The mutual exclusion scheduling problem for permutation and comparability graphs*

Klaus Jansen
Max-Planck Institute for Computer Science
Im Stadtwald
66 123 Saarbrücken, Germany
jansen@mpi-sb.mpg.de

Abstract

In this paper, we consider the mutual exclusion scheduling problem for comparability graphs. Given an undirected graph G and a fixed constant m, the problem is to find a minimum coloring of G such that each color is used at most m times. The complexity of this problem for comparability graphs was mentioned as an open problem by Möhring (1985) and for permutation graphs (a subclass of comparability graphs) as an open problem by Lonc (1991). We prove that this problem is already NP-complete for permutation graphs and for each fixed constant $m \geq 6$.

1 Introduction

The following problem arises in scheduling theory: there are n jobs that must be completed on m processors in minimum time t. A processor can execute only one job at a time, and each job requires one time unit for completion. The scheduling is complicated by additional resources (e.g. I-O devices, communication links). A job can only be scheduled onto a processor in a given time unit after it has an exclusive lock on all required resources. Problems of this form have been studied in the Operations Research literature [6, 17, 18]. Another problem arises in load balancing the parallel solution of partial differential equations (pde's) by domain decomposition [5, 2]. The domain for the pde's is decomposed into regions where each region corresponds to a subcomputation. The subcomputations are scheduled on m processors so that subcomputations corresponding to regions that touch at even one point are not performed simultaneously. Other applications are in constructing school course time tables [16] and scheduling in communication systems [14].

1.1 Mutual exclusion scheduling

These scheduling problems can be solved by creating an undirected graph $G = (V, E)$ with a vertex for each of the n jobs, and an edge between every pair of conflicting jobs. In each time

*This work was done while the author was associated with the University Trier and supported in part by DIMACS and by EU ESPRIT LTR Project No. 20244 (ALCOM-IT).

step, we can execute any subset $U \subset V$ of jobs for which $|U| \leq m$ and U is an independent set in G. A minimum length schedule corresponds to a partition of V into a minimum number t of such independent sets. Baker and Coffman called this graph theoretical problem *Mutual Exclusion Scheduling* (short: MES). Bodlaender and Jansen [7] studied the decision problem of a complementary scheduling problem. Their initial interest was in *compatibility scheduling* (short: CS) which has the same instance and makespan objective function as MES but has a different meaning on the adjacency in G: if two tasks are adjacent in G then they can not be executed on the same processor. Therefore, in MES an independent set is processed in a time unit, whereas in CS an independent set is executed on one processor.

Lonc [19] showed that MES for split graphs can be solved in polynomial time. He proved that MES is NP-complete for complements of comparability graphs and fixed $m \geq 3$ and that MES is polynomial solvable for complements of interval graphs and every m and for cographs and fixed m. However, Bodlaender and Jansen [7] showed that MES is NP-complete when G is restricted to cographs, bipartite or interval graphs. They also proved the following result: if either t or m is a fixed constant, then MES is in P for cographs; if t is a fixed constant, then the problem is in P for interval graphs, and if m is a constant, then it is in P for bipartite graphs. MES remains NP-complete for bipartite graphs and any fixed $t \geq 3$, and for interval graphs and any fixed $m \geq 4$. Independently, Hansen et al. [13] proved that MES restricted to biparite graphs and fixed m is solvable in polynomial time.

Corneil [10] reports that Kirkpatrick has shown the NP-completeness of MES restricted to chordal graphs and fixed $m \geq 3$. For $m = 2$, MES is equivalent to the maximum matching problem in the complement graph and, therefore, in P. On the other hand, for $m = 3$ the complexity of MES is the same as that of the NP-complete problem partition into triangles in the complement graph. Moreover, Baker and Coffman [2] proved that MES is in P for forests and general $t, m \in \mathbb{N}$. A linear time algorithm was proposed in [15] for MES restricted to graphs with constant treewidth and fixed m. Furthermore, MES is NP-complete for complements of line graphs and fixed $m \geq 3$ [9] and polynomial for line graphs and every fixed m [1].

In this paper, we prove the following main result.

Theorem 1.1 *For each fixed constant $m \geq 6$, the problem MUTUAL EXCLUSION SCHEDULING is NP-complete for permutation graphs (and also comparability graphs).*

1.2 M-machine scheduling

A *partial order* will be denoted by $P = (V, <_P)$ where V is the set of vertices and $<_P$ is the order relation, i.e. an irreflexive and transitive relation whose pairs $(a, b) \in <_P$ are written as $a <_P b$ (for $a, b \in V$). If the relation is clear, we write $<$ instead of $<_P$. Two elements $a, b \in V$ are *comparable* in P if $a < b$ or $b < a$. A set of pairwise comparable elements is called a *chain*, and a set of pairwise incomparable elements is called an *antichain*. An element a is *minimal* in P, if it has no predecessor. With each partial order $P = (V, <)$, we may associate an undirected graph $G(P)$ as follows. The vertices of $G(P)$ are the elements in V, and two vertices are connected by an edge in $G(P)$ if they are comparable in P. $G(P)$ is called the comparability graph of P. In general, an undirected graph G is called a *comparability graph*, if $G = G(P)$ for some partial order on its vertex set. Algorithmic aspects of comparability graphs are given e.g. in [20].

Let us consider another famous scheduling problem. The *m-machine scheduling* problem with unit times can be modeled by a partition of a partial order $P = (V, <)$ into antichains $A_1, A_2, \ldots, A_t$ such that $|A_i| \leq m$ and

(*) each set A_i consists of some minimal elements of $V \setminus (A_1 \cup \ldots A_{i-1})$ for $i = 1, \ldots, t$.

Inspite of intensive research by several scientists, the complexity status of the problem of finding such a partition with minimum t for each fixed constant $m \geq 3$ still remains unsolved [4]. In fact, this problem is mentioned already in the original list of ten basic open problems in complexity theory [11]. For an overview about different classes of partial orders and complexity results, we refer to [4, 22]. Möhring [21] proposed 1985 a related problem. He asked for the complexity of the problem with condition $(*)$ dropped. This amounts to the mutual exclusion problem restricted to comparability graphs and fixed constant $m \geq 3$. Furthermore, Lonc [19] asked 1991 for the complexity of MES restricted to permutation graphs (a subclass of all comparability graphs) and fixed $m \geq 3$.

In the classical m-machine scheduling problem, the jobs in antichains $A_1, \ldots, A_t$ with $|A_i| \leq m$ corresponding to a partition that satisfies property $(*)$ can be executed on m processors one after another in t time steps. This follows from the fact that each vertex v in an antichain A_i has no predecessor in $V \setminus (A_1 \cup \ldots \cup A_{i-1})$. The makespan of the corresponding schedule is equal to the number of antichains. If we drop condition $(*)$, it is possible that the antichains can not be ordered to form a feasible schedule. Furthermore for each $m \geq 3$, there exists a comparability graph G and a number t such that the answer to the problem MES is *yes* for G and t, but the answer to the m machine scheduling problem is *no* for t and any partial order P (such that G is the comparability graph of P). However, it could be possible to modify the graph in the construction or to extend the ideas in the proof to get a NP-completeness result for the m - machine scheduling problem.

A *linear order* is a partial order without incomparable elements. A linear extension of a partial order $P = (V, <_P)$ is a linear order $L = (V, <_L)$ on the same ground set V that extends P, i.e. $a <_P b$ implies $a <_L b$ for all $a, b \in V$. Linear orders can be written as sequences $L = x_1 \ldots x_n$ defining the order relation $x_1 <_L x_2 <_L \ldots <_L x_n$. The *dimension* $dim(P)$ of a partial order P is the smallest number of linear extensions $L_1, \ldots, L_k$ of P, $L_i = (V, <_i)$ whose intersection is P, i.e. $a <_P b$ if and only if $a <_i b$ for $i = 1, \ldots, k$. A partial order is called k - *dimensional* for $k \in \mathbb{N}$, if $dim(P) \leq k$.

Baker et al. [3] have given the following graph theoretic characterization: A partial order P is 2 - dimensional, if and only if the complement graph $G(P)^c$ is also a comparability graph. This implies that a graph G is a comparability graph of a 2 - dimensional partial order if and only if G and its complement graph G^c are comparability graphs. These graphs have been studied also under the name *permutation graphs* [23, 12].

Let $\Pi = (i_1, \ldots, i_n)$ be a permutation of $\{1, \ldots, n\}$. We denote with $\Pi^{-1}(i)$ the position of i in Π. A graph $G = (V, E)$ with $V = \{1, \ldots, n\}$ is a *permutation graph*, if there is a permutation Π such that $\{i, j\} \in E$, if and only if $(i - j) \cdot (\Pi^{-1}(i) - \Pi^{-1}(j)) < 0$. Different techniques for solving algorithmic problems on a permutation graph are given in [8]. In our proof, the permutation graph is given as a *set of points* $\{p_j = (j, \Pi^{-1}(j)) | 1 \leq j \leq n\}$ in the plane $\mathbb{N}^2$. An independent set $U = \{a_1, \ldots, a_u\}$ in G corresponds to the point set $S_U = \{(a_1, \Pi^{-1}(a_1)), \ldots, (a_u, \Pi^{-1}(a_u))\}$

in increasing order (with $a_1 < \ldots < a_u$ and $\Pi^{-1}(a_1) < \ldots < \Pi^{-1}(a_u)$). Moreover, a clique C corresponds to a point set S_C in decreasing order.

1.3 Main ideas

The main ideas of the NP-completeness proof are the following. First, we choose a restricted SAT problem where each variable occurs three times (once negated and twice unnegated or once unnegated and twice negated) and where each clause contains two or three literals. Let n be the number of variables and r be the number of clauses. We prove that the restricted SAT problem remains NP-complete under a separation condition: for each variable x_i there exists an index $l_i \in \{1, \ldots, r-1\}$ such that x_i appears only negated in the first l_i clauses and unnegated in the other clauses, or vice versa.

Next, we construct a pointset in a rectangle that represents the permutation graph and that simulates the SAT formula. The rectangle can be divided into n horizontal and r vertical stripes such that each horizontal stripe corresponds to a variable and that each vertical stripe corresponds to a clause. If x_i or $\bar{x}_i$ occurs in a clause c_j then we place a point in the square corresponding to the i.th horizontal and j.th vertical stripe. To simulate the variable setting, we differ between two parallel lines in each horizontal stripe and place the points on the upper or lower line in dependence whether the variable is negated or unnegated.

Finally, we place some points (in two chains) around the rectangle such that each independent set in an optimum solution of the restricted coloring problem for the permutation graph covers only points in a horizontal or vertical stripe. Our proof was inspired by the NP-completeness proof of K. Wagner [24].

2 A restricted satisfiability problem

To prove the main theorem, we consider the following restricted satisfiability problem:

Restricted SAT

Given: A set of unnegated variables $X = \{x_1, \ldots, x_n\}$ and negated variables $\bar{X} = \{\bar{x_1}, \ldots, \bar{x_n}\}$, a collection of clauses $c_1, \ldots, c_r$ over $X \cup \bar{X}$ (subsets of $X \cup \bar{X}$) such that

- **(i)** each clause c_i contains two or three literals $y \in X \cup \bar{X}$,
- **(ii)** each variable x_j appears either twice unnegated and once negated or twice negated and once unnegated in the clauses,
- **(iii)** no clause c_i contains a pair x_j, $\bar{x_j}$.

Question: Does there exist a truth mapping for the variables such that each clause is satisfied?

The *literals* of a clause c_i are denoted by $y_{i,1}$ and $y_{i,2}$ (and $y_{i,3}$ if we have three literals).

Lemma 2.1 *The restricted SAT problem is NP-complete.*

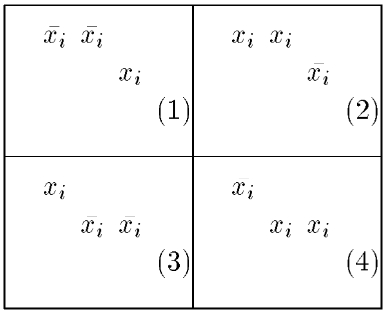

Figure 1: The four cases for a variable

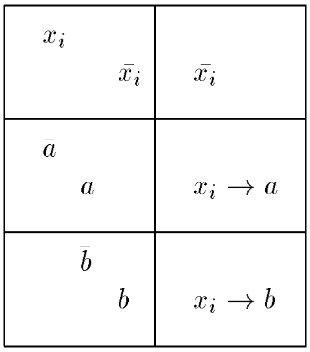

Figure 2: The transformation for case 1

Proof: By a reduction from the NP-complete SAT problem where each clause contains exactly three literals [11]. We may assume that each variable appears not only negated or unnegated; otherwise we replace x_j (or $\bar{x_j}$) by the truth values *TRUE* and get a smaller instance. If a clause contains only one literal $y \in \{x_j, \bar{x_j}\}$ (after this reduction), we replace the corresponding variable x_j in all clauses by *TRUE* if $y = x_j$ or by *FALSE* if $y = \bar{x_j}$. If a clause contains a pair x_j and $\bar{x_j}$, we can remove the clause. Then, we have to consider the following remaining cases:

Case 1: A variable x_j appears once unnegated and once negated. In this case, we choose a new variable a_1, insert the clauses $(x_j \vee \bar{a_1}), (\bar{x_j} \vee a_1)$ and replace the second occurrence of x_j in the old clauses by a_1.

Case 2: A variable x_j appears $k \geq 4$ times in the clauses. In this case, we choose new variables $a_1, a_2, \ldots, a_{k-1}$, insert the clauses $(x_j \vee \bar{a_1})$, $(a_1 \vee \bar{a_2})$, ..., $(a_{k-1} \vee \bar{x_j})$ and replace the h.th occurrence of x_j in the old clauses by a_{h-1} for $1 < h \leq k$.

In both cases, we obtain that the truth value x_j is equal to $a_1 = \ldots = a_{k-1}$ and that each variable appears now in three different clauses. □

For our reduction, we need a further property for the restricted SAT problem. Let us assume that the clauses are numbered by $1, \ldots, r$ and that the number of the j.th occurrence of variable x_i in the clauses is denoted by $i[j]$. Since each clause contains a variable x_i at most once, we have $i[1] < i[2] < i[3]$. We say that an instance I has the **separation** property if for each variable x_i

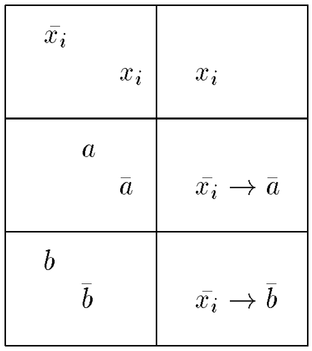

Figure 3: The transformation for case 2

one of the following cases is true (observe that each variable appears either twice negated and once unnegated or twice unnegated and once negated):

(1) x_i appears negated in clauses $i[1]$ and $i[2]$ and unnegated in $i[3]$,

(2) x_i appears unnegated in clauses $i[1]$ and $i[2]$ and negated in $i[3]$,

(3) x_i appears negated in clauses $i[2]$ and $i[3]$ and unnegated in $i[1]$,

(4) x_i appears unnegated in clauses $i[2]$ and $i[3]$ and negated in $i[1]$.

The separation property means that only the cases in Figure 1 are possible. This property is necessary for our reduction in section 3. The Figure 1 illustrates how the points corresponding to x_i and $\bar{x}_i$ are placed in the horizontal stripe corresponding to variable x_i.

Lemma 2.2 *The restricted SAT problem with separation property is also NP-complete (even if there are only case* (3) *or* (4) *variables).*

Proof: By a reduction from the restricted SAT problem. For each variable x_i consider the following two possibilities:

Case 1: Variable x_i appears twice unnegated and once negated in the clauses. In this case, we choose two new variables a, b and insert the new clauses $(x_i \vee \bar{a})$, $(a \vee \bar{b})$, $(b \vee \bar{x}_i)$ at the beginning of the sequence of clauses. Furthermore, we replace the first unnegated occurrence of x_i by a and the second unnegated occurrence by b in the old clauses. Then, we obtain the ordering as illustrated in Figure 2 and observe that the variables x_i, a and b now have the desired property.

Case 2: Variable x_i appears twice negated and once unnegated in the clauses. We insert the same clauses as in case 1, but we change the order as follows: $(\bar{x}_i \vee b)$, $(\bar{b} \vee a)$, $\bar{a} \vee x_i)$. Next, we replace the first negated occurrence of x_i by $\bar{a}$ and the second negated occurrence by $\bar{b}$. Then, we obtain the ordering in Figure 3 and get the desired property for the variables x_i, a and b. □

3 The proof of the main theorem

The permutation graph in our proof is represented by a *set of points* $\{p_j = (j, \Pi^{-1}(j)) | 1 \le j \le n\}$ in the plane $\mathbb{N}^2$. We notice that an independent set $U = \{a_1, \ldots, a_u\}$ in G corresponds to the point set $S_U = \{(a_1, \Pi^{-1}(a_1)), \ldots, (a_u, \Pi^{-1}(a_u))\}$ in increasing order (with $a_1 < \ldots < a_u$ and $\Pi^{-1}(a_1) < \ldots < \Pi^{-1}(a_u)$). Moreover, a clique C corresponds to a point set S_C in decreasing order. In the following, we give a proof of the main theorem with fixed constant $m = 6$. The proof can be simply modified such that the result is true for each fixed constant $m \ge 6$.

Proof of main theorem: The theorem is proved by a reduction from the restricted SAT problem with separation property (and with only case (3) and (4) variables). Let I_1 be an instance of such problem containing a collection of r clauses $c_1, \ldots, c_r$ over a set of unnegated and negated variables $\{x_1, \bar{x_1}, \ldots, x_n, \bar{x_n}\}$. Let r_2 (and r_3) be the number of 2-clauses (3-clauses) in I_1. The *literals* of clause c_i are denoted by $y_{i,1}$ and $y_{i,2}$ (and $y_{i,3}$ if we have three literals). We denote with $\ell(i)$ the number of literals in clause c_i.

An instance I_2 of the mutual exclusion scheduling problem with bound 6 for each independent set is constructed as follows. The constructed point set for the restricted SAT instance $(x_1 \vee \bar{x}_2 \vee x_i) \wedge (\bar{x}_1 \vee x_2 \vee x_n) \wedge \ldots \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_n)$ is given in Figure 4. In this example, x_1 and x_n are case (3) variables and x_2 is a case (4) variable. The permutation graph G is given by the corresponding point set

$$\begin{aligned} P = \; & \bigcup_{i=1}^{n} [A_i \cup B_i] \cup \bigcup_{i=1}^{r} [E_i \cup D_i \cup \{a_{i,1}, a_{i,2}\}] \cup \\ & \bigcup_{1 \le i \le r, \ell(i)=3} [E'_i \cup D'_i \cup \{a_{i,3}\}] \end{aligned}$$

where

- A_i consists of one point and B_i consists of three points on the increasing line as marked in Figure 4.
- E_i (and E'_i if clause c_i contains three literals) consists of four points on the increasing line and D_i (and D'_i if c_i contains three literals) consists of one point as marked in Figure 4.
- a_{ik} is a point situated in the crossing of the decreasing line h_i and the increasing line marked by f_ℓ or g_ℓ for $y_{i,k} \in \{x_\ell, \bar{x}_\ell\}$.

We choose the line f_ℓ or g_ℓ in dependence to the cases (3) and (4) for the variable x_ℓ (as constructed in the proof of Lemma 2.2). We place the points as given in Figure 5. If the first occurrence of variable x_ℓ is unnegated, we have case (3) (otherwise we have case (4)). In case (3), the unnegated literal is placed on line f_ℓ and the negated literals on line g_ℓ. This means for $y_{i,1} = x_\ell$ that $a_{i,1}$ lies in the crossing of the lines h_i and f_ℓ. In case (4), we place the negated literal on line f_ℓ and the unnegated literals on line g_ℓ.

The maximum clique size $\omega(G)$ of the constructed permutation graph is equal to

$$t = n + r_2 + 2 \cdot r_3.$$

One maximum clique is given e.g. by

$$K = A_1 \cup \ldots \cup A_n \cup \{z_i | 1 \le i \le r\} \cup \{z'_i | E'_i \ne \emptyset\}$$

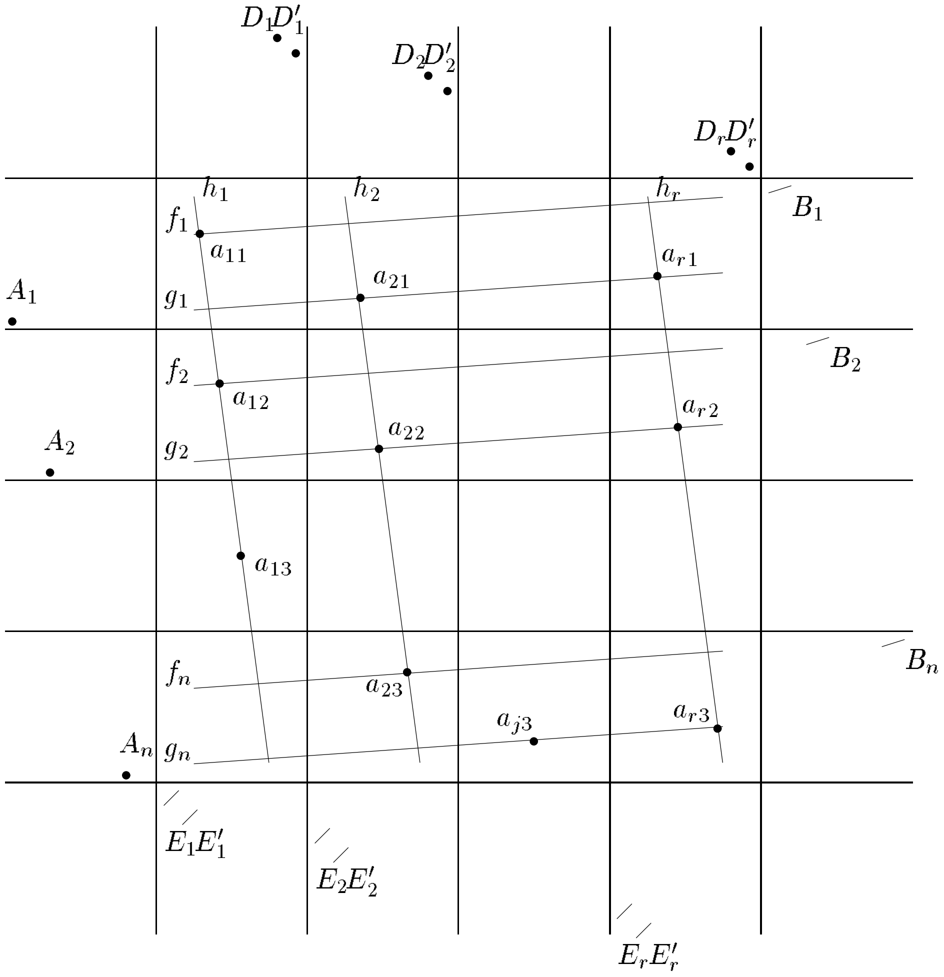

Figure 4: The point set P constructed for a restricted SAT instance $(x_1 \vee \bar{x}_2 \vee x_i) \wedge (\bar{x}_1 \vee x_2 \vee x_n) \wedge \ldots \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_n)$.

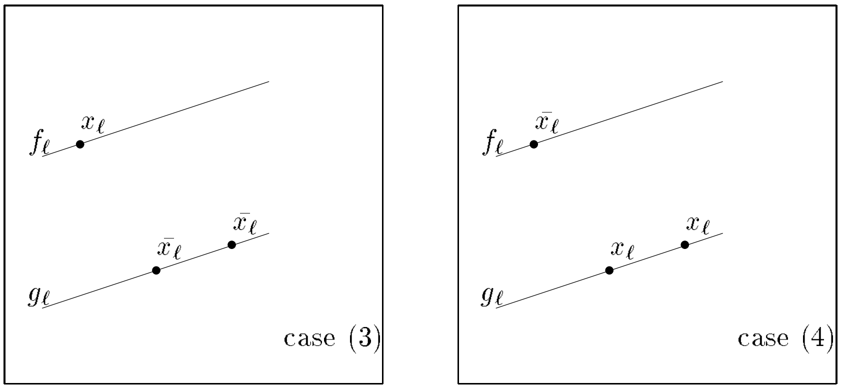

Figure 5: The way to place the points on the lines f_ℓ or g_ℓ

with arbitrarily choosen $z_i \in E_i$ and $z'_i \in E'_i$ (if c_i is a 3-clause). As illustrated in Figure 4, the point set $\{a_{i,k}|1 \leq i \leq r, 1 \leq k \leq \ell(i)\}$ can be divided into n horizontal (one stripe for each variable) and r vertical stripes (one stripe for each clause).

We prove the following **claim**: I_1 is a yes - instance of the restricted SAT problem with separation property if and only if there is a coloring of the permutation graph in instance I_2 with at most t colors where each color is used at most 6 times.

Suppose that α is a truth assignment for the variables which makes the clauses $c_1, \ldots, c_r$ true. The first n colors cover the points of $A_j \cup B_j$ and all points (at most two) on f_j if $[\alpha(x_j) = FALSE$ and x_j is a case (4) variable$]$ or $[\alpha(x_j) = TRUE$ and x_j is a case (3) variable$]$ or all points on g_j if $[\alpha(x_j) = FALSE$ and x_j is a case (3) variable$]$ or $[\alpha(x_j) = TRUE$ and x_j is a case (4) variable$]$. Then, at least one of the points $a_{i,1}, \ldots, a_{i,\ell(i)}$ is colored with one of these n colors for each clause c_i. The remaining $t - n$ colors are used to color the points of D_i, D'_i, E_i, E'_i and the remaining uncolored points in the vertical stripes. We use two colors for a 3-clause and one color for a 2-clause. This gives us a t-coloring of G such that each color is used at most 6-times.

Conversely, suppose that we have a t**-coloring** f of the permutation graph G where each color is used at most 6 times. First, we observe that

$$K = A_1 \cup \ldots \cup A_n \cup \{z_i|1 \leq i \leq r\} \cup \{z'_i|E'_i \neq \emptyset\}$$

with arbitrarily choosen $z_i \in E_i$ and $z'_i \in E'_i$ and that

$$\bar{K} = D_1 \cup \ldots \cup D_r \cup \{D'_i|1 \leq i \leq r, \ell(i) = 3\} \cup \{b_1, \ldots, b_n\}$$

with arbitrarily choosen $b_i \in B_i$ are cliques in G of size t. This implies that the vertices in the cliques K (and $\bar{K}$) must be colored differently and that points in every chain E_i (and E'_i, B_j) must be colored with the same color. This means e.g. that the number of colors $|\{f(z)|z \in E_i\}|$ is equal 1. Since the cliques K and $\bar{K}$ are disjoint, there must be an $1-1$ assignment between the vertices in K and $\bar{K}$ to obtain a t- coloring.

Feasible assignments: Since each color can be used at most 6 times, it is not possible to take the same color for a E_i chain (or E'_i chain) and a B_j chain; otherwise we have 7 vertices colored with the same color. This implies that only the following assignments (or colorings) are possible:

(1) the color set $\{f(A_j)|1 \leq j \leq n\}$ is equal to $\{f(b_j)|1 \leq j \leq n, b_j \in B_j\}$,

(2) the color set $\{f(z_i), f(z'_i)|z_i \in E_i, z'_i \in E'_i, 1 \leq i \leq r\}$ is equal to $\{f(D_i), f(D'_i)|1 \leq i \leq r\}$.

W.l.o.g. we may assume that $f(A_j) = j$ for $1 \leq j \leq n$. In the next step, we can prove that

(3) $f(A_j) = f(b_j) = j$ with $b_j \in B_j$, $1 \leq j \leq n$,

(4) $f(z_i) = f(D_i)$ with $z_i \in E_i$ and $f(z'_i) = f(D'_i)$ with $z'_i \in E'_i$ for a 3-clause c_i.

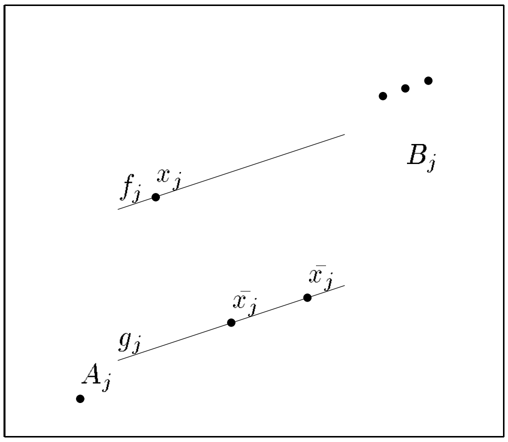

Figure 6: A horizontal stripe with points A_j and B_j

For example, consider the points in the sets A_j, B_j. We know already that the color sets

$$\begin{aligned}\{f(A_j)|1 \leq j \leq n\} &= \{f(b_j)|1 \leq j \leq n, b_j \in B_j\} \\ &= \{1, \ldots, n\}.\end{aligned}$$

Since A_1 is in conflict to all points in the chain $B_2, \ldots, B_n$, we get $f(A_1) = f(b_1) = 1$ with $b_1 \in B_1$. For the induction step, observe that A_i is in conflict to the points in $B_{i+1}, \ldots, B_n$ and to the points $A_1, \ldots, A_{i-1}$. Therefore, $f(A_i) \notin \{f(b_j)|1 \leq j \leq n, j \neq i, b_j \in B_j\}$. This implies that $f(A_i) = f(b_i)$ with $b_i \in B_i$ and proves property (3) above. A similar idea can be used to prove property (4). We notice that it is possible that $f(z_i) = f(D_i')$ and $f(z_i') = f(D_i)$. But we can exchange these colors without problems to obtain a coloring such that property (4) is satisfied.

Horizontal and vertical stripes: Finally, we look at a horizontal stripe with vertices A_j and $b_j \in B_j$ colored with the same color j.

Using the separation property for the variable x_j we get a stripe as in Figure 6 (here given for a case (3) variable x_j). Since the point A_j and the points in B_j are colored with j, only the point on f_j or the two points on g_j can be colored with color j. Notice that an independent set must form an increasing chain of points in the point model. Let U_i be the set of points colored with color i, for $1 \leq i \leq t$. We may assume that either the point on f_j or both points on g_j are colored with color j.

We define a truth assignment α for the variables $x_1, \ldots, x_n$ as follows:

$$\alpha(x_j) = \begin{cases} true & \text{if } U_j \text{ contains both points on line } g_j \text{ and if } x_j \text{ is a case (4) variable} \\ true & \text{if } U_j \text{ contains a point on line } f_j \text{ and if } x_j \text{ is a case (3) variable} \\ false & \text{otherwise.} \end{cases}$$

Since $f(z_i) = f(D_i)$ with $z_i \in E_i$ and $f(z_i') = f(D_i')$ with $z_i' \in E_i'$ and since the points $a_{i,k}$ lie on a decreasing line h_i, each independent set U_j with $n + 1 \leq j \leq t$ contains at most one point in a vertical stripe. Therefore, at least one point in each vertical stripe must be covered by the sets $U_1, \ldots, U_n$. This implies that the mapping α is a truth assignment that satisfies all clauses in I_1. $\Box$

4 Conclusion

In this paper, we have proved that mutual exclusion scheduling (MES) restricted to permutation (and comparability graphs) is NP-complete for each fixed constant $m \geq 6$. Finding the complexity of MES for smaller constants $m = 3, 4, 5$ could be a step to the solution of the famous m-machine scheduling problem with unit-times. The MES problem for permutation graphs and $m = 3$ is equivalent to the following problem [19]:

Problem S(3)

Given: A sequence of $3 \cdot n$ distinct positive integers.

Question: Is there a partition of the sequence into n increasing subsequences each of length three?

References

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

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

[3] K.A. Baker, P.C. Fishburn and F.S. Roberts: Partial orders of dimension 2, *Networks* 2 (1971), 11-28.

[4] M. Bartusch, R.H. Möhring and F.J. Radermacher: M-machine unit time scheduling: A report on ongoing research, in: Optimization, Parallel Processing and Applications (ed. A. Kurzhanski), LNEMS 304, 1988, 165-213.

[5] P. Bjorstad, W.M. Coughran and E. Grosse: Parallel domain decomposition applied to coupled transport equations, in: Domain Decomposition Methods in Scientific and Engineering Computing (eds. D.E. Keys, J. Xu), AMS, Providence, 1995, 369-380.

[6] J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt and J. Weglarz: Scheduling computer and manufacturing processes, Springer, Heidelberg, 1996.

[7] H.L. Bodlaender and K. Jansen: On the complexity of scheduling incompatible jobs with unit-times, *Mathematical Foundations of Computer Science*, MFCS 93, LNCS 711, 291-300.

[8] A. Brandstädt: On improved time bounds for permutation graph problems, *Graph Theoretic Concepts in Computer Science*, WG 92, LNCS 657, 1-10.

[9] E. Cohen and M. Tarsi: NP-completeness of graph decomposition problem, *Journal of Complexity* 7 (1991), 200-212.

[10] D.G. Corneil: The complexity of generalized clique packing, *Discrete Applied Mathematics* 12 (1985), 233-239.

[11] M.R. Garey and D.S. Johnson: Computers and Intractability: A Guide to the Theorey of NP-Completeness, Freeman, New York, 1979.

[12] P.C. Gilmore and A.J. Hoffman: A characterization of comparability graphs and of interval graphs, *Canadian Journal of Mathematics* 16 (1964), 539-548.

[13] P. Hansen, A. Hertz and J. Kuplinsky: Bounded vertex colorings of graphs, *Discrete Mathematics* 111 (1993), 305-312.

[14] S. Irani and V. Leung: Scheduling with conflicts, and applications to traffic signal control, *SIAM Symposium on Discrete Algorithms*, SODA 96, 85-94.

[15] D. Kaller, A. Gupta and T. Shermer: The χ_t - coloring problem, *Symposium on Theoretical Aspects of Computer Science*, STACS 95, LNCS 900, 409-420.

[16] F. Kitagawa and H. Ikeda: An existential problem of a weight - controlled subset and its application to school time table construction, *Discrete Mathematics* 72 (1988), 195-211.

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

[18] V. Lofti and S. Sarin: A graph coloring algorithm for large scale scheduling problems, *Computers and Operations Research* 13 (1986), 27-32.

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

[20] R.H. Möhring: Algorithmic aspects of comparability graphs and interval graphs, in: Graphs and Orders (ed. I. Rival), Reidel Publishing, Dordrecht 1985, 41-101.

[21] R.H. Möhring: Problem 9.10, in: Graphs and Orders (ed. I. Rival), Reidel Publishing, Dordrecht 1985, 583.

[22] R.H. Möhring: Computationally tractable classes of ordered sets, in: Algorithms and Orders (ed. I. Rival), Kluwer Acad. Publishing, Dordrecht 1989, 105-193.

[23] A. Pnueli, A. Lempel and W. Even: Transitive orientation of graphs and identification of permutation graphs, *Canadian Journal of Mathematics* 23 (1971), 160-175.

[24] K. Wagner: Monotonic coverings of finite sets, *Journal of Information Processing and Cybernetics - EIK* 20 (1984), 633 – 639.