matching $\mathcal{M}_c$ in B_c;
  $\mathcal{S} \leftarrow$ COMPLETE-STABLES$(\mathcal{S}, \mathcal{M}_c)$;
stage 3:
  $\mathcal{M} \leftarrow \emptyset$;
  for each stable $S_u \in \mathcal{S}$ of size one **do**
    remove S_u from $\mathcal{S}$ and add it to $\mathcal{M}$;
  if the number of remaining intervals in $\mathcal{S}$ is odd **then**
    remove one interval from any stable of odd size and add it to $\mathcal{M}$;
  compute a perfect disjoint matching in $\mathcal{S}$ and add it to $\mathcal{M}$;
  return $\mathcal{M}$;
End;

2.1.2 Complexity of the algorithm

An interval representation $\mathcal{I} = \{I_1, \ldots, I_n\}$ with the two orders $<_l$ and $<_r$ on $\mathcal{I}$ is assumed to be given in input. These two orders allow us to obtain in $O(n)$

time and space the list of the n intervals ordered according to $<_l$ or $<_r$.

The complexity of stage 1 is dominated by the complexity of computing a minimum coloring of $\mathcal{I}$. Since the orders $<_l$ and $<_r$ are given, this computation can be done in $O(n)$ time and space [27] (see also the new coloring algorithm given in [20, pp. 42–47]). Now, the complexity of stage 2 relies on the following property.

Lemma 2.4 *The bipartite graph B_c is Y-convex.*

Proof. Recall that the set X corresponds to a maximum clique $C \in \mathcal{I}$ and the set Y to $\mathcal{I} \setminus C$. This second set is subdivided into two disjoint sets $\mathcal{I}_l$ and $\mathcal{I}_r$, respectively the set of intervals on the left of C and the set of intervals on the right of C (an interval can not belong to the one and the other without belonging to the clique). Having ordered $\mathcal{I}_r$ according to $<_l$ and $\mathcal{I}_l$ according to $<_r$, the linear order on Y is obtained by concatenating the two sets $\mathcal{I}_r$ and $\mathcal{I}_l$ such that the last interval of $\mathcal{I}_r$ appears before the first interval of $\mathcal{I}_l$ in the order. Now, for each $c_u \in C$, set $a_u = \min\{i \,|\, r(c_u) \geq l(I_i) \text{ and } I_i \in \mathcal{I}_r\}$ and $b_u = \min\{i \,|\, r(I_i) \geq l(c_u) \text{ and } I_i \in \mathcal{I}_l\}$. Then, it is easy to verify that for any $i \in \{a_u, \ldots, b_u\}$, the intervals c_u and I_i are disjoint (see Fig. 4). Consequently, the bipartite graph B_c is Y-convex. $\square$

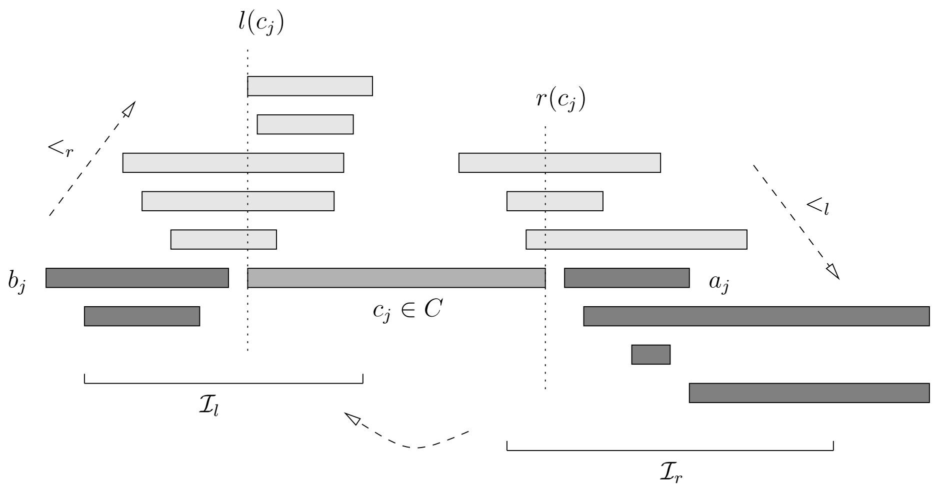

Fig. 4. The proof of Lemma 2.4.

Since the bipartite graph B_c is convex, a maximum matching $\mathcal{M}_c$ can be computed in $O(n)$ time by using the algorithm of Steiner and Yeomans [42]. Their algorithm requires as input the following representation of B_c: the linear order on Y and for each $u \in X$, the two values a_u and b_u. Here we describe how we efficiently compute this representation. When the intervals are ordered, a maximum clique is obtained in $O(n)$ time and space [28] (see also the algorithm given in [20, pp. 42–47]). This maximum clique, denoted by $C = \{c_1, \ldots, c_{\chi(\mathcal{I})}\}$, is defined in such a way that c_u corresponds to the interval of C which belongs to the stable $S_u \in \mathcal{S}$. Then, the linear order on Y (as defined in Lemma 2.4) is obtained in $O(n)$ time thanks to orders $<_l$ and $<_r$. Finally, the indices a_u of the intervals $c_u \in C$ are determined by sweeping the set $\mathcal{I}_r$ ordered according to $<_l$, provided that the intervals of C are ordered according to $<_r$ (because $c_u <_r c_{u'}$ implies that $a_u \leq a_{u'}$). Obviously, the indices b_u can be determined in a symmetric way by sweeping the set $\mathcal{I}_l$, which completes the construction of the bipartite graph B_c.

To conclude the analysis of stage 2, an implementation of Complete-Stables is given whose execution time is linear. The size of each stable is assumed to be computed in $O(1)$ time, just as, for an interval, the index of the stable to which it belongs. $\mathcal{M}_c$ is considered as an array in which is stored at u the index of the interval of $\mathcal{I} \setminus C$ matched to $c_u \in C$ (or zero if this one is unmatched). Observe that once a stable S_u is removed from $\mathcal{S}'$, it can not be added to this set in the next iterations of the loop, which ensures a linear running time.

Algorithm Complete-Stables;
Input: a minimum coloring $\mathcal{S}$ of $\mathcal{I}$, a maximum matching $\mathcal{M}_c$ of B_c;
Output: a minimum coloring $\mathcal{S}$ such that $\vartheta(\mathcal{I})$ is minimum;
Begin;
  $\mathcal{S}' \leftarrow \emptyset$;
  for each stable $S_u \in \mathcal{S}$ **do**
    if S_u is of size one **then** $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S_u\}$;
  while $\mathcal{S}' \neq \emptyset$ **do**
    $\mathcal{S}' \leftarrow \mathcal{S}' \setminus \{S_u\}$;
    let i be the index stored at u in $\mathcal{M}_c$;
    if $i \neq 0$ **then**
      let v be the index of the stable to which I_i belongs;
      $S_v \leftarrow S_v \setminus \{I_i\}$, $S_u \leftarrow S_u \cup \{I_i\}$;
      if S_v is of size one **then** $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S_v\}$;
  return $\mathcal{S}$;
End;

Finally, stage 3 takes $O(n)$ time too. The proof of Lemma 2.2 provides a simple linear-time algorithm to compute a perfect disjoint matching, since $\mathcal{S}$ contains only stables of size at least two and an even number of intervals. The space used all along the algorithm not exceeding $O(n)$ (even during the execution of the Steiner-Yeomans algorithm [42]), we hold the following result.

Proposition 2.5 *The algorithm* 2-MES-Intervals *computes in $O(n)$ time and space an optimal solution to the 2-MES problem given a set $\mathcal{I}$ of n intervals and the orders $<_l$ and $<_r$ on $\mathcal{I}$ in input.*

Since an ordered interval representation (according to $<_l$ or $<_r$) is computed in linear time and space given an interval graph or its complement [29], we obtain the following corollary.

Corollary 2.6 *The 2-MES problem (resp. the maximum matching problem) is solved in linear time and space for interval graphs (resp. complements of interval graphs).*

2.2 Linear-time algorithms for subclasses of interval graphs

Baker and Coffman [2] gave an $O(k^2 \log k + n)$-time algorithm to solve MES for forests. Since any bipartite interval graph is isomorphic to a forest (observe that such a graph contains no induced cycle of length greater than or equal to three), we obtain the following result.

Proposition 2.7 (Baker and Coffman, 1996) *The MES problem is solved in $O(k^2 \log k + n)$ time for bipartite interval graphs.*

In this section, we present linear-time algorithms to solve MES restricted to two other subclasses of interval graphs, namely proper interval graphs and threshold graphs.

2.2.1 The case of proper interval graphs

In their paper, Andrews et al. [1] propose a simpler algorithm to determine a maximum disjoint matching in a set of intervals when these intervals are proper. Their algorithm runs in $O(\log n)$ time on an EREW PRAM with $O(n/\log n)$ processors, if the endpoints of intervals are given sorted in input. Its serialization takes $O(n)$ time and space under the same conditions. Here we go beyond this result by presenting a greedy algorithm to solve the MES problem for proper interval graphs in linear time and space, for any value of k. For the sake of simplicity, the intervals and the stables are numbered from zero. The set $\mathcal{I}$ of open intervals is assumed to be ordered according to $<_l$ in input.

Algorithm MES-Proper-Intervals;
Input: an ordered set $\mathcal{I} = \{I_0, \ldots, I_{n-1}\}$ of proper intervals, an integer k;
Output: an optimal solution $\mathcal{S}$ to the MES problem for $\mathcal{I}$;
Begin;
  compute $\omega(\mathcal{I})$;
  $\chi(\mathcal{I}, k) \leftarrow \max(\omega(\mathcal{I}), \lceil n/k \rceil)$;
  $S_0 \leftarrow \cdots \leftarrow S_{\chi(\mathcal{I},k)-1} \leftarrow \emptyset$;
  for i from 0 to $n-1$ **do**
    $u \leftarrow i \bmod \chi(\mathcal{I}, k)$, $S_u \leftarrow S_u \cup \{I_i\}$;
  return $\mathcal{S} \leftarrow \{S_1, \ldots, S_{\chi(\mathcal{I},k)}\}$;

End;

Proposition 2.8 *The algorithm* MES-PROPER-INTERVALS *computes in $O(n)$ time and space an optimal solution to the MES problem, given a set $\mathcal{I}$ of n proper intervals ordered according to $<_l$ in input.*

Proof. The size $\omega(\mathcal{I})$ of a maximum clique of $\mathcal{I}$ is obtained in $O(n)$ time and space [28] (see also the algorithm given in [20, pp. 42–47]). Then, the remainder of the algorithm runs in $O(n)$ time and space. To conclude, we prove that the output set $\mathcal{S}$ of stables is an optimal solution to the MES problem.

First, we claim that the stables $S_0, \ldots, S_{\chi(\mathcal{I},k)-1}$ are all of size at most k. According to the algorithm, the sizes of two stables differ from at most one. Thus, the existence of a stable of size strictly greater than k implies $n > k \cdot \chi(\mathcal{I}, k)$, which is a contradiction. Now, suppose that two intervals $I_i, I_j \in S_u$ with $i < j$ are intersecting. According to the algorithm, we have $i = u + \alpha \cdot \chi(\mathcal{I}, k)$ and $j = u + \beta \cdot \chi(\mathcal{I}, k)$ with $\alpha < \beta$. When the intervals are proper, the left endpoints appear in the same order than the right endpoints. Hence, the intervals $I_i, I_{i+1}, \ldots, I_{j-1}, I_j$ contain all the portion $]l(I_j), r(I_i)[$ of the line, inducing a clique of size $j - i + 1 = (\beta - \alpha) \cdot \chi(\mathcal{I}, k) + 1 > \chi(\mathcal{I}, k)$, which is in contradiction with the hypothesis. Consequently, the set $\mathcal{S}$ forms a partition of $\mathcal{I}$ into stables of size at most k; since $\max(\omega(\mathcal{I}), \lceil n/k \rceil)$ is a lower bound for $\chi(\mathcal{I}, k)$, this one has a minimum cardinality. $\square$

Corollary 2.9 *Let G be a proper interval graph. Then, the equality*

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

holds for all integers $k \geq 1$.

Since an ordered proper interval representation is computed in linear time and space given a proper interval graph [11], we obtain the following corollary.

Corollary 2.10 *The MES problem is solved in linear time and space for proper interval graphs.*

2.2.2 The case of threshold graphs

The threshold graphs form a subclass of interval graphs, but also of split graphs. A *split graph* is a graph whose vertices admit a partition into two subsets S and C, where S is a stable and C a clique. By analogy with bipartite graphs, such a graph is denoted $G = (S, C, E)$; we shall write $s = |S|$ and $c = |C|$. For more details on split graphs, the reader is referred to [24, pp. 149–156]. Independently, Lonc [37] and Bodlaender and Jansen [7] have shown that

MES becomes polynomial-time solvable when restricted to split graphs. Hence, MES is solvable in polynomial time for threshold graphs too. Having reminded the result of Lonc [37], we show that linear time and space suffice to solve the MES problem for threshold graphs, and even for a larger class which is called convex split graphs (by analogy with convex bipartite graphs).

The algorithms given by Lonc [37] and Bodlaender and Jansen [7] are based on the following observation, where $\vartheta(G)$ denotes the maximum number of vertices of S which belong to disjoint stables of size at most k and containing each one a vertex of C.

Proposition 2.11 *Let $G = (S, C, E)$ be a split graph. Then, the equality*

$$\chi(G, k) = c + \lceil (s - \vartheta(G))/k \rceil$$

holds for all integers $k \geq 1$.

Proof. The vertices of C must be placed in different stables, which implies that $\chi(G, k) \geq c$. Having extracted the $\vartheta(G)$ vertices which belong to disjoint stables of size at most k and containing each one a vertex of C, the remaining vertices of S can be grouped in $\lceil (s - \vartheta(G))/k \rceil$ stables of size at most k. $\square$

Lonc [37] shows that the number $\vartheta(G)$ corresponds to the size of a maximum matching in a certain bipartite graph B_l, obtained as follows: replace each vertex $u \in C$ by $k-1$ vertices $u_1, \ldots, u_{k-1}$ and join each one to all vertices in S not connected to u, then remove all the old edges of the graph (including those of C). Having a maximum matching in B_l, an optimal partition of G into stables of size at most k is computed as follows. First, for each vertex $u \in C$, define one stable containing u and the vertices of S matched to vertices $u_1, \ldots, u_{k-1}$. Then, partition in an optimal way the set of vertices remaining in S.

By analogy with bipartite graphs, a split graph $G = (S, C, E)$ is S-*convex* if the vertices of S admit a linear order such that for all $i \in C$, the vertices of S connected to $i \in C$ appear consecutively in this order. A S-convex representation of G is given by the order on the vertices of S and for each vertex $i \in C$, two values a_i and b_i, respectively the index of the first and the index of the last vertices in the (ordered) interval of vertices adjacent to i.

Proposition 2.12 *The MES problem is solved in $O(n)$ time and space for S-convex split graphs, given a S-convex representation of the graph in input.*

Proof. The proof relies on the fact that the bipartite graph B_l defined by Lonc [37] becomes circular-convex in the case of S-convex split graphs. Circular-convex bipartite graphs are a natural extension of convex bipartite graphs where the notion of convexity is represented by arcs around the circle instead of intervals of the line [36]. Indeed, let $G = (S, C, E)$ be a S-convex split graph with $<$ the linear order on S. The vertices of S connected to any vertex $i \in C$ appear consecutively in the order $<$. By extending $<$ to a circular order $<_c$ (the first vertex of S in the order $<$ becomes the successor of the last vertex of S in this same order), the vertices of S not connected to i appear consecutively in the order $<_c$.

A circular-convex representation of B_l is obtained from the S-convex representation of G by sweeping the vertices of C. Liang and Blum [36] have shown that a maximum matching in the bipartite graph B_l can be determined by using two passes of Glover's heuristic for maximum matching in convex bipartite graphs [22]. By modifying Glover's heuristic in order to allow the selection of $k-1$ incident edges for each vertex $i \in C$ (and not only one), Lonc's algorithm can be simulated in $O(s + c)$ time and space without explicitly constructing the graph B_l (indeed, the number of vertices selected to match with vertices of C remains lower than s). $\square$

Since a S-convex representation is computed in linear time and space by using a recognition algorithm for the consecutive ones property [29], we have the following corollary.

Corollary 2.13 *The MES problem is solved in linear time and space for S-convex split graphs.*

According to the following lemma, threshold graphs form a very special class of convex split graphs.

Lemma 2.14 *Any threshold graph $G = (S, C, E)$ is a S-convex and C-convex split graph. Moreover, the vertices of S can be ordered such that for all $i \in C$ connected to at least one vertex of S, we have $a_i = 1$.*

The proof is simply derived from the definition of threshold graphs given in introduction. Then, the reader shall notice that for threshold graphs, a simple linear sweeping of the vertices of C and S in the order suffices to compute an optimal solution to the MES problem.

Corollary 2.15 *The MES problem is solved in linear time and space for threshold graphs.*

3 Mutual exclusion scheduling with circular-arc graphs

The result of Bodlaender and Jansen (Theorem 1.1) seems to condemn the quest of a polynomial-time algorithm for MES restricted to circular-arc graphs (except for the case $k = 3$ whose complexity remains unknown). In this section, the problem is approached for a natural subclass of circular-arc graphs, namely proper circular-arc graphs. This class encompasses proper interval graphs, studied in the previous section, and unit circular-arc graphs, (i.e., the graphs having a circular-arc representation in which all arcs have the same length). We show that MES is solvable in $O(n^2)$ time and linear space when restricted to proper circular-arc graphs, and even in linear time and space in the case $k = 2$. Note that all the circular-arcs considered throughout the section are open.

3.1 A quadratic-time algorithm for the general case

The algorithm relies on the paradigm of bichromatic exchange of vertices, particularly employed by De Werra [43] in the context of edge-coloring and timetabling problems.

Lemma 3.1 *Let G be a proper circular-arc graph and S_u, S_v two disjoint stables of G. If the stables S_u and S_v have different sizes, then any connected component of the bipartite graph induced by these two stables is isomorphic to a chain.*

Proof. Since G is a proper circular-arc graph, this one can not contain $K_{1,3}$ as an induced subgraph. From this point we deduce that any connected component of the bipartite graph induced by S_u and S_v is isomorphic to a chain or an even cycle (all vertices of the bipartite graph has degree at most two). Now, consider an even cycle C in a proper circular-arc representation of this bipartite graph. Clearly, the arcs corresponding to the vertices of the cycle C must cover the entire circle (see Fig. 5).

Since the stables S_u and S_v have different sizes, assume without loss of generality that $|S_u| > |S_v| > 1$. Clearly, a vertex of S_u exists which does not belong to the cycle C. Now, the arc corresponding to this vertex is necessarily inserted between two arcs of the stable S_u which belong to the cycle C. Consequently, this one is entirely covered by an arc of S_v, which contradicts the fact that the arcs are proper. Thereby, any connected component of the bipartite graph is isomorphic to a chain. $\square$

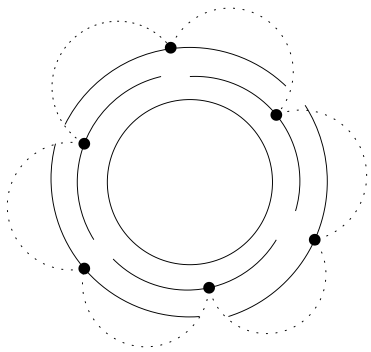

Fig. 5. A proper circular-arc representation of the even cycle C_6.

Lemma 3.2 *Let G be a proper circular-arc graph and k a positive integer. Then, a minimum coloring of G exists which satisfies one of the two following assertions: (a) each color appears at least k times, (b) each color appears at most k times.*

Proof. Let $\mathcal{S} = \{S_1, \ldots, S_{\chi(G)}\}$ be a minimum coloring of G. We show that if the coloring $\mathcal{S}$ does not the satisfy the conditions (a) or (b), then the algorithm described below brings us back to one of these cases. In this algorithm, the procedure Connected-Components is employed which returns the set $\mathcal{B}$ of connected components of the bipartite graph induced by two disjoint stables S_u, S_v of G. For each connected component $B_r \in \mathcal{B}$, we are able to access to the set B_r^u (resp. B_r^v) of vertices of B_r which belong to S_u (resp. S_v).

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;

The correctness of this algorithm is established. Having determined the connected components of the bipartite graph induced by the stables S_u and S_v, we claim that one component $B_r \in \mathcal{B}$ exists such that $|B_r^u| = |B_r^v| + 1$ while $|S_u| > k$ and $|S_v| < k$. According to Lemma 3.1, each connected component B_r meets one of the three following conditions: (i) B_r is an odd chain and $|B_r^u| = |B_r^v|$, (ii) B_r is an even chain and $|B_r^u| + 1 = |B_r^v|$, (iii) B_r is an even chain and $|B_r^u| = |B_r^v| + 1$. Since the inequalities $|S_u| > k$ and $|S_v| < k$ impose that $|S_u| \geq |S_v| + 2$, at least two connected components of the bipartite graph must satisfy the condition (iii), which justifies our claim. Finally, at the end of each outer while loop, the size of one stable of $\mathcal{S}$ is fixed to k. Thus, after at most $\chi(G)$ loops, the algorithm returns a minimum coloring satisfying one of the two conditions of the lemma. $\square$

Now, we are able to describe the complete algorithm for solving the MES problem for proper circular-arc graphs. For the sake of simplicity, the arcs and the stables are numbered from zero. The set $\mathcal{A}$ of proper arcs is assumed to be arranged according to the circular order in input.

Algorithm MES-Proper-Circular-Arcs;
Input: an ordered set $\mathcal{A} = \{A_0, \dots, A_{n-1}\}$ of proper arcs, an integer k;
Output: an optimal solution $\mathcal{S}^*$ to the MES problem for $\mathcal{A}$;
Begin;
  compute a minimum coloring $\mathcal{S} = \{S_0, \dots, S_{\chi(\mathcal{A})-1}\}$ of $\mathcal{A}$;
  $\mathcal{S} \leftarrow$ Refine-Coloring$(\mathcal{S}, k)$, $\mathcal{S}^* \leftarrow \emptyset$;
  if all the stables of $\mathcal{S}$ have a size at most k **then** $\mathcal{S}^* \leftarrow \mathcal{S}$;
  else
    if n is not a multiple of k **then**
      extract from any stable of $\mathcal{S}$ one stable S' of size $n \bmod k$;
      add S' to $\mathcal{S}^*$ and remove from $\mathcal{A}$ the arcs of S' ($n \leftarrow n - n \bmod k$);
      number the remaining arcs in $\mathcal{A}$ from 0 to $n-1$ in the circular order;
    $S_0 \leftarrow \dots \leftarrow S_{n/k-1} \leftarrow \emptyset$;
    for i from 0 to $n-1$ **do**
      $u \leftarrow i \bmod n/k$, $S_u \leftarrow S_u \cup \{A_i\}$;
    $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \{S_0, \dots, S_{n/k-1}\}$;
  return $\mathcal{S}^*$;
End;

According to Lemma 3.2, two different cases arise having refined the coloring $\mathcal{S}$: (a) all the stables have a size lower than k (i.e., $\chi(\mathcal{A}) \geq \lceil n/k \rceil$), (b) all the stables have a size greater than k and at least one has a size strictly greater than k (i.e., $\chi(\mathcal{A}) < \lceil n/k \rceil$). In the case (a), the coloring $\mathcal{S}$ forms a trivial solution to the MES problem. Now, let us analyse the case (b). First, note that extracting one stable S' of size $n \bmod k < k$ from any stable of $\mathcal{S}$ is possible since all have a size greater than k. In the same way, it is easy to verify that all the stables $S_0, \dots, S_{n/k-1}$ have a size at most k by construction. Now, suppose that two arcs $A_i, A_j \in \mathcal{A}$ are intersecting in a stable S_u ($0 \leq u \leq n/k - 1$). According to the algorithm, we have $i = u + \alpha \cdot n/k$ and $j = u + \beta \cdot n/k$ with $\alpha \neq \beta$. Here we consider the case $\alpha < \beta$; the proof of the other case is similar. When the arcs are proper, clockwise endpoints appear in the same order as counterclockwise endpoints. Hence, all the arcs $A_i, A_{i+1}, \dots, A_{j-1}, A_j$ contain the portion $]ccw(A_j), cw(A_i)[$ of the circle, inducing a clique of size $j - i + 1 = (\beta - \alpha) \cdot n/k + 1 > \chi(\mathcal{A})$, which is a contradiction. Consequently, the set $\mathcal{S}^*$ forms a partition of $\mathcal{A}$ into stables of size at most k; since $\lceil n/k \rceil$ is a lower bound for $\chi(\mathcal{A}, k)$, this one has a minimum cardinality.

To conclude, the complexity of the algorithm is addressed. A minimum coloring of $\mathcal{A}$ is computed in $O(n^{1.5})$ time when the arcs are ordered [41]. The complexity of the remainder of the algorithm is dominated by the complexity of the procedure Refine-Coloring. In the proof of Lemma 3.2, we have proved that this procedure stops after $\chi(\mathcal{A})$ loops in the worst case. The connected components of the bipartite graph induced by S_u and S_v is determined in $O(|S_u|+|S_v|)$ time and space by sweeping the arcs of $S_u \cup S_v$ in the circular order (the order on $S_u \cup S_v$ is obtained by merging the orders on S_u and S_v). Correctly implemented, the exchange of vertices between components is done in $O(|S_u| + |S_v|)$ time and space too. To summarize, the algorithm Refine-Coloring runs in $O(\chi(\mathcal{A})\, n)$ time and $O(n)$ space in the worst case.

Proposition 3.3 *The algorithm* MES-Proper-Circular-Arcs *computes an optimal solution to the MES problem in $O(n^2)$ time and $O(n)$ space, given an ordered set $\mathcal{A}$ of n proper arcs in input.*

Corollary 3.4 *Let G be a proper circular-arc graph. Then, the equality*

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

holds for all integers $k \geq 1$.

Since an ordered proper circular-arc representation is computed in linear time and space [13], we have the following corollary.

Corollary 3.5 *The MES problem is solved in $O(n^2)$ time and linear space for proper circular-arc graphs.*

3.2 A linear-time algorithm for the case $k = 2$

In this section, a linear-time and space algorithm is proposed to solve the 2-MES problem for proper circular-arc graphs. Similarly to the general case, the algorithm works on an ordered proper circular-arc representation $\mathcal{A} = \{A_1, \ldots, A_n\}$. Here are described the broad lines of the algorithm.

Having computed a maximum clique C in $\mathcal{A}$, two cases are considered. If the maximum clique is not too large (i.e., $\omega(\mathcal{A}) \leq \lfloor n/2 \rfloor$), then a maximum disjoint matching in $\mathcal{A}$ is greedily computed. Otherwise (i.e., $\omega(\mathcal{A}) > \lfloor n/2 \rfloor$), the arcs of the clique C are divided into two categories: the α-arcs and the β-arcs. Denote by c_i the arc having the smallest counterclockwise endpoint in C and c_j the arc having the largest counterclockwise endpoint in C which contains $cw(c_i)$. Some arcs may exist in C containing clockwise the endpoints $cw(c_j)$

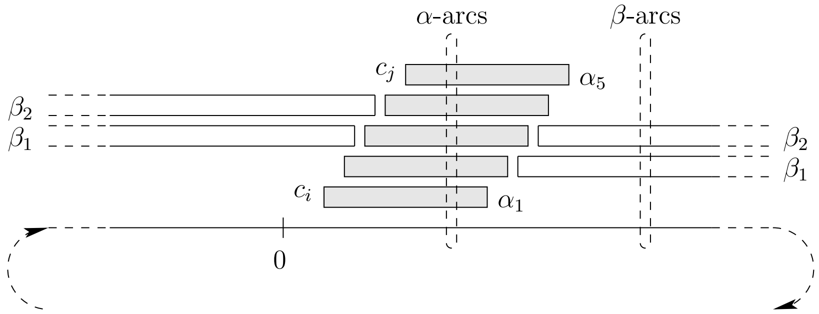

Fig. 6. The α-arcs and β-arcs of a clique.

and $ccw(c_i)$ (see Fig. 6). These arcs are called β-arcs, and on the opposite, all the other arcs of C are called α-arcs (including those which contain neither the endpoint $cw(c_i)$, nor the endpoint $cw(c_j)$). Note that the status of the arcs of C depends on the representation of the graph. The sets of α-arcs and β-arcs of C are respectively denoted by C_α and C_β ($|C| = |C_\alpha| + |C_\beta|$). The two sets $C_\alpha = \{\alpha_1, \ldots, \alpha_u\}$ and $C_\beta = \{\beta_1, \ldots, \beta_v\}$ are ordered as follows: the first arc of the set contains the counterclockwise endpoints of all the other arcs and the next arcs are arranged clockwise. Then, the set of arcs of $\mathcal{A} \setminus C$ which are candidates to match with arcs of C_α (resp. C_β) are denoted by $m(C_\alpha)$ (resp. $m(C_\beta)$). In the case $\omega(\mathcal{A}) > \lfloor n/2 \rfloor$, a maximum matching is obtained by performing a maximum disjoint matching between arcs of C_α and $m(C_\alpha)$, and then between arcs of C_β and $m(C_\beta)$. Next, we show how to determine efficiently these two matchings.

Algorithm 2-MES-Proper-Circular-Arcs;
Input: an ordered set $\mathcal{A} = \{A_1, \ldots, A_n\}$ of proper arcs;
Output: an optimal solution $\mathcal{M}$ to the 2-MES problem for $\mathcal{A}$;
Begin;
  compute a maximum clique C of $\mathcal{A}$;
  $\mathcal{M} \leftarrow \emptyset$;
  if $\omega(\mathcal{A}) \leq \lfloor n/2 \rfloor$ **then**
    for i from 1 to $\lfloor n/2 \rfloor$ **do** $\mathcal{M} \leftarrow \mathcal{M} \cup \{(A_i, A_{\lceil n/2 \rceil + i})\}$;
    if n is odd **then** add to $\mathcal{M}$ the arc $A_{\lceil n/2 \rceil}$ which remains unmatched;
  else
    compute the sets C_α and C_β;
    compute a maximum matching $\mathcal{M}_\beta$ between the arcs of C_β and $m(C_\beta)$;
    compute a maximum matching $\mathcal{M}_\alpha$ between the arcs of C_α and $m(C_\alpha)$;
    $\mathcal{M} \leftarrow \mathcal{M}_\alpha \cup \mathcal{M}_\beta$;
  return $\mathcal{M}$;
End;

The computation of a maximum clique takes $O(n)$ time and space when the arcs are proper and ordered [4,38]. Then, computing $\mathcal{M}$, which is easily done in linear time when $\omega(\mathcal{A}) \leq \lfloor n/2 \rfloor$, seems to be more complicated in the case $\omega(\mathcal{A}) > \lfloor n/2 \rfloor$. The following lemma shows that the matchings $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$

are easy to obtain too. Let $B_\alpha = (X, Y, E)$ be the bipartite graph with $X = C_\alpha$, $Y = m(C_\alpha)$ and $E = \{(\alpha_j, A_i) \,|\, \alpha_j \in C_\alpha, A_i \in m(C_\alpha) \text{ and } \alpha_j \cap A_i = \emptyset\}$. The bipartite graph B_β is defined in the same way with the sets C_β and $m(C_\beta)$.

Lemma 3.6 *The bipartite graphs B_α and B_β are convex. Moreover, the intervals defining the neighborhood of each vertex $x \in X$ in the linearly ordered set Y are proper.*

Proof. The assertion is only proved for the bipartite graph B_α, the proof is symmetric for B_β. The arcs of C_α and $m(C_\alpha)$ are ordered so that their endpoints appear clockwise. Now, consider the neighborhood of a vertex of X corresponding to the arc $\alpha_j \in C_\alpha$ $(1 \leq j \leq u)$. Denote by a_j the smallest index of an arc $m \in m(C_\alpha)$ such that $ccw(m) \notin \alpha_j$ and b_j the largest index of an arc $m \in m(C_\alpha)$ such that $cw(m) \notin \alpha_j$. Then, observe that the interval $[a_j, \ldots, |m(C_\alpha)|] \cap [1, \ldots, b_j]$ corresponds to the indices of arcs in $m(C_\alpha)$ which are matchable to α_j. If $a_j \leq b_j$, then this interval is not empty and corresponds exactly to $[a_j, \ldots, b_j]$, which establishes the convexity of B_α. To conclude, consider two arcs $\alpha_j, \alpha_{j'} \in C_\alpha$ with $j < j'$. Since the arcs are proper, we have $a_j \leq a_{j'}$ and $b_j \leq b_{j'}$, which implies that the intervals $[a_j, \ldots, b_j]$ defining the neighborhood of each vertex $\alpha_j \in C_\alpha$ in $m(C_\alpha)$ are proper. $\square$

According to this lemma, a maximum matching $\mathcal{M}_\alpha$ is determined in $O(n)$ time and space by sweeping the arcs of C_α and $m(C_\alpha)$, having arranged them clockwise (the maximum matching $\mathcal{M}_\beta$ can be determined similarly):

Algorithm COMPUTE-$\mathcal{M}_\alpha$;
Input: the ordered sets $C_\alpha = \{\alpha_1, \ldots, \alpha_u\}$ and $m(C_\alpha) = \{m_1, \ldots, m_{u'}\}$;
Output: the set $\mathcal{M}_\alpha$;
Begin;
  $\mathcal{M}_\alpha \leftarrow \emptyset$, $j \leftarrow 1$;
  for i from 1 to u **do**
    while $j \leq u'$ and $\alpha_i \cap m_j \neq \emptyset$ **do** $j \leftarrow j + 1$;
    if $j \leq u'$ **then** $\mathcal{M}_\alpha \leftarrow \mathcal{M}_\alpha \cup \{(\alpha_i, m_j)\}$;
    else $\mathcal{M}_\alpha \leftarrow \mathcal{M}_\alpha \cup \{(\alpha_i)\}$;
  return $\mathcal{M}_\alpha$;
End;

Now, the correctness of the entire algorithm is established. First, consider the case $\omega(\mathcal{A}) \leq \lfloor n/2 \rfloor$ and remind that the arcs of $\mathcal{A}$ are ordered.

Lemma 3.7 *If $\omega(\mathcal{A}) \leq \lfloor n/2 \rfloor$, then the set $\mathcal{M}$ forms an optimal solution to the 2-MES problem for $\mathcal{A}$.*

Proof. Suppose that the set $\mathcal{M}$ is such that two arcs A_i and $A_{\lceil n/2 \rceil + i}$ $(1 \leq$

$i \leq \lfloor n/2 \rfloor$) are intersecting. Two proper arcs can not overlap from both sides of the circle [24, pp. 191–192]. Thus, we have either $cw(A_i) \in A_{\lceil n/2 \rceil + i}$, or $ccw(A_i) \in A_{\lceil n/2 \rceil + i}$. When the arcs are proper, the order of counterclockwise endpoints is the same than the order of clockwise endpoints. In the first case, this implies the intersection of all the arcs between A_i and $A_{\lceil n/2 \rceil + i}$ in the circular order, and in the second case, the intersection of all the arcs $A_{\lceil n/2 \rceil + i}$ and A_i. In both cases, the existence of a clique of size $\lfloor n/2 \rfloor + 1$ is shown, which contradicts the initial hypothesis. Since $\lceil n/2 \rceil$ is a lower bound for $\chi(\mathcal{A}, 2)$, the set $\mathcal{M}$ has a minimum cardinality. $\square$

Then, consider the case $\omega(\mathcal{A}) > \lfloor n/2 \rfloor$. To make the proof clearer, we introduce two subsets of C_α, namely $C_{\alpha'}$ and $C_{\alpha''}$ (see Fig. 7). The set $C_{\alpha'}$ (resp. $C_{\alpha''}$) contains the arcs $\alpha_1, \ldots, \alpha_i$ (resp. $\alpha_j, \ldots, \alpha_u$) with i (resp. j) the largest (resp. smallest) index of an α-arc which does not contain the point $ccw(\beta_1)$ (resp. $cw(\beta_v)$). We can easily observe that the sets $C_{\alpha'}$ and $C_{\alpha''}$ are disjoint ($i < j$). As previously, the set $m(C_{\alpha'})$ (resp. $m(C_{\alpha''})$) is defined as the arcs which are candidates to match with the arcs of $C_{\alpha'}$ (resp. $C_{\alpha''}$).

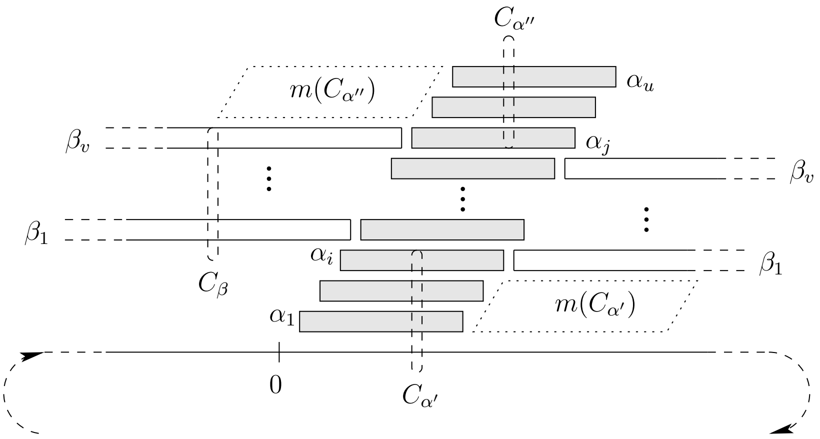

Fig. 7. The clique sets $C_{\alpha'}$, $C_{\alpha''}$, C_β.

Lemma 3.8 *The three following sets induce each one a clique:* $m(C_\beta) \cup C_\alpha$, $m(C_{\alpha'}) \cup C_{\alpha''} \cup C_\beta$, $m(C_{\alpha''}) \cup C_{\alpha'} \cup C_\beta$. *Moreover,* $m(C_\alpha) = m(C_{\alpha'}) \cup m(C_{\alpha''})$.

Proof. First, we show that the set $C_\alpha \cup m(C_\beta)$ induces a clique. According to the definition of C_β, all arc of $m(C_\beta)$ must be included in the portion $]ccw(\alpha_1), cw(\alpha_u)[$ of the circle. Since the arcs are proper, any arc of $m(C_\beta)$ contains the portion $]ccw(\alpha_u), cw(\alpha_1)[$ of the circle and $C_\alpha \cup m(C_\beta)$ induces well a clique.

Then, we demonstrate that any arc of $m(C_{\alpha'})$ (resp. $m(C_{\alpha''})$) induces a clique with the arcs of $C_{\alpha''} \cup C_\beta$ (resp. $C_{\alpha'} \cup C_\beta$). Any arc $m(C_{\alpha'})$ (resp. $m(C_{\alpha''})$) can not be included in the portion $]ccw(\beta_1), cw(\beta_v)[$ of the circle (otherwise this one is strictly contained in an arc of C_β). Such an arc can not have its two endpoints included in the portion $]ccw(\alpha_1), cw(\alpha_u)[$ of the circle too. Consequently, any arc of $m(C_{\alpha'})$ (resp. $m(C_{\alpha''})$) contains clockwise (resp. counterclockwise) the points $ccw(\beta_1)$, $cw(\alpha_j)$, $ccw(\beta_v)$, $cw(\alpha_u)$ (resp. the points $cw(\beta_v)$, $ccw(\alpha_i)$, $cw(\beta_1)$, $ccw(\alpha_1)$), which allows to conclude.

Following the previous discussion, any arc which does not intersect an arc among $\alpha_{i+1}, \ldots, \alpha_{j-1}$ is necessarily contained either in β_1 or in β_v, which is a contradiction. Thus, no arc can be matched to arcs $\alpha_{i+1}, \ldots, \alpha_{j-1}$ of C_α, implying that $m(C_\alpha) = m(C_{\alpha'}) \cup m(C_{\alpha''})$. $\square$

Now, we are ready to establish that the set $\mathcal{M} = \mathcal{M}_\alpha \cup \mathcal{M}_\beta$ forms well a maximum disjoint matching in $\mathcal{A}$.

Lemma 3.9 *The sets $m(C_{\alpha'})$, $m(C_{\alpha''})$ and $m(C_\beta)$ are disjoint. Moreover, all arcs in $m(C_{\alpha'})$, $m(C_{\alpha''})$ and $m(C_\beta)$ are matched.*

Proof. The first assertion follows immediately from the previous lemma. Now, we establish that all arcs in $m(C_{\alpha'})$ are matched; the proof is similar for sets $m(C_{\alpha''})$ and $m(C_\beta)$. According to the celebrated Hall's marriage theorem [14, pp. 35–37], all arcs in $m(C_{\alpha'})$ are matched if and only if for any subset $S \subseteq m(C_{\alpha'})$, $|S| \leq |N(S)|$ with $N(S)$ the neighborhood of S. Thus, we show that the Hall condition holds for the bipartite graph induced by $C_{\alpha'}$ and $m(C_{\alpha'})$. Assume on the contrary that a set $S \subseteq m(C_{\alpha'})$ exists such that $|S| > |N(S)|$. Since all arcs in S intersect all arcs in $C_{\alpha'} \setminus N(S)$ (according to the definition of $N(S)$) and $S \cup C_{\alpha''} \cup C_\beta$ forms a clique (according to Lemma 3.8), we observe that the set $S \cup (C_{\alpha'} \setminus N(S)) \cup C_{\alpha''} \cup C_\beta$ induces a clique of size strictly greater than $\omega(A)$, which is a contradiction. $\square$

Remark 3.10 *Following the previous discussions, we can observe that for any proper circular-arc graph, a maximum clique $C = C_\alpha \cup C_\beta$ exists such that $m(C_\beta) = \emptyset$. Such a maximum clique, which can be viewed as standard, is simply obtained by replacing each arc $\beta_j \in C_\beta$ by the arc $m \in m(C_\beta)$ to which β_j is matched (if one exists).*

Proposition 3.11 *The algorithm* 2-MES-PROPER-CIRCULAR-ARCS *computes in $O(n)$ time and space an optimal solution to the 2-MES problem, given an ordered set $\mathcal{A}$ of n proper arcs in input.*

Corollary 3.12 *Let G be a proper circular-arc graph. If $\omega(G) \geq \lceil n/2 \rceil$, then $\omega(G) = \chi(G)$. Hence, the equality $\chi(G, 2) = \max\{\omega(G), \lceil n/2 \rceil\}$ holds.*

Since an ordered proper circular-arc representation is computed in linear time and space from a proper circular-arc graph [13], we also obtain the following corollary.

Corollary 3.13 *The 2-MES problem is solved in linear time and space for proper circular-arc graphs.*

4 Mutual exclusion scheduling with tolerance graphs

Bodlaender and Jansen [7] have established the $\mathcal{NP}$-hardness of MES for interval graphs by performing a reduction from the problem NUMERICAL 3-D MATCHING [16]. This kind of reduction was previously employed by Jansen [30] to show the hardness of a scheduling problem restricted to interval orders. In our turn, we are inspired from this technique to prove that 3-MES remains $\mathcal{NP}$-hard for a subclass of tolerance graphs closed to the class of interval graphs. The proposition extends the result of Lonc [37] who established that 3-MES is $\mathcal{NP}$-hard for complements of comparability graphs.

Proposition 4.1 *The 3-MES problem remains $\mathcal{NP}$-hard for bounded tolerance graphs, even if every cycle of length greater than or equal to five has two chords.*

Proof. An instance of NUMERICAL 3-D MATCHING is given by three disjoint sets $W = \{w_1, \ldots, w_m\}$, $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_m\}$ containing each one m elements, the size $s(a) \in \mathbb{N}$ of each element $a \in W \cup X \cup Y$, and a bound Z such that $\sum_{a \in W \cup X \cup Y} s(a) = mZ$. The question is to decide if $W \cup X \cup Y$ admits a partition into m disjoint sets $\{A_i\}_{i=1,\ldots,m}$ such that each one contains exactly one element from each set W, X and Y and $\sum_{a \in A_i} s(a) = Z$ for $1 \leq i \leq m$. The problem remains $\mathcal{NP}$-hard if $0 < s(a) < Z/2$ for all $a \in W \cup X \cup Y$ and $1 < m < Z$. This is proved by transforming the original problem into another one where the assertion is verified. For this, add the value $Z + m$ to each $a \in W \cup X \cup Y$ and setting $Z' = Z + 3(Z + m)$.

Given an instance NUMERICAL 3-D MATCHING, a graph is built corresponding to an instance of the 3-MES problem. This graph is represented by a set of intervals, having each one a tolerance. All the intervals are open and their endpoints are integer; all the tolerances are bounded and integer. Thus, the subjacent graph is a bounded tolerance graph. Later, we shall show that any cycle of length greater than or equal to five in this graph always has two chords. Here is the set of intervals in question:

(1) for each $1 \leq i \leq m$, set m intervals $a_{i,l} =]0, i+1[$ with tolerance $t(a_{i,l}) = 0$;

(2) for all pairs $w_i \in W$, $x_j \in X$, set one interval $b_{i,j} =]i+1, w_i + x_j + jZ + 1[$ with tolerance $t(b_{i,j}) = 0$;
(3) for all pairs $x_j \in X$, $y_k \in Y$, set one interval $c_{j,k} =](j+1)Z - y_k + 1, (m+3)Z + k + 1[$ with tolerance $t(c_{j,k}) = 2k + 1$;
(4) for each $1 \leq k \leq m$, set $(m-1)$ intervals $d_{k,l} =](m+3)Z - k, (m+5)Z + 1[$ with tolerance $t(d_{k,l}) = 2k + 1$;
(5) for each $1 \leq j \leq m$, set $(m-1)$ intervals $h_{j,l} =]0, jZ + 1[$ and $(m-1)$ intervals $g_{j,l} =](j+1)Z, (m+3)Z + 1[$ with tolerance $t(g_{j,l}) = t(h_{j,l}) = 0$.

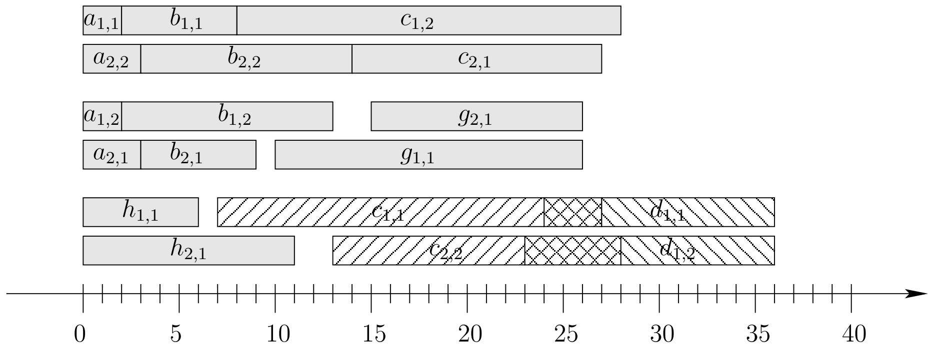

Fig. 8. An example of construction.

An example of construction is given on Fig. 8 with $w_1 = 1$, $w_2 = 2$, $x_1 = 1$, $x_2 = 1$, $y_1 = 2$, $y_2 = 3$ and $Z = 5$, $m = 2$. The sets of intervals $a_{i,l}$ is denoted by the letter A; in the same way, the sets B, C, D, G, H are defined. Some of these sets induce a clique independently of the instance considered: $A \cup H$, $B \cup H$, $C \cup G$, $D \cup G$. The intervals of sets A, B, G, H tolerate no intersection and all the intervals of C (resp. D) overlap the portion $[(m+1)Z + 1, (m+3)Z + 1)]$ (resp. $[(m+3)Z, (m+5)Z]$) of the line, whose length is greater than the maximum of their tolerances ($2Z > 2m + 1$). Note that any stable is of size at most four in this graph. The cardinality of the different sets is given by $|A| = |B| = |C| = m^2$ and $|D| = |G| = |H| = m(m-1)$. On the whole, there are $6m^2 - 3m$ intervals. Thus, the problem consists in finding a partition of the graph into $2m^2 - m$ stables of size at most three, which turns to determine an optimal solution to the 3-MES problem for this graph. In fact, each stable of the partition must be of size exactly three.

Let us consider a stable U of size three which contains an interval $h \in H$. The only way to complete this stable U is to choose an interval $c \in C$ and an interval $d \in D$. In the same way, for a stable U which contains $g \in G$, we can only take an interval $b \in B$ and an interval $a \in A$. Having removed these intervals, only m elements remain in A, B and C. Now, the composition of the stables which belong to an optimal solution is detailed. For this, we analyse the possible (disjoint) matchings between intervals coming from the following sets: A and B, $B \cup H$ and $C \cup G$, C and D.

Case (a): the sets A and B. These two sets contain each one m^2 intervals. According to the previous discussion, each interval from A must be matched to an interval from B in an optimal partition into stables of size three. We show that the vertices $a_{i,l} \in A$ and $b_{i',j} \in B$ belong to the same stable if and only if $i = i'$. In other words, an optimal partition can not contain some stable U with $\{a_{i,l}, b_{i',j}\} \subset U$ if $i \neq i'$. Let us suppose the contrary. We have m^2 stables, each one containing an element from A and an element from B. Let $a_{i,l} \in A$ be the interval of smallest index i which belongs to a stable containing an interval $b_{i',j}$ with $i \neq i'$. If $i > i'$, then the intervals in question are intersecting by construction, which leads to a contradiction. Now, let us see the case $i < i'$. According to the hypothesis, the intervals of A having an index lower than i are correctly matched to an interval of B. Among the m intervals of B having as first index i, at least one exists which is matched to an interval $a_{i'',l'}$ with $i'' > i$. Since these intervals are intersecting by construction, we still obtain a contradiction.

Case (b): the sets $B \cup H$ and $C \cup G$. These two sets contain each one $2m^2 - m$ intervals and induce each one a clique. Consequently, any stable of an optimal partition must include one element of $B \cup H$ and one element of $C \cup G$. Having observed that the interval $b_{i,j}$ overlaps the interval $g_{j',l}$ if $j' < j$ and the interval $c_{j,k}$ overlaps the interval $h_{j',l}$ if $j' > j$, we fall into the same situation as in the case (a) and prove in a similar way that any pair of intervals $\{b_{i,j}, c_{j',k}\}$, $\{b_{i,j}, g_{j',l}\}$ or $\{h_{j,l}, c_{j',k}\}$ belongs to a same stable of an optimal partition if and only if $j = j'$.

Case (c): the sets C and D. For all pairs $c_{j,k}$ and $d_{k',l}$ of intervals, we have $|c_{j,k} \cap d_{k',l}| = k + k' + 1$. Since the tolerances of these two intervals are respectively equal to $2k+1$ and $2k'+1$, these ones are matchable only if $k+k'+1 \leq 2k+1$ and $k + k' + 1 \leq 2k' + 1$, that is, only if $k = k'$.

first interval	second interval
$a_{i,-}$	$b_{i,-}$
$b_{-,j}$ or $h_{j,-}$	$c_{j,-}$ or $g_{j,-}$
$c_{-,k}$	$d_{k,-}$

Fig. 9. Summary of the analysis of cases (a), (b) and (c).

The results of the analysis of cases (a), (b) and (c) are summarized on Fig. 9. Each row of the table represents the pairs of intervals contained by the stables of an optimal partition for 3-MES. For example, the first line means that no stable exists which contains some intervals $\{a_i, b_{i',l}\}$ with $i \neq i'$. Then, remove from an optimal solution all the stables U containing $h \in H$ or $g \in G$. Since each interval $g_{j,-}$ (resp. $h_{j,-}$) is matched to an interval $b_{-,j}$ (resp. $c_{j,-}$), one and only one interval $b_{-,j}$ (resp. $c_{j,-}$) remains in B (resp. C) for all $j = 1, \ldots, m$.

In the same way, the stables which contain the intervals of G and of B (resp. of H and of C) are necessarily completed by some intervals of A (resp. D). Consequently, one interval $a_{i,l}$ remains for all $i = 1, \dots, m$ too, but no interval of D (because this one has the same cardinality than H). Finally, only m stables $U_i = \{a_i, b_{i,j}, c_{j,k}\}$ remain after suppression of all the stables U. We are now able to prove that there exists a partition of $W \cup X \cup Y$ into m sets $\{A_i\}_{i=1,\dots,m}$, each one containing an element of W, X and Y such that $\sum_{a \in A_i} s(a) = Z$, if and only if the tolerance graph admits a partition into $2m^2 - m$ stables of size three.

Let $U_1, \dots, U_{2m^2-m}$ be such a partition. According to the previous discussion, we can admit without loss of generality that the first m stables of the partition have the form $U_i = \{a_{i,l}, b_{i,j}, c_{j,k}\}$ in such a way that each index i, j or k appears one and only one time ($1 \leq i, j, k \leq m$). Since U_i is a stable, we have $w_i + x_j + jZ + 1 \leq (j+1)Z - y_k + 1$, and then $w_i + x_j + y_k \leq Z$. Because each index appears exactly one time, we obtain that $\sum_{i=1}^{m} w_i + \sum_{j=1}^{m} x_j + \sum_{k=1}^{m} y_k = mZ$. Thus, $w_i + x_j + y_k = Z$ and the sets $A_i = \{w_i, x_j, y_k\}$ defined from stables U_i form a solution to the problem Numerical 3-D Matching.

Let us prove the reverse implication. Let $A_i = \{w_i, x_j, y_k\}$ be the m sets such that $\sum_{a \in A_i} s(a) = Z$ ($1 \leq i \leq m$). Given these m sets, we shall construct an optimal partition of the tolerance graph into stables of size at most three. First, for all $i = 1, \dots, m$, define one stable $U_i = \{a_{i,l}, b_{i,j}, c_{j,k}\}$. Clearly, the intervals a_i and $b_{i,j}$ are not intersecting, as well as the intervals $b_{i,j}$ and $c_{j,k}$ (the equality $w_i + x_j + y_k = Z$ implies that $w_i + x_j + jZ + 1 \leq (j+1)Z - y_k + 1$). Thus, each set U_i induces well a stable. Now, denote by B' the set of vertices of B which is not covered by the stables U_i. For each $b_{i,j} \in B'$, define a set which contains the intervals $a_{i,l}, b_{i,j}$ and $g_{j,l'}$. Clearly, such a set induces a stable. Moreover, the construction is correct since each index i or j appears only $(m-1)$ times in B'. Finally, denote by C' the set of vertices of C which remain uncovered and define a set which contains the vertices $h_{j,l}, c_{j,k}, d_{k,l'}$ for each $c_{j,k} \in C'$. The tolerances of intervals $c_{j,k}$ and $d_{k,l'}$ are such that the corresponding vertices in the tolerance graph are not connected. On the other hand, intervals $d_{k,l'}$ can not overlap intervals of $h_{j,l}$. Consequently, such sets induce stables too. Finally, all the vertices are partitioned into $2m^2 - m$ stables.

The $\mathcal{NP}$-hardness of the 3-MES problem is established for bounded tolerance graphs. To close definitively the proof, we show that every cycle of length greater than or equal to five in the graph has two chords. The only vertices suitable to violate the condition in question are those having a tolerance strictly greater than zero, that is, those of $C \cup D$. In effect, the subgraph induced by all the intervals excepted those of C (resp. D) is clearly an interval graph and then contains no chordless cycle of length greater than or equal to four. Now, assume the existence of a cycle induced by vertices of $C \cup D$ which does not verify the condition. A cycle of length greater than or equal to five always contains at least three vertices either from C, or from D. If it contains four or more, then these ones induce at least two chords (because the sets C and D are some cliques). The situation is the same if the cycle contains three vertices of C (resp. D) not appearing consecutively on the cycle. Consequently, one case remains to tackle: the cycle contains exactly three vertices of C (resp. D) such that these ones appear consecutively on the cycle. Without loss of generality, assume that these three vertices belong to the set C and denote by $\{c_{j_1,k_1}, c_{j_2,k_2}, c_{j_3,k_3}, d_{k_4,l_4}, d_{k_5,l_5}, c_{j_1,k_1}\}$ the cycle in question. Since the set C is a clique, the vertices c_{j_1,k_1} and c_{j_3,k_3} are connected by a chord. According to the analysis of case (c), the vertices $c_{j,k} \in C$ and $d_{k',l} \in D$ of the tolerance graph are not connected if and only if $k = k'$. Since the vertex c_{j_1,k_1} is connected to d_{k_5,l_5} but not to d_{k_4,l_4}, we obtain that $k_1 \neq k_5$ and $k_1 = k_4$ (symmetrically, we have with the vertex c_{j_3,k_3} that $k_3 = k_5$ and $k_3 \neq k_4$). Hence, we deduce that $k_4 \neq k_5$, and that the vertex c_{j_2,k_2} is necessarily connected to one of the two vertices d_{k_4,l_4} or d_{k_5,l_5}, which completes the proof. $\square$

The graphs aimed by Proposition 4.1 satisfy the following conditions: every cycle of length greater than or equal to five has two chords and the complement graph is transitively orientable. Indeed, complements of bounded tolerance graphs are comparability graphs (see [8,25]). On the other hand, interval graphs are exactly the graphs where every cycle of length greater than or equal to four owns one chord and the complement graph is transitively orientable. The graphs aimed by the proposition differ from these ones because they induce, under certain conditions, chordless cycles of length four.

Bounded tolerance graphs (also known as parallelogram graphs) form a subclass of tolerance graphs, trapezoid graphs and weakly triangulated graphs. On the other hand, Meyniel graphs are the graphs satisfying the property that every odd cycle of length greater than or equal to five has at least two chords. (The interested reader is referred to [8] for more details concerning these classes of graphs.) Consequently, we have the following corollary.

Corollary 4.2 *For each fixed $k \geq 3$, the k-MES problem is $\mathcal{NP}$-hard for tolerance graphs, trapezoid graphs, weakly triangulated graphs and Meyniel graphs, even if the complement graph is transitively orientable.*

Remark 4.3 *Whereas the k-MES problem is solvable in polynomial time for perfect graphs of stability at most k, the proof of Proposition 4.1 provides that the problem becomes $\mathcal{NP}$-hard for bounded tolerance graphs of stability at least $k+1$.*

5 Conclusion

The following tables summarize all the results presented throughout the paper about the complexity of the mutual exclusion scheduling problem for interval graphs, circular-arc graphs and tolerance graphs.

	Proper interval graphs	Threshold graphs	Interval graphs
$k = 2$	$O(n+m)$	$O(n+m)$	$O(n+m)$
$k = 3$	$O(n+m)$	$O(n+m)$	*open*
$k \geq 4$	$O(n+m)$	$O(n+m)$	$\mathcal{NP}$-hard [7]

	Proper circular-arc graphs	Circular-arc graphs
$k = 2$	$O(n+m)$	maximum matching [23]
$k = 3$	$O(n^2)$	*open*
$k \geq 4$	$O(n^2)$	$\mathcal{NP}$-hard (even if perfect) [7]

	Proper tolerance graphs	Tolerance graphs
$k = 2$	maximum matching [23]	maximum matching [23]
$k \geq 3$	*open*	$\mathcal{NP}$-hard (even if bounded)

A cartography of the complexity of the problem is given on Fig. 10. Although the problem remains $\mathcal{NP}$-hard for many classes of (perfect) graphs, some interesting questions remain open concerning the complexity of MES for interval graphs and permutation graphs when k is a small fixed parameter. Another topic is the complexity of MES for proper tolerance graphs, as well as for complements of circular-arc graphs or complements of tolerance graphs. Finding practical efficient algorithms (in particular linear-time algorithms) to solve the 2-MES problem for circular-arc graphs or tolerance graphs is also of interest.

Acknowledgements

This paper was written during Christmas 2005 while I stayed in Goult, my native village in Provence, with my grandmother Odette Gardi; the paper is dedicated to her memory. We also express our gratitude to the two anonymous referees for their meticulous reviewing, which has resulted in an improved paper.

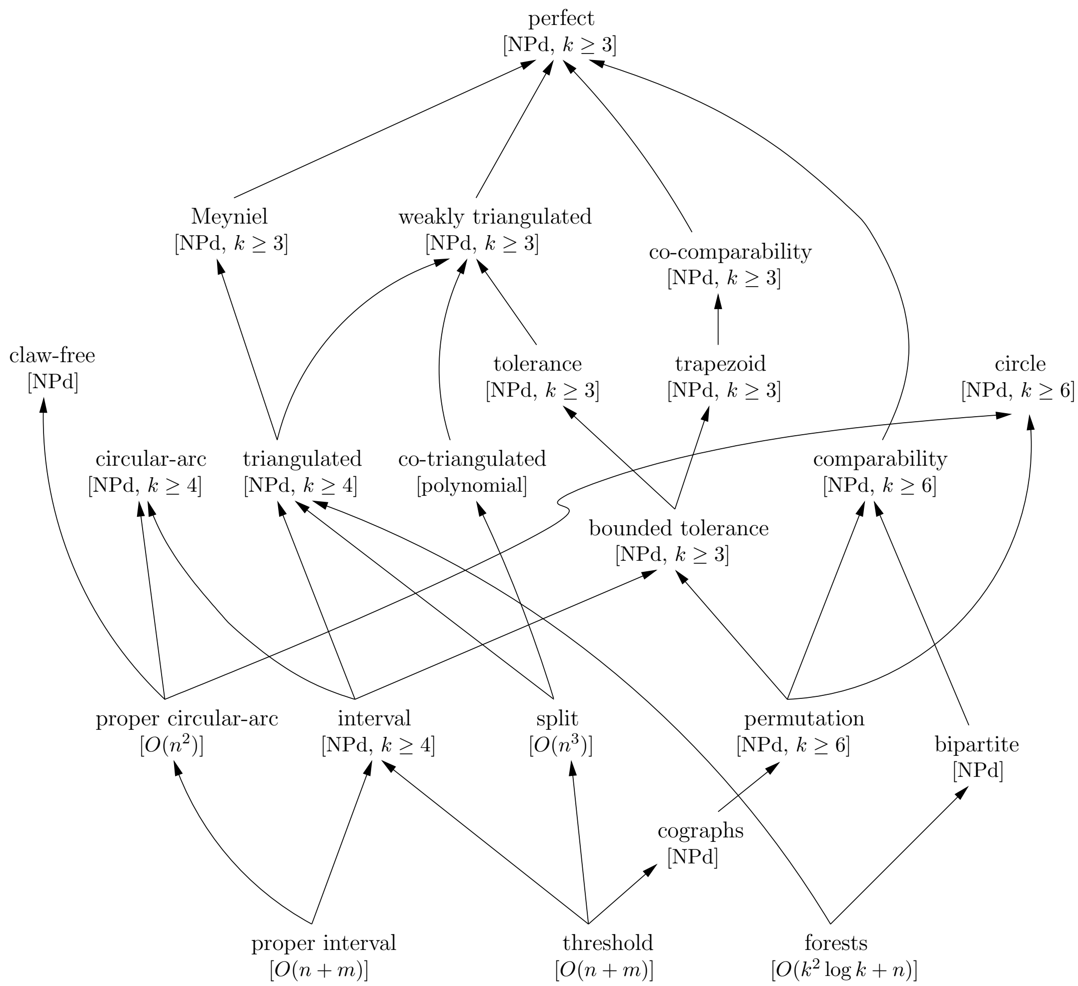

Fig. 10. A cartography of the complexity of the MES problem.

References

[1] M.G. Andrews, M.J. Atallah, D.Z. Chen and D.T. Lee (2000). Parallel algorithms for maximum matching in complements of interval graphs and related problems. *Algorithmica* 26, pp. 263–289.

[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/solutions/bamboo/`

[4] B.K. Bhattacharya, P. Hell and J. Huang (1996). A linear algorithm for maximum weight cliques in proper circular arc graphs. *SIAM Journal on Discrete Mathematics* 9(2), pp. 274–289.

[5] J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt and J. Weglarz (2001). *Scheduling Computer and Manufacturing Processes.* Springer-Verlag, Berlin,

Germany. (2nd edition)

[6] 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.

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

[8] 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.

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

[10] T. Cormen, C. Leiserson and R. Rivest (1994). *Introduction à l'Algorithmique.* Dunod, Paris, France. (French edition)

[11] 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.

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

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

[14] R. Diestel (2005). *Graph Theory.* Graduate Texts in Mathematics 173, Springer-Verlag, New York, NY. (3rd edition)

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

[16] M.R. Garey and D.S. Johnson (1979). *Computer and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness.* W.H. Freeman & Co, San Francisco, CA.

[17] M.R. Garey, D.S. Johnson, G.L. Miller and C.H. Papadimitriou (1980). The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic and Discrete Methods* 1(2), pp. 216–227.

[18] F. Gardi (2003). Efficient algorithms for disjoint matchings among intervals and related problems. In *Proceedings of the 4th International Conference on Discrete Mathematics and Theoretical Computer Science* (C.S. Calude, M.J. Dinneen and V. Vajnovszki, eds), *LNCS* 2731, pp. 168–180. Springer-Verlag, Berlin, Germany.

[19] F. Gardi (2003). Planification d'horaires de travail et colorations de graphes. In *Actes des Journées Graphes, Réseaux et Modélisation.* Laboratoire d'Infomatique Algorithmique, Fondements et Applications (LIAFA, Université Denis Diderot – Paris VII), Paris, France.

[20] 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`

[21] F. Gardi (2005). 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)

[22] F. Glover (1967). Maximum matchings in a convex bipartite graph. *Naval Research Logistics Quarterly* 14(3), pp. 313–316.

[23] A.V. Golberg and A.V. Karzanov (1995). Maximum skew-symmetric flows. In *Proceedings of the Third Annual European Symposium on Algorithms* (P.G. Spirakis, ed), *LNCS* 2385, pp. 155–170. Springer-Verlag, Berlin, Germany.

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

[25] M.C. Golumbic and A. Siani (2002). Coloring algorithms for tolerance graphs: reasoning and scheduling with interval constraints. In *Proceedings of Artificial Intelligence, Automated Reasoning, and Symbolic Computation, Joint International Conferences* (J. Calmet, B. Benhamou, O. Caprotti, L. Henocque and V. Sorge, eds), *LNCS* 2385, pp. 196–207. Springer-Verlag, Berlin, Germany.

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

[27] U.I. Gupta, D.T. Lee and J.Y.-T. Leung (1979). An optimal solution for the channel-assignment problem. *IEEE Transactions on Computers* C-28(11), pp. 459–467.

[28] U.I. Gupta, D.T. Lee and J.Y.-T. Leung (1982). Efficient algorithms for interval graphs and circular-arc graphs. *Networks* 12, pp. 459–467.

[29] 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.

[30] K. Jansen (1993). Scheduling with constrained processor allocation for interval orders. *Computers and Operations Research* 20(6), pp. 587–595.

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

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

[33] V. Jost (2006). Ordonnancement chromatique : polyhèdres, complexité et classification. PhD Thesis. Université Joseph Fourier – Grenoble I, Grenoble, France. `http://roso.epfl.ch/vj/These_VJOST.pdf`

[34] R.M. Karp (1972). Reducibilty among combinatorial problems. In *Complexity of Computer Computations* (R.E. Miller and J.W. Thatcher, eds), pp. 85–103. Plenum Press, New York, NY.

[35] J. Krarup and D. De Werra (1982). Chromatic optimization: limitations, objectives, uses, references. *European Journal of Operational Research* 11, pp. 1–19.

[36] Y.D. Liang and N. Blum (1995). Circular convex bipartite graphs: maximum matching and hamiltonian circuits. *Information Processing Letters* 56, pp. 215–219.

[37] 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.

[38] G.K. Manacher and T.A. Mankus (1997). Finding a maximum clique in a set of proper circular arcs in time $O(n)$ with applications. *International Journal of Foundations of Computer Science* 8(4), pp. 443–467.

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

[40] 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.

[41] W.-K. Shih and W.-L. Hsu (1985). An $O(n^{1.5})$ algorithm to color proper circular-arc graphs. *Discrete Applied Mathematics* 25(3), pp. 321–323.

[42] G. Steiner and J.S. Yeomans (1996). A linear time algorithm for maximum matchings in convex, bipartite graphs. *Computers and Mathematics with Applications* 31(12), pp. 91–96.

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

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