LIF

Laboratoire d'Informatique Fondamentale
de Marseille

Unité Mixte de Recherche 6166
CNRS – Université de Provence – Université de la Méditerranée

The mutual exclusion scheduling problem for proper interval graphs

Frédéric Gardi

Rapport/Report 02-2002

Avril 2002

The mutual exclusion scheduling problem for proper interval graphs

Frédéric Gardi

LIF – Laboratoire d'Informatique Fondamentale de Marseille
UMR 6166
CNRS – Université de Provence – Université de la Méditerranée

Parc Scientifique et Technologique de Luminy,
163, avenue de Luminy - Case 901,
F-13288 Marseille Cedex 9, France.

gardi@lidil.univ-mrs.fr

Abstract/Résumé

In this paper, the mutual exclusion scheduling (MES) problem for proper interval graphs is considered. Given an undirected graph G and an integer k, the problem is to find a minimum coloring of G, such that each colour is used at most k times. The complexity of this problem is known to be $\mathcal{NP}$-complete for interval graphs. We prove that MES can be solved in linear time for proper interval graphs thanks to a greedy algorithm. As a byproduct of the proof of this result, we also obtain a linear-time algorithm to solve MES for interval graphs under certain conditions. Finally, this study yields some results about the complexity of finding an uniform coloring of proper interval graphs. **Keywords:** mutual exclusion scheduling, working schedules planning, proper interval graphs, colorings, linear-time algorithms.

Dans ce papier, un problème d'ordonnancement avec exclusion mutuelle (appelé MES) est étudié pour les graphes d'intervalles propres. Etant donné un graphe G non-orienté et un entier k, le problème est de trouver une coloration minimum de G, tel que chaque couleur ne soit pas utilisée plus de k fois. Ce problème est $\mathcal{NP}$-complet pour les graphes d'intervalles. Nous montrons que le MES peut être résolu en temps linéaire pour les graphes d'intervalles propres par un algorithme glouton. La preuve de ce résultat nous fournit aussi un algorithme linéaire en temps pour résoudre le MES pour les graphes d'intervalles sous certaines conditions. Enfin, au travers de cette étude nous obtenons des résultats annexes portant sur la complexité du problème de la coloration uniforme des graphes d'intervalles propres. **Mots-clés:** ordonnancement avec exclusion mutuelle, planification d'horaires de travail, graphes d'intervalles propres, colorations, algorithmes linéaires en temps.

Relecteurs/Reviewers: Michel Van Canegehem, Victor Chepoi.

1 Introduction

The following problem arises in scheduling theory: n unit-time jobs must be complete on k processors in a minimum time with the constraint that some jobs cannot be executed at the same time because they share a same ressource. Many variants of this problem have been studied in Combinatorial Optimization and Operations Research litterature (see [3, 8, 12] for some practical applications). Therefore, such scheduling problems can be alternatively formulated in graph theoretic terms. Indeed, by creating an undirected graph $G = (V, E)$ with a vertex for each of the n jobs and an edge between each pair of conflicting jobs, we can see that a minimum length of schedule corresponds to a partition of V into a minimum number of independent sets of size at most k. In this way, B.S. Baker and E.G. Coffman called *Mutual Exclusion Scheduling* (shortly MES) the following graph theoretical problem: given an undirected graph G and $k \in \mathbb{N}$, find a minimum coloring of G, such that each colour is used at most k times [2]. In this paper we will analyse the complexity of the MES problem in the case where G is a proper interval graph.
A graph $G = (V, E)$ is an *interval graph* iff to each vertex $v \in V$ a closed (resp. open) interval I_v in the real line can be associated, such that for each pair of vertices $u, v \in V$, $u \neq v$, $uv \in E$ if and only if $I_u \cap I_v \neq \emptyset$. An *interval representation* of G will be noted $\{I_v\}_{v \in V}$, with $left(I_v)$, $right(I_v) \in \mathbb{R}$ the left and right endpoints of I_v. The complement $\overline{G}$ of an interval graph can be transitively oriented with $(u, v) \in F$ iff $right(I_u) < left(I_v)$. This orientation F of the edges induces a partial order $P = (V, F)$ which is called an *interval order*. We will write $I_u \prec I_v$ iff $right(I_u) < left(I_v)$. Now, G is called a *proper interval graph* iff there is an interval representation for G in which no interval contains properly (strictly) another. In the same way, G is called an *unit interval graph* iff there is an interval representation for G in which each interval has a unit size. Interval graphs arise in many practical applications because they modelize many structures of the real-life world. They appear notably in areas like genetics, psychology, sociology, archæology, scheduling and others. The interested reader can consult [6] and [15] for surveys.

Our interest to the particular case of MES for (proper) interval graphs comes from the following *working schedules planning* (WSP) problem, which has actually inspired this research.

WSP problem:

Let $T_1, \ldots, T_n$ be n tasks such that $T_i = [l_i, r_i]$ where $l_i, r_i \in \mathbb{N}$ are the starting and ending dates of T_i. Let $m \in \mathbb{N}$ be the number of employees available and qualified to execute these tasks. Given that the tasks allocated to an employee must not overlap and the reglementation imposes no more than $k \in \mathbb{N}$ tasks by employees, are there enough employees to execute all the tasks ?

Clearly, we can see that the WSP problem is equivalent to the decision version of MES for interval graphs.
The MES problem is $\mathcal{NP}$-complete for general graphs since that the coloring problem is $\mathcal{NP}$-complete [10]. Thus, MES was studied for classes of graphs for which the coloring problem is polynomial, notably for some classes of perfect

graphs (see [6] for an introduction to the world of perfect graphs). Therefore, the MES problem remains $\mathcal{NP}$-complete for many classes of perfect graphs: permutation graphs [9], complement of comparability graphs [13], bipartite graphs, cographs and interval graphs [4]. At our knowledge, MES is polynomially solvable only for split graphs [13] and complement of interval graphs [4].
The MES problem (and so the WSP problem) are also $\mathcal{NP}$-complete for interval graphs if k is a constant such that $k \geq 4$ [4] (its complexity for a constant $k = 3$ is still an open question at our acquaintance). Our motivation for this work was to study the complexity of MES for the basic subclass of interval graphs: the class of proper interval graphs. Here we establish that a greedy algorithm can solve it in linear time. As a byproduct of the proof of this result, we also obtain a linear-time algorithm to solve the MES problem for interval graphs under certain conditions (which will be given in Lemma 2). Finally, this study yields some results about the complexity of finding an uniform coloring of proper interval graphs.

2 Preliminaries

First, we recall some definitions from graph theory (see [6] for more details). Let $G = (V, E)$ be an undirected graph. Given a subset $A \subseteq V$ of vertices, we define the *subgraph induced by* A to be $G_A = (A, E_A)$, where $E_A = \{xy \in E \mid x \in A$ and $y \in A\}$. A *clique* of G is a set $C \subseteq V$ such that $xy \in E$ for all $x, y \in C$. An *independent set* or a *stable* of G is a set $S \subseteq V$ such that $xy \notin E$ for all $x, y \in S$. A *q-coloring* or a *partition of size q into stables of* G is a partition $\mathcal{S} = \{S_1, \ldots, S_q\}$ of V such that each S_i is a stable. The *chromatic number* $\chi(G)$ of G is the size of a partition of G into the least number of stables. We will denote by $\chi(G, k)$ the size of a minimum partition of G into stables of size at most $k \in \mathbb{N}$. Immediatly, we can give a trivial lower bound for $\chi(G, k)$.

Proposition 1 *Let G be an undirected graph with n vertices and $k \in \mathbb{N}$,*

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

Finally, a q-coloring is called *uniform* if all the stables induced by the coloring have size $\lfloor \frac{n}{q} \rfloor$ or $\lceil \frac{n}{q} \rceil$. A graph is called *$K_{1,3}$-free* iff it does not contain $K_{1,3}$ as an induced subgraph (see Fig. 1).

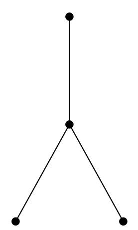


Fig. 1. The graph $K_{1,3}$.

F.S. Roberts has proved that "proper = unit = $K_{1,3}$-free" for interval graphs (see [14] or [6] for details and proofs).

Proposition 2 (Roberts, 1969) *Let G be an undirected graph. The following conditions are equivalent:*

(1) G is a proper interval graph,

(2) G is an unit interval graph,

(3) G is a $K_{1,3}$-free interval graph.

To conclude, we must precise that proper interval graphs have significant algorithmic properties: recognizing them or finding a minimum coloring of their vertices can be done in linear time by simple algorithms [5, 7].

3 The MES problem for proper interval graphs

In order to show that the MES problem is solvable in linear time for proper interval graphs, we proceed in three stages. First, we will prove that for a n-vertex proper interval graph G, there exists a minimum partition $S_1, \dots, S_{\chi(G)}$ into stables such that each stable S_i has a size at most k (a) or each stable S_i has a size at least k (b). Then we will show that in the case (a), $\chi(G, k) = \chi(G)$ and in the case (b), $\chi(G, k) = \lceil \frac{n}{k} \rceil$, which will allow us to conclude that $\chi(G, k) = \max(\chi(G), \lceil \frac{n}{k} \rceil)$. Finally, we will prove that always there exists a partition of G into stables such that the intervals satisfy a certain order (which will be described precisely in Lemma 3). Remind that an interval representation of G is denoted by $\{I_v\}_{v \in V}$, let us begin the first stage.

Lemma 1 *Let $G = (V, E)$ be a proper interval graph, $k \in \mathbb{N}$ and $q \in \{\chi(G), \dots, n\}$. Then there exists a partition $S_1, \dots, S_q$ of G into stables such that one of the two conditions below is satisfied:*

(a) for all $i = 1, \dots, q$, $|S_i| \leq k$,

(b) for all $i = 1, \dots, q$, $|S_i| \geq k$.

Proof. Let $S_1, \dots, S_q$ be a partition of G into stables. If there are some S_i and S_j such that $|S_i| > k$ and $|S_j| < k$, we employ the following algorithm to bring us back to the cases (a) or (b). The principle of this one is to exchange some vertices of the stables S_i and S_j in order to have one the two new stables of size k.

Algorithm A
Input: $S_1, \dots, S_q$ a partition of G into stables;
Output: $S_1, \dots, S_q$ satisfying (a) or (b);

begin;
Stage 1:
if $\forall i$, $|S_i| \leq k$ do return $S_1, \dots, S_q$;
if $\forall i$, $|S_i| \geq k$ do return $S_1, \dots, S_q$;
choose two stables S_i and S_j such that $|S_i| > k$ and $|S_j| < k$;
Stage 2:
exchange some vertices of S_i and S_j in order to redefine

two new stables S_i and S_j such that $|S_i| = k$ or $|S_j| = k$;
goto Stage 1;
end;

We are now going to show the correctness of this algorithm by proving there is always an exchange of vertices as described in Stage 2. First, we can prove the next claim.

Claim 1 *Let S_i and S_j be two stables of G such that $|S_i| = k + r_i$ and $|S_j| = k - r_j$ with $r_i, r_j \in \mathbb{N}^*$. There is an exchange of vertices of the stables S_i and S_j redefining these such that $|S_i| = k + r_i - \alpha$ and $|S_j| = k - r_j + \alpha$ with $\alpha \in \{1, \dots, \min(r_i, r_j)\}$.*

Proof of Claim 1. Let $S_i = \{I_{i_1}, \dots, I_{i_{k+r_i}}\}$ and $S_j = \{I_{j_1}, \dots, I_{j_{k-r_j}}\}$ where $I_{i_1} \prec \dots \prec I_{i_{k+r_i}}$ and $I_{j_1} \prec \dots \prec I_{j_{k-r_j}}$. First, if there is an interval $I \in S_i$ such that $\forall J \in S_j$, $I \cap J = \emptyset$ then the exchange of vertices $S_i \leftarrow S_i \setminus \{I\}$ and $S_j \leftarrow S_j \cup \{I\}$ proves the lemma with $\alpha = 1$. Oterwise, each interval in S_i is intersected by at least one interval of S_j. Since the graph G is $K_{1,3}$-free (Proposition 2), each interval of S_j can intersect at most two intervals of S_i, which must be consecutive. There are at least k pairs of consecutive intervals in S_i ($r_i \geq 1$). On the other hand, there are at most $k-1$ intervals in S_j ($r_j \geq 1$). Consequently, there exists necessarily one pair of consecutive intervals $\{I_{i_u}, I_{i_{u+1}}\} \in S_i$ with $u \in \{1, \dots, k + r_i - 1\}$ such that $\{\nexists J \in S_j \mid J \cap I_{i_u} \neq \emptyset$ and $J \cap I_{i_{u+1}} \neq \emptyset\}$ (see Fig. 2 for an example).

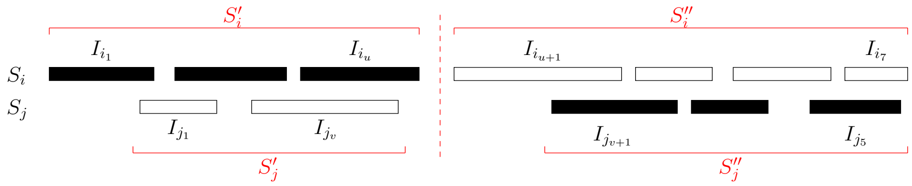


Fig. 2. An example with $k = 6$: from a stable of size 7 and a stable of size 5, we redefine two stables of size 6 (the black and the white).

In this way, we can define $S_i = S_i' \cup S_i''$, with $S_i' = \{I_{i_1}, \dots, I_{i_u}\}$ and $S_i'' = \{I_{i_{u+1}}, \dots, I_{i_{k+r_i}}\}$, and $S_j = S_j' \cup S_j''$, with $S_j' = \{I_{j_1}, \dots, I_{j_v}\}$ and $S_j'' = \{I_{j_{v+1}}, \dots, I_{j_{k-r_j}}\}$, such that $I_{j_v} \cap I_{i_{u+1}} = \emptyset$ and $I_{i_u} \cap I_{j_{v+1}} = \emptyset$. By the previous discussion, we have $|S_j'| < |S_i'|$ and $|S_j''| < |S_i''|$ and so we can pose $|S_j''| = |S_i''| - \alpha$ with $\alpha \in \{1, \dots, \min(r_i, r_j)\}$. Consequently, we can do the exchange of vertices $S_i \leftarrow S_i' \cup S_j''$ and $S_j \leftarrow S_j' \cup S_i''$ in order to have $|S_i| = k + r_i - \alpha$ and $|S_j| = k - r_j + \alpha$.

(Proof of Claim 1) $\diamondsuit$

From Claim 1, we can establish the next result which definitely allows us to conclude the correctness of Algorithm A.

Claim 2 *Let S_i and S_j be two stables of G such that $|S_i| = k + r_i$ and $|S_j| = k - r_j$ with $r_i, r_j \in \mathbb{N}^*$. There is an exchange of vertices of the stables S_i and S_j redefining these such that:*

- *if* $r_i = r_j$ *then* $|S_i| = k$ *and* $|S_j| = k$,
- *if* $r_i > r_j$ *then* $|S_i| = k + r_i - r_j$ *and* $|S_j| = k$,
- *if* $r_i < r_j$ *then* $|S_i| = k$ *and* $|S_j| = k - r_j + r_i$.

Proof of Claim 2. By successive applications of Claim 1 while there exist S_i and S_j such that $|S_i| > k$ or $|S_j| < k$, we obtain the desired result.

(Proof of Claim 2) $\diamondsuit$

$\diamondsuit$

Corollary 1 *A proper interval graph G admits an uniform q-coloring for $q \in \{\chi(G), \ldots, n\}$.*

Proof. Let $S_1, \ldots, S_q$ be a partition of G into q stables. By applying Lemma 1 to $S_1, \ldots, S_q$ with $k = \lfloor \frac{n}{q} \rfloor$ and then $k = \lceil \frac{n}{q} \rceil$, we will effectively have an uniform q-coloring of $S_1, \ldots, S_q$.

$\diamondsuit$

Now we approach the second part of the demonstration. The reader will notice that the next lemma is stated for all interval graphs.

Lemma 2 *Let $G = (V, E)$ be an interval graph with n vertices and $k \in \mathbb{N}$. Let $S_1, \ldots, S_{\chi(G)}$ be a minimum partition of G into stables.*

- *If for all $i = 1, \ldots, \chi(G)$, $|S_i| \leq k$ then $\chi(G, k) = \chi(G)$,*
- *If for all $i = 1, \ldots, \chi(G)$, $|S_i| \geq k$ then $\chi(G, k) = \lceil \frac{n}{k} \rceil$.*

Proof. The first point is immediate. The proof of the second assertion is algorithmic, notably it is based on the next claim.

Claim 3 *Let $S_1, \ldots, S_t$ be t stables of G such that:*

- $t \in \{1, \ldots, k\}$,
- *for* $i \in \{1, \ldots, t\}$, $|S_i| = k + r_i$, $r_i \in \{1, \ldots, k-1\}$,
- $r = \sum_{i=1}^{t} r_i$ *and* $r \in \{1, \ldots, k\}$.

Then there exists a stable $S' = \{I'_1, \ldots, I'_r\}$ of size r such that for $i \in \{1, \ldots, t\}$, r_i intervals of S' belong to S_i (c). In other words, there exists a partition of the set of intervals of the t stables $S_1, \ldots, S_t$ into t stables of size k and one stable of size r.

Proof of Claim 3. We propose an algorithm for the construction of S'. First, let us consider that the intervals of each stable are ordered according to the relation $\prec$. We will call the rank of an interval in a stable its number in this order. The principle of the algorithm is the following: we select at stage j, $j \in \{0, \ldots, r-1\}$, the interval I'_{r-j} (of rank $r - j$ in S') which has the biggest left endpoint among intervals of rank $k + r_i - j$ of stables S_i, $i \in \{1, \ldots, t\}$, in which we have not already selected r_i intervals. The complete algorithm is detailled below (see Fig. 3 for an example of its execution).

Algorithm B
Input: a set of stables $S_1, \ldots, S_t$ satisfying the conditions of Claim 3 such that:
$S_1 = \{I_{1,1}, \ldots, I_{1,k+r_1}\}$ with $I_{1,1} \prec \cdots \prec I_{1,k+r_1}$,
$\vdots$
$S_t = \{I_{t,1}, \ldots, I_{t,k+r_t}\}$ with $I_{t,1} \prec \cdots \prec I_{t,k+r_t}$;
Output: a stable $S' = \{I'_1, \ldots, I'_r\}$ satisfying the condition (c);

```
begin;
    S' ← ∅, j ← 0;
    for each i ∈ {1, ..., t} do
        let q_i ← 0 be the number of intervals selected in the stable S_i;
    while |S'| < r do
        F ← ∅;
        for each i ∈ {1, ..., t} do
            if q_i < r_i do
                F ← F ∪ {I_{i,k+r_i-j}};
        select in F the interval I'_{r-j} such that ∀I ∈ F, left(I) ≤ left(I'_{r-j});
        let s be the index of the stable which I'_{r-j} belongs to;
        q_s ← q_s + 1, S' ← S' ∪ {I'_{r-j}}, j ← j + 1;
    return S';
end;
```

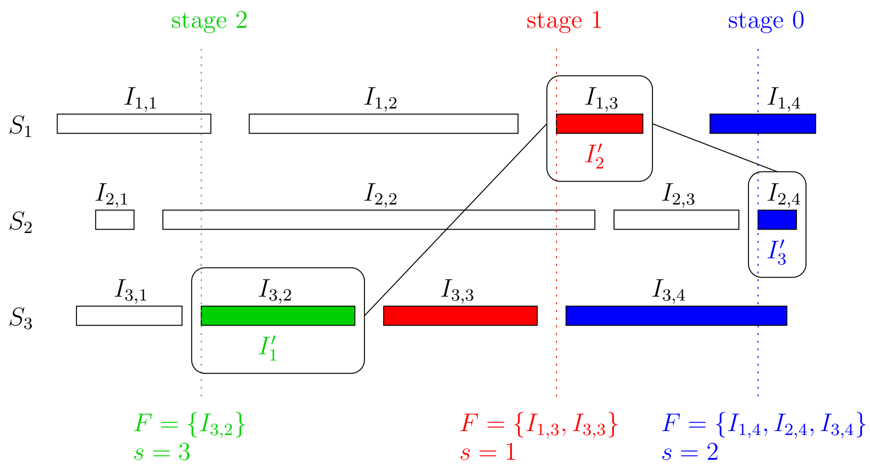


Fig. 3. An example of the execution of Algorithm B: from 3 stables of size 4, we extract the stable $\{I'_1, I'_2, I'_3\}$ of size 3 which owns an interval in each of the initial stables S_1, S_2, S_3.

Now let us prove the correctness of Algorithm B. The major point of the validity relies on the fact that the interval I'_{r-j} selected at stage j cannot intersect I'_{r-j-1} selected at stage $j+1$, i.e. Algorithm B finds a valid stable S'. Indeed, without loss of generality, let us consider that $I'_{r-j} \in S_u$ and $I'_{r-j-1} \in S_v$ with $u, v \in \{1, \ldots, t\}$, i.e. $I'_{r-j} \equiv I_{u,k+r_u-j}$ and $I'_{r-j-1} \equiv I_{v,k+r_v-j-1}$. Supposing that $I'_{r-j-1} \cap I'_{r-j} \neq \emptyset$, we would have $left(I_{u,k+r_u-j}) \leq right(I_{v,r+r_v-j-1})$

and so $left(I_{u,k+r_u-j}) < left(I_{v,k+r_v-j})$. Now $I_{v,k+r_v-j-1} \in F$ at stage $j+1$ implies that $I_{v,k+r_v-j} \in F$ at stage j. Consequently, $I_{u,k+r_u-j} \equiv I'_{r-j}$ would not be the interval which has the biggest left endpoint among intervals of F at stage j: a contradiction. Now, by the algorithm and the fact that $r \leq k$, we deduce that this stable S' satisfies the condition (c).

(Proof of Claim 3) $\diamondsuit$

Therefore, we can prove Lemma 2 as follows. We apply Claim 3 with $r = k$ to extract stables of exact size k while there exist in the partition at least k stables of size greater than k. When it remains less than k stables in the partition, while the sum of their r_i values is larger than k, we still apply Claim 3 with $r = k$ to extract stables of size k. Finally, when $r \leq k$, Claim 3 allows us to conclude.

$\diamondsuit$

Corollary 2 *Let G be an interval graph with n vertices and m edges. Let $S_1, \ldots, S_q$ be a partition of G into stables with $q \in \{\chi(G), \ldots, n\}$. If for all $i = 1, \ldots, q$, $|S_i| \geq k$ then the MES problem for G is solvable in linear time.*

Proof. As we did it to prove Lemma 2, we can clearly use Claim 3 and Algorithm B to design a linear-time algorithm to solve the MES problem in this case. Indeed, the time complexity of Algorithm B is linear in the number of considered intervals if these are ordered according to the increasing left endpoints. Such an ordered interval representation of G can be computed in $O(n+m)$ time (G given by adjacency lists in input) [5]. On the other hand, if we have an interval representation of G (i.e. G given by the list of endpoints of intervals in input), a simple sorting suffices to obtain this order on the set of intervals.

$\diamondsuit$

Theorem 1 *Let G be a proper interval graph with n vertices and $k \in \mathbb{N}$,*

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

Proof. By Lemma 1 with $q = \chi(G)$ and Lemma 2.

$\diamondsuit$

Here is the third and last stage of the demonstration. As previously, we will denote by $I_{i,j}$ an interval of rank j in a stable S_i.

Lemma 3 *Let $G = (V, E)$ be a proper interval graph with n vertices and $q \in \{\chi(G), \ldots, n\}$. Let $<$ be a linear order on V such that for all $I_i, I_j \in V$, $I_i < I_j$ iff $left(I_i) < left(I_j)$ or ($left(I_i) = left(I_j)$ and $right(I_i) \leq right(I_j)$). Let $p = n \mod q$ and $t = \lfloor \frac{n}{q} \rfloor$. Then there exists a partition $S_1, \ldots, S_q$ of G into q stables such that:*

(i) $\forall i \in \{1, \ldots, p\}, |S_i| = t + 1$,

(ii) $\forall i \in \{p+1, \ldots, q\}, |S_i| = t$,

(iii) $\forall i \in \{1, \ldots, q-1\}$, $\forall j \in \{1, \ldots, t\}$, $I_{i,j} < I_{i+1,j}$ *and* $I_{q,j} < I_{1,j+1}$,

(iv) $\forall i \in \{1, \ldots, p-1\}$, $I_{i,t+1} < I_{i+1,t+1}$.

Proof. The points (i) and (ii) are obtained by the application of Corollary 1. Now, let us suppose that we have $S_1, \ldots, S_q$ satisfying the points (i) and (ii), we can obtain the points (iii) and (iv) from the following claim.

Claim 4 *Let $S_i = \{I_{i,1}, \ldots, I_{i,t+1}\}$ and $S_j = \{I_{j,1}, \ldots, I_{j,t+1}\}$ be two stables of G with $i, j \in \{1, \ldots, p\}$. There is an exchange of vertices of the stables S_i and S_j redefining these such that $I_{i,1} < I_{j,1} < \cdots < I_{i,t+1} < I_{j,t+1}$.*

Proof of Claim 4. If $I_{j,1} < I_{i,1}$ then we can redefine S_i and S_j as follows: $S_i \leftarrow \{I_{j,1}, \ldots, I_{j,t+1}\}$ and $S_j \leftarrow \{I_{i,1}, \ldots, I_{i,t+1}\}$. Now let us suppose the claim true up to the rank $r \in \{1, \ldots, t\}$, i.e. S_i and S_j are such that $I_{i,1} < I_{j,1} < \cdots < I_{i,r} < I_{j,r} < I_{j,r+1} < I_{i,r+1}$. The intervals are proper, so $I_{i,r} \cap I_{j,r+1} = \emptyset$ and $I_{j,r} \cap I_{i,r+1} = \emptyset$ (see Fig. 4). Consequently, we can redefine S_i and S_j as follows: $S_i \leftarrow \{I_{i,1}, \ldots, I_{i,r}, I_{j,r+1}, \ldots, I_{j,t+1}\}$ and $S_j \leftarrow \{I_{j,1}, \ldots, I_{j,r}, I_{i,r+1}, \ldots, I_{i,t+1}\}$ and the claim is now true at the rank $r+1$ too. By repeating this recurrent constructive process up to the rank t, we can prove the claim.

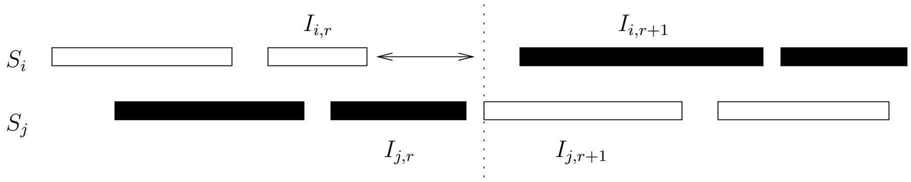


Fig. 4. An example of a redefinition of two stables S_i and S_j in the proof of Claim 4: the new stables are the black and the white.

(Proof of Claim 4) $\diamondsuit$

It is easy to see that the claim is always true with $S_i = \{I_{i,1}, \ldots, I_{i,t+1}\}$ and $S_j = \{I_{j,1}, \ldots, I_{j,t}\}$. Thus, we can apply Claim 4 to order successively according to $<$ the intervals of rank 1 in the partition, next of rank 2, and so on, until the rank $t+1$.

$\diamondsuit$

Now we conclude this section by giving the main result of the paper.

Theorem 2 *The mutual exclusion scheduling problem is solvable in linear time for proper interval graphs.*

Proof. Let $G = (V, E)$ be a proper interval graph with n vertices and m edges and $k \in \mathbb{N}$. Thanks to the previous results, we can now define an efficient algorithm to solve the MES problem for G. First, we can calculate $\chi(G, k)$, i.e. the minimum partition of G into stables of size at most k, by computing $\chi(G)$ and using Theorem 1. Then, according to the order $<$ defined in Lemma 3, we can construct greedily the $\chi(G, k)$ stables satisfying the four points of Lemma 3. The complete algorithm is detailled below (see Fig. 5 for an example of its execution).

Algorithm C
Input: $G = (V, E)$ a proper interval graph with n vertices and $k \in \mathbb{N}$;
Output: a minimum partition of G into stables of size at most k;

begin;

order $V = \{I_1, \ldots, I_n\}$ according to the order $<$;

compute $\chi(G)$ and $\chi(G, k) \leftarrow \max(\chi(G), \lceil \frac{n}{k} \rceil)$;

let $S_1, \ldots, S_{\chi(G,k)}$ be a set of stables;

$S_1 \leftarrow \emptyset, \ldots, S_{\chi(G,k)} \leftarrow \emptyset$ and $j \leftarrow 1$;

for i from 1 to n do

$S_j \leftarrow S_j \cup \{I_i\}$, $j \leftarrow j + 1$;

if $j > \chi(G, k)$ do $j \leftarrow 1$;

return $S_1, \ldots, S_{\chi(G,k)}$;

end;

The correctness of Algorithm C follows from the previous discussion. To conclude, let us analyse its time complexity. Ordering V according to the linear order $<$ can be done in $O(n + m)$ time [5] or in $O(n \log n)$ time if we have an interval representation of G. Now, having this order on V, computing $\chi(G)$, and so $\chi(G, k)$, can be done in $O(n)$ time [7]. Finally, the greedy construction of the $\chi(G, k)$ stables is done in $O(n)$ time. Consequently, Algorithm C can run in $O(n + m)$ time (G given by adjacency lists in input) or in $O(n \log n)$ time if we have an interval representation of G (i.e. G given by the list of endpoints of intervals in input). We can notice that in this last case, if the list of endpoints is already sorted then Algorithm C runs in $O(n)$ time.

$\diamondsuit$

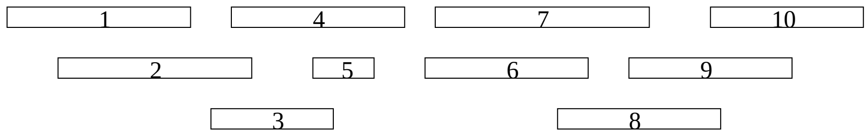
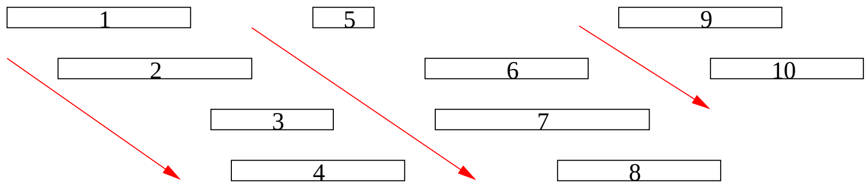


Fig. 5. An example of the execution of Algorithm C with $k = 3$: above is a set of 10 intervals inducing a proper interval graph G with $\chi(G) = 3$ and below is the partition of size $\chi(G, 3) = \max(3, \lceil \frac{10}{3} \rceil) = 4$ made by the algorithm.



Remark. Theorem 2 simplifies and generalizes a recent result of M.G. Andrews et al. about finding a maximum matching among disjoint proper intervals which is equivalent to the MES problem for proper interval graphs with $k = 2$ (see [1] pages 284-287). They showed that this problem can be solved in linear time by using a “red-blue matching algorithm” [11] (given the set of endpoints of the intervals sorted).

Corollary 3 *The problem of finding an uniform coloring is solvable in linear time for proper interval graphs.*

Proof. Let G be a proper interval graph with n vertices. By assuming $k = \lceil \frac{n}{\chi(G)} \rceil$ or directly $\chi(G,k) = \chi(G)$ in Algorithm C, we find an uniform coloring of G in linear time.

$\diamondsuit$

We conclude this paper with the complexity of the working schedules planning problem mentionned in the introduction.

Corollary 4 *The WSP problem is solvable in linear time if the intersection graph of tasks is $K_{1,3}$-free.*

We can note that this condition can be tested in $O(n \log n + m)$ time by constructing the intersection graph of tasks and using the recognition algorithm of unit interval graphs of [5].

Acknowledgements

We would like to thank Professor Michel Van Caneghem and Professor Victor Chepoi for their precious advice and the firm PROLOGIA - Groupe Air Liquide for its grant.

References

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

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

[3] J. Blazewick, K.H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Scheduling computer and manufacturing processes*. Springer, Heidelberg, 1996.

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

[5] D.G. Corneil, H. Kim, S. Natarajan, S. Olariu, and A.P. Sprague. Simple linear time recognition of unit interval graphs. *Information Processing Letters*, 55 (2):99–104, 1995.

[6] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Computer Science and Applied Mathematics. Academic Press, New-York, 1980.

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

[8] S. Irani and V. Leung. Scheduling with conflicts, and applications to traffic signal control. In *SIAM Symposium on Discrete Algorithms*, pages 85–94, SODA 96.

[9] K. Jansen. The mutual exclusion scheduling problem for permutation and comparability graphs. Technical Report 98-8, DIMACS, January 1998.

[10] R.M. Karp. Reduciblity among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, Plenum, New-York, 1972.

[11] S.K. Kim. Optimal parallel algorithms on sorted intervals. In *Proc. 27th Annual Allerton Conference on Communication, Control and Computing*, pages 766–775, Monticello, IL, 1989.

[12] J. Krarup and D. de Werra. Chromatic optimization: limitations, objectives, uses, references. *European Journal of Operations Research*, 11:1–19, 1982.

[13] Z. Lonc. On complexity of some chain and antichain partition problem. In *Graph Theoretical Concepts in Computer Science*, pages 97–104. LNCS 570, WG 91.

[14] F.S. Roberts. Indifference graphs. In Frank Harary, editor, *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, New-York, 1969.

[15] F.S. Roberts. *Graph Theory and its Applications to Problems of Society.* SIAM, Philadelphia, 1978.