The Optimal Cost Chromatic Partition Problem for Trees and Interval Graphs

Leo G. Kroon
Erasmus University Rotterdam
P.O. Box 1738, NL-3000 DR Rotterdam
The Netherlands

Arunabha Sen
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287
USA

Haiyong Deng
Computer Aided Design Group
Intel Corporation
Santa Clara, CA 95052
USA

Asim Roy
Department of Decision and Information Sciences
Arizona State University
Tempe, AZ 85287
USA

Abstract. In this paper we study the Optimal Cost Chromatic Partition (OCCP) problem for trees and interval graphs. The OCCP problem is the problem of coloring the nodes of a graph in such a way that adjacent nodes obtain different colors and that the total coloring costs are minimum.

In this paper we first give a linear time algorithm for the OCCP problem for trees. The OCCP problem for interval graphs is equivalent to the Fixed Interval Scheduling Problem with machine-dependent processing costs. We show that the OCCP problem for interval graphs can be solved in polynomial time if there are only two different values for the coloring costs. However, if there are at least four different values for the coloring costs, then the OCCP problem for interval graphs is shown to be NP-hard. We also give a formulation of the latter problem as an integer linear program, and prove that the corresponding coefficient matrix is perfect if and only if the associated intersection graph does not contain an odd hole of size 7 or more as a node-induced subgraph. Thereby we prove that the Strong Perfect Graph Conjecture holds for graphs of the form $K \Theta G$, where K is a clique and G is an interval graph.

1 Introduction

In this paper we study the Optimal Cost Chromatic Partition (OCCP) problem for trees and interval graphs. The general OCCP problem can be described as follows: given a graph $G = (V, E)$ with n nodes and a sequence of coloring costs $(k_1, \ldots, k_n)$, find a proper coloring $C(v) \in \{1, \ldots, n\}$ of each node $v \in V$ such that the total coloring costs $\sum_{v=1}^{n} k_{C(v)}$ are minimum. Here a coloring is proper if adjacent nodes have different colors. Thus an alternative formulation of the OCCP problem is the following: Given a graph $G = (V, E)$ with n nodes and a sequence of coloring costs $(k_1, \ldots, k_n)$, partition the vertex set V into independent sets $V_1, \ldots, V_s$ such that $\sum_{c=1}^{s} k_c |V_c|$ is minimum. Here all nodes in the independent set V_c are colored by the same color c. Without loss of generality we assume $k_c \leq k_{c'}$ whenever $c < c'$ throughout this paper.

The OCCP problem was introduced by Supowit [22] in the VLSI context. He referred to the problem as the *Weighted Coloring Problem* of a graph. However, the term *Weighted Coloring Problem* is used by Grötschel et al. [12] to describe a different problem. In order to avoid further confusion, we refer to the problem as the *Optimal Cost Chromatic Partition* problem. This problem is a generalization of the *Chromatic Sum* problem introduced by Kubicka & Schenk [17].

It is not difficult to show that the OCCP problem is NP-hard for arbitrary graphs. Supowit [22] deals with the OCCP problem for circle graphs. It follows from the results of Garey et al. [10] that this problem is NP-hard. Furthermore, Sen et al. [21] consider the OCCP problem for permutation graphs.

In this paper we study the OCCP problem for trees and interval graphs. The OCCP problem for interval graphs is equivalent to the Fixed Interval Scheduling Problem (FISP) with machine-dependent processing costs. In this scheduling problem each job j to be carried out requires processing during a fixed time interval (s_j, f_j). A sufficient number of machines is available, and all jobs may be carried out by all machines. However, the processing costs are machine-dependent. That is, if job j is carried out by machine m, then the associated processing costs are k_m. The objective is to find a feasible non-preemptive schedule for all jobs against minimum total processing costs.

The outline of this paper is as follows: In Section 2 we describe an integer linear program that can be used to solve the general OCCP problem. Next, in Section 3 we show that the OCCP problem for trees can be solved in linear time. In Section 4 we study the OCCP problem for interval graphs. We show that this problem can be solved in polynomial time if there are only two different values for the coloring costs. However, if there are at least four different values for the coloring costs, then the problem is shown to be NP-hard. We improve the given integer linear program by describing all clique inequalities, and we prove that the corresponding coefficient matrix is perfect if and only if the associated intersection graph does not contain an odd hole of size 7 or more as a node-induced subgraph. Thereby we show that the Strong Perfect Graph Conjecture holds for graphs of the form $K \times G$, where K is a clique and G is an interval graph.

2 Model formulation

In this section we give a formulation of the OCCP problem as an integer linear program. To that end, suppose we have an instance I of the OCCP problem containing a graph $G = (V, E)$ with n nodes and a sequence of coloring costs $(k_1, \ldots, k_n)$. Then the integer program to solve this instance of the OCCP problem uses the binary decision variables $x_{v,c}$ indicating whether or not node v is colored by color c $(v, c = 1, \ldots, n)$. The objective and the constraints of the model can be described as follows:

$$\min \sum_{v=1}^{n} \sum_{c=1}^{n} k_c x_{v,c} \tag{0}$$

subject to

$$\sum_{c=1}^{n} x_{v,c} = 1 \qquad v = 1, \ldots, n \tag{1}$$

$$x_{v,c} + x_{v',c} \leq 1 \qquad (v, v') \in E;\ c = 1, \ldots, n \tag{2}$$

$$x_{v,c} \in \{0, 1\} \qquad v = 1, \ldots, n;\ c = 1, \ldots, n \tag{3}$$

The objective function (0) specifies that we are interested in minimizing the total coloring costs. Constraints (1) require each node to be colored exactly once. Constraints (2) guarantee that two nodes v and v' that are connected by an edge $(v, v') \in E$ are colored by different colors. Note that these constraints could be tightened by replacing them by the corresponding *clique* constraints. Unfortunately, for arbitrary graphs the number of cliques and clique constraints may be extremely large, and it may require quite some time to find them. However, for trees and interval graphs the number of cliques is linear in the number of nodes. In fact, the only cliques of a tree are the edges. Furthermore, in Section 4.2 we will show how for an interval graph the constraints (2) can be replaced by the corresponding clique constraints. Finally, the constraints (3) declare the variables $x_{v,c}$ as binary variables.

The coefficient matrix corresponding to the restrictions (1) to (3) associated with an instance I of the OCCP problem is called $M(I)$. Note that $M(I)$ is a zero/one matrix. A zero/one matrix M is said to be *perfect* if the polyhedron

$$P(M) = \{X | M.X \leq \mathbb{1} \text{ and } X \geq 0\}$$

has only integral extremal points. Here the vector $\mathbb{1}$ is a vector containing all 1's. It follows that the problem (0) to (3) can be solved by applying a linear programming algorithm if the matrix $M(I)$ is perfect. Padberg [18] gives a complete characterisation of perfect matrices in terms of forbidden submatrices.

3 The OCCP problem for Trees

Sen et al. [21] show that the coefficient matrix $M(I)$ associated with the integer linear program (1) to (3) is perfect if the graph G is a tree. Thus the OCCP problem for trees can be solved in polynomial time by any polynomial linear programming algorithm (Grötschel et al. [12]).

However, in this section we describe a linear time algorithm for solving the OCCP problem for trees. To that end, let T be a tree with n nodes, and let $(k_1, \ldots, k_n)$ be the corresponding sequence of coloring costs. In this section we denote the degree of node v, which is the number of different neighbours of node v, by D_v. The maximum node degree over all nodes is denoted by D.

Next, we will show that for any graph G the number of colors used by a proper coloring of minimum costs may be bounded by $D+1$. This upper bound is sharp, as is demonstrated by the complete graphs (cliques).

Lemma 1. *For a graph G there exists a proper coloring C of minimum costs such that $C(v) \leq D_v + 1$ for all nodes v.*

Proof. Let C be a proper coloring of minimum costs, and suppose there is a node v with $C(v) = c > D_v + 1$. Then there exists a color c' with $c' \leq D_v + 1$ such that $c' \neq C(v')$ for all neighbour v' of v. Now we define the following alternative coloring C' of the graph G.

$$C'(v) = \begin{cases} c' & \text{if } v = n, \\ C(v) & \text{if } v \neq n. \end{cases}$$

Clearly, C' is a proper coloring of G. Furthermore, $k_{c'} \leq k_c$, since $c' < c$. Thus C' is also a coloring of minimum costs. By repeating this argument as often as necessary, we obtain a proper coloring of the graph of minimum costs satisfying the condition of the lemma. □

The algorithm for solving the OCCP problem for trees uses the result of Lemma 1 Recall that the chromatic number of a tree is 2 (or 1). On the other hand, it is not difficult to create an instance of the OCCP problem for trees for which the number of colors used in an optimal solution is arbitrarily large.

The algorithm for solving the OCCP problem for trees is based on the idea of dynamic programming. To that end, we first add a direction to the edges of T by choosing an arbitrary node as the root r, and by directing an edge $(v, v') \in E$ from v to v' if the unique path in T from root r to node v does not visit node v'. In the obtained directed tree each node v is the root of the subtree rooted at node v.

The algorithm to solve the OCCP problem for trees keeps track of the following information for each node v:

(i) $K_1(v)$, representing the minimum costs of coloring the subtree rooted at node v (this is called the *primary* coloring of the subtree rooted at node v), *(ii)* $C(v)$, representing the color used for node v in the primary coloring of the subtree rooted at node v,

(iii) $K_2(v)$, representing the minimum costs of coloring the subtree rooted at node v with node v colored differently from $C(v)$ (this is called the *secondary* coloring of the subtree rooted at node v).

Obviously, if v is a leaf node, then $K_1(v) = c_1$, $C(v) = 1$, and $K_2(v) = c_2$.

If v is a non-leaf node with d children, $v_1, \ldots, v_d$, then first the values $K_1(v_i)$, $C(v_i)$, and $K_2(v_i)$ are determined recursively for $i = 1, \ldots, d$. Next, the coloring costs $K(v, c)$ are determined for $c = 1, \ldots, d+1$ by equation (4) below. The coloring costs $K(v, c)$ denote the costs of coloring the subtree rooted at node v, when node v is colored with color c and all subtrees rooted at the children of node v are colored as cheap as possible. Thus,

$$K(v,c) := k_c + \sum_{i:C(v_i)\neq c} K_1(v_i) + \sum_{i:C(v_i)=c} K_2(v_i) \quad \text{for } c = 1, \ldots, d+1 \qquad (4)$$

Note that (4) can be calculated in an amount of time that is linear in the number of subtrees d by the following steps:

$$K := \sum_{i=1}^{d} K_1(v_i)$$

For $c := 1$ **to** $d+1$ **do** $\Delta(c) := K$

For $i := 1$ **to** d **do**

If $C(v_i) \leq d+1$ **then** $\Delta(C(v_i)) := \Delta(C(v_i)) + K_2(v_i) - K_1(v_i)$

For $c := 1$ **to** $d+1$ **do** $K(v,c) := k_c + \Delta(c)$

Given the coloring costs $K(v, c)$ for $c = 1, \ldots, d+1$, the values $K_1(v)$, $C(v)$, and $K_2(v)$ are computed using the following formulas.

$$K_1(v) := \min \{ K(v,c) \mid c = 1, \ldots, d+1 \}, \qquad (5)$$

$$C(v) := \operatorname{argmin} \{ K(v,c) \mid c = 1, \ldots, d+1 \}, \qquad (6)$$

$$K_2(v) := \min \{ K(v,c) \mid c = 1, \ldots, d+1;\ c \neq C(v) \}. \qquad (7)$$

In (5) the minimum costs of coloring the subtree rooted at node v are determined, and in (6) the corresponding color is determined. Next, in (7) the costs of a secondary coloring of the subtree rooted at node v are determined.

Obviously, the complexity of the above algorithm for coloring node v is $\mathcal{O}(d)$, where d is the number of children of node v. Therefore the overall complexity of the algorithm for computing the minimum costs of coloring the entire tree is proportional to the number of edges, which, in a tree, gives rise to an $\mathcal{O}(n)$ algorithm.

4 The OCCP problem for interval graphs

An interval graph $G = (V, E)$ is a graph where each node corresponds to a time interval (s_j, f_j), and where two nodes are connected by an edge if and only if the corresponding intervals are overlapping. Thus the OCCP problem for an interval graph G can be considered as the problem of coloring the corresponding intervals (s_j, f_j) in such a way that overlapping intervals obtain different colors, and such that the total coloring costs are minimum. The costs of coloring an interval with a certain color only depend on the color to be used.

As was mentioned already in the introduction of this paper, the OCCP problem for interval graphs is equivalent to the Fixed Interval Scheduling Problem (FISP) with machine-dependent processing costs. In this scheduling problem each job j to be carried out requires processing during a fixed time interval (s_j, f_j). A sufficient number of machines is available, and all jobs may be carried out by all machines. However, the processing costs are machine-dependent. That is, if job j is carried out by machine m, then the associated processing costs are k_m. The objective is to find a feasible non-preemptive schedule for all jobs against minimum total processing costs. It follows that the intervals and the colors of the OCCP problem for interval graphs correspond with the jobs and the machines of FISP with machine-dependent processing costs, respectively.

Several other variants of FISP have been considered in the literature (cf. Arkin & Silverberg [1], Dondeti & Emmons [4, 5], Fischetti, Martello & Toth [6, 7, 8], and Kolen & Kroon [13, 14, 15, 16]). The computational complexity of these variants of FISP has been studied extensively.

4.1 Computational complexity

In this section we prove that the OCCP problem for interval graphs is NP-hard if there are at least four different values for the coloring costs. This result is somewhat surprising, because almost every other combinatorial problem related to interval graphs, such as the computation of the chromatic number, the maximum independent set, the maximum clique, and the dominating set can be accomplished in polynomial time.

On the other hand, if the first s coloring costs are equal and the last $n - s$ coloring costs are equal as well, i.e., $k_1 = k_2 = \ldots = k_s = A_1$, and $k_{s+1} = k_{s+2} = \ldots = k_n = A_2$ (where it may be assumed, without loss of generality, that $A_1 < A_2$), then the optimal solution is obtained if the largest s-colorable subgraph is colored with the colors $1, \ldots, s$, and the other nodes are colored with the remaining colors. For interval graphs, the problem of finding the largest s-colorable subgraph can be solved in polynomial time by the greedy algorithm of Yanakakis & Gavril [25]. Note that, if there are only two different values for the coloring costs, then the OCCP problem for any graph is equivalent to the problem of finding the largest s-colorable subgraph of the graph.

Next we show that the OCCP problem for interval graphs is NP-hard if there are at least four different values for the coloring costs. The proof uses a reduction from the problem Numerical Three Dimensional Matching (N3DM), which is defined in following way:

Instance of N3DM:

- A positive integer t and $3t$ rational numbers a_i, b_i and c_i satisfying $\sum_{i=1}^{t}(a_i+b_i+c_i)=t$ and $0<a_i,b_i,c_i<1$ for $i=1,\ldots,t$.

Question:

- Do there exist permutations ρ and σ of $\{1,\ldots,t\}$ such that $a_i+b_{\rho(i)}+c_{\sigma(i)}=1$ for $i=1,\ldots,t$?

It is well-known that N3DM is NP-complete in the strong sense (Garey and Johnson [9]). Therefore any problem in NP that is more general than N3DM is NP-complete as well. The proof of Theorem 2 is illustrated with the following instance of N3DM with $t=3$: $(a_1,a_2,a_3)=(1/8,1/4,3/8)$, $(b_1,b_2,b_3)=(1/8,1/4,1/2)$ and $(c_1,c_2,c_3)=(1/4,3/8,3/4)$. This instance of N3DM is a yes-instance, since $a_1+b_1+c_3=a_2+b_3+c_1=a_3+b_2+c_2=1$.

Theorem 2. *The OCCP problem for interval graphs is NP-hard if there are at least four different values for the coloring costs.*

The idea behind the proof is that as many as possible intervals should be colored by the cheaper colors and as few as possible should be colored by the more expensive ones.

Proof. The theorem is proved by a reduction from N3DM. Hence let I_1 be an instance of N3DM containing the integer t and the rational numbers a_i, b_i and c_i for $i=1,\ldots,t$. Next, for $i,j=1,\ldots,t$ the rational numbers A_i, B_j and $X_{i,j}$ are chosen in such a way that all these numbers are different and that $4<A_i<5<B_j<6$ and $7<X_{i,j}<9$ for $i,j=1,\ldots,t$.

Based on these data, an instance I_2 of the OCCP problem for interval graphs is constructed. The intervals that have to be colored in I_2 are the following.

t times $(0,1)$,		$(11-c_k,13)$	for $k=1,\ldots,t$,
t times $(1,2)$,		$t-1$ times $(0,A_i)$	for $i=1,\ldots,t$,
t times $(13,14)$,		$t-1$ times $(3,B_j)$	for $j=1,\ldots,t$,
t^2-t times $(0,3)$,		$(A_i,X_{i,j})$	for $i,j=1,\ldots,t$,
t^2-t times $(12,14)$,		$(B_j,X_{i,j})$	for $i,j=1,\ldots,t$,
$(0,B_j)$	for $j=1,\ldots,t$,	$(X_{i,j},14)$	for $i,j=1,\ldots,t$,
$(2,A_i)$	for $i=1,\ldots,t$,	$(X_{i,j},10+a_i+b_j)$	for $i,j=1,\ldots,t$.

Furthermore, for coloring these intervals there are t different colors with costs 0, there are t^2-t different colors with costs 1, there are t^2 different colors with costs 2, and all other colors have costs $12t^2$. Note that the construction of I_2

can be carried out in a polynomial amount of time. The instance I_2 constructed from an instance I_1 of N3DM defined above is shown in Figure 1.

Figure 1. An instance of the OCCP problem for interval graphs.

Now we will prove the following statement: I_1 is a yes-instance of N3DM if and only if the minimum total coloring costs for I_2 don't exceed $11t^2 - 5t$.

Suppose the minimum total costs of coloring all intervals of I_2 don't exceed $11t^2 - 5t$. Then only the $2t^2$ cheapest colors are used, because otherwise the total costs would be at least $12t^2$. Furthermore, the overlap of the intervals during the time interval (0,10) and during the time interval (12,14) equals $2t^2$, as can be verified easily. Therefore at least $2t^2$ colors are required to color all intervals. It can be concluded that all intervals are colored by t colors with costs 0, by $t^2 - t$ colors with costs 1, and by t^2 colors with costs 2.

Since all intervals $(A_i, X_{i,j})$ and $(B_j, X_{i,j})$ are overlapping, and since the total number of these intervals ($= 2t^2$) equals the total number of used colors, each color is used for coloring exactly one of these intervals. Furthermore, the overlap of the intervals in the time interval (0,10) and in the time interval (12,14) equals the total number of used colors. Therefore, if an interval finishes either in the time interval (0,10) or in the interval (12,14), then the next interval that is colored by the same color fits seamlessly to its predecessor.

Thus an interval $(A_i, X_{i,j})$ will be colored by the same color as a sequence of intervals of the form $(0, 1)$, $(1, 2)$, $(2, A_i)$, or as one of the intervals $(0, A_i)$. An interval $(B_j, X_{i,j})$ will be colored by the same color as a sequence of intervals of the form $(0, 3)$, $(3, B_j)$, or as an interval $(0, B_j)$. Furthermore, an interval $(A_i, X_{i,j})$ or $(B_j, X_{i,j})$ will be colored by the same color as a sequence of intervals of the form $(X_{i,j}, 10 + a_i + b_j)$, $(11 - c_k, 13)$, $(13, 14)$, as a sequence of intervals of the form $(X_{i,j}, 10 + a_i + b_j)$, $(12, 14)$, or as one of the intervals $(X_{i,j}, 14)$.

From this list of sequences, it can be concluded that each color colors at least 3 intervals and at most 7. If each color with costs 0 colors 7 intervals, each color with costs 1 colors 5 intervals, and each color with costs 2 colors 3 intervals, then the total costs equal $t \times 7 \times 0 + (t^2 - t) \times 5 \times 1 + t^2 \times 3 \times 2 = 11t^2 - 5t$. Also, any other assignment of colors to sequences of intervals will lead to higher total coloring costs. Thus if the total costs don't exceed $11t^2 - 5t$, then each color with costs 0 colors a sequence of intervals of the form $(0, 1)$, $(1, 2)$, $(2, A_i)$, $(A_i, X_{i,j})$, $(X_{i,j}, 10 + a_i + b_j)$, $(11 - c_k, 13)$, $(13, 14)$, where each i and each k occur exactly once. Further, each color with costs 1 colors a sequence of intervals of the form $(0, 3)$, $(3, B_j)$, $(B_j, X_{i,j})$, $(X_{i,j}, 10+a_i+b_j)$, $(12, 14)$, where each j occurs exactly $t - 1$ times. Hence among the intervals $(X_{i,j}, 10 + a_i + b_j)$ that are colored by the colors with costs 0 each j also occurs exactly once.

The fact $\sum_{i=1}^{t}(a_i + b_i + c_i) = t$ implies that, if two consecutive intervals are colored by the same color with costs 0, then they fit seamlessly after another. It can be concluded that, if an interval $(X_{i,j}, 10 + a_i + b_j)$ is colored by the same color with costs 0 as an interval $(11 - c_k, 13)$, then $10 + a_i + b_j = 11 - c_k$. This means that $a_i + b_j + c_k = 1$. So if we define $\rho(i) = j$ and $\sigma(i) = k$ whenever interval $(X_{i,j}, 10 + a_i + b_j)$ is colored by the same color with costs 0 as interval $(11 - c_k, 13)$, then ρ and σ are the required permutations for I_1. It can be concluded that I_1 is a yes-instance of N3DM.

Conversely, given a feasible solution for I_1, the construction can be reversed to find a feasible coloring for all intervals of I_2 with total costs $11t^2 - 5t$. As N3DM is NP-complete, the OCCP problem for interval graphs is NP-hard. $\square$

In the above proof it is assumed that the four different values for the coloring costs are 0, 1, 2, or $12t^2$. However, the statement "I_1 is a yes-instance of N3DM if and only if the minimum total costs of all intervals of I_2 don't exceed $11t^2 - 5t$" also holds if the coloring costs are 0, 1, 2, and 3. However, in that case it takes somewhat more effort to see that in a coloring with minimum total costs no intervals are colored by colors with costs 3.

As was noted already, if there are only two different values for the coloring costs, then the OCCP problem for interval graphs can be solved in polynomial time. Hence an open question still to be answered asks for the computational complexity of the OCCP problem for interval graphs, if there are exactly three different values for the coloring costs. We conjecture this problem to be NP-hard as well. This is a subject for further research.

4.2 Improved model formulation

In this section we improve the integer linear program (0) to (3) by replacing constraints (2) by the corresponding *clique* constraints. To that end, let I be an instance of the OCCP problem for interval graphs containing n intervals (s_j, f_j) and a sequence of coloring costs $(k_1, k_2, \ldots, k_n)$. Furthermore, suppose the set $\{t_r | r = 0, \ldots, R\}$ contains the start times of the intervals in chronological order. That is, $\{t_r | r = 0, \ldots, R\} = \{s_j | j = 1, \ldots, n\}$ and $t_{r-1} < t_r$ for $r = 1, \ldots, R$. Then constraints (2) may be replaced by the following constraints $(2')$:

$$\sum_{\{j | s_j \leq t_r < f_j\}} x_{j,c} \leq 1 \qquad c = 1, \ldots, n;\ r = 0, \ldots, R. \tag{2'}$$

Constraints $(2')$ state that at most one of the intervals overlapping the start time t_r of a certain interval can be colored with color c. Note that, if two intervals are overlapping, then either the first interval overlaps the start time of the second interval, or vice versa. Thus constraints $(2')$ guarantee that overlapping intervals are colored by different colors. Note that the number of constraints $(2')$ does not exceed the number of intervals n.

In the following section necessary and sufficient conditions for the coefficient matrix $M'(I)$ of the integer program (0), (1), $(2')$, (3) to be perfect are expressed in terms of the associated *intersection* graph $G(I)$. The intersection graph of a zero/one matrix M is a graph G containing exactly one node for each column of the matrix M. Two nodes x and y of G are connected if and only if $M_{r,x}.M_{r,y} = 1$ for some row r of the matrix M.

Note that the nodeset of $G(I)$ is the set $\{v_{j,c} | j, c = 1, \ldots, n\}$. Two different nodes $v_{j,c}$ and $v_{j',c'}$ are connected if $j = j'$ and $c \neq c'$, or if $c = c'$ and $(s_j, f_j) \cap (s_{j'}, f_{j'}) \neq \emptyset$. An edge connecting two nodes $v_{j,c}$ and $v_{j',c'}$ with $j = j'$ and $c \neq c'$ is called a *bridge edge*. An edge connecting the nodes $v_{j,c}$ and $v_{j',c'}$ with $c = c'$ and $(s_j, f_j) \cap (s_{j'}, f_{j'}) \neq \emptyset$ is called an *overlap edge*.

Note that for $c = 1, \ldots, n$ the subgraph $G_c(I)$ induced by the set of nodes $\{v_{j,c} | j = 1, \ldots, n\}$ is equivalent to the interval graph G corresponding to the intervals (s_j, f_j). The subgraphs $G_c(I)$ are coupled by the bridge edges. It follows that $G(I)$ has the form $K \times G$, where K is a clique and G is an interval graph.

Lemma 3. *If I is an instance of the OCCP problem for interval graphs, then $M'(I)$ is a clique matrix of $G(I)$.*

Proof. It follows from the definition of $G(I)$ that each clique of $G(I)$ contains either bridge edges or overlap edges, but not edges of both types.

If K is a maximum clique of $G(I)$ containing only bridge edges, then there exists an interval j such that the nodeset of K is the set $\{v_{j,c} | c = 1, \ldots, n\}$. Thus K is represented in $M'(I)$ by the row corresponding to the constraint $\sum_{c=1}^{n} x_{j,c} = 1$.

Next, let K be a maximum clique of $G(I)$ containing only overlap edges. Suppose K is a subgraph of $G_c(I)$ corresponding to color c, and let the nodeset of K be denoted by $\{v_{j,c} | j \in S\}$. Then the intervals p and q are defined as follows: $s_p = \max\{s_j | j \in S\}$ and $f_q = \min\{f_j | j \in S\}$. Now, by definition, $(s_p, f_p) \cap (s_q, f_q) \neq \emptyset$, which implies $s_p < f_q$. Note that this is trivially true if $p = q$. The definition of p and q implies that the time interval (s_p, f_q) is a subset of all intervals in the set S. It follows that $S = \{j | s_j \leq s_p < f_j\}$. Thus K is represented in $M'(I)$ by the row corresponding to the constraint $\sum_{\{j | s_j \leq s_p < f_j\}} x_{j,c} \leq 1$ in $(2')$. This completes the proof of Lemma 3. □

4.3 Strong Perfect Graph Conjecture

In this section necessary and sufficient conditions for the matrix $M'(I)$ to be perfect are expressed in terms of the associated intersection graph $G(I)$. To that end, an undirected graph $G = (V, E)$ is said to be *perfect* if for each subset $V' \subset V$ the node-induced subgraph $G' = (V', E')$ has the property

$$\alpha(G') \times \omega(G') \geq |V'|, \tag{8}$$

where $\alpha(G')$ denotes the size of a maximum independent set in G' and where $\omega(G')$ denotes the size of a maximum clique in G'. If a graph G is not perfect, then it contains a node-induced *p-critical* subgraph G'. A p-critical graph is not perfect, but all of its node-induced subgraphs are perfect.

It is well known that interval graphs are perfect. Typical examples of non-perfect graphs are the *odd holes* and the *odd antiholes* of size 5 or more. A hole is a graph with nodes $1, \ldots, n$ and edges $(v, v+1)$ for $v = 1, \ldots, n-1$ as well as the edge $(n, 1)$. An antihole is the complement of a hole.

If G is an odd hole of size 5 or more, then $\alpha(G) = (|V|-1)/2$ and $\omega(G) = 2$. Similarly, if G is an odd antihole of size 5 or more, then $\alpha(G) = 2$ and $\omega(G) = (|V|-1)/2$. Thus if G is an odd hole or an odd antihole of size 5 or more, then inequality (8) is not satisfied for G itself. It follows that any graph that contains an odd hole or an odd antihole of size 5 or more as a node-induced subgraph is not perfect. The converse of this statement is called the *Strong Perfect Graph Conjecture* (Berge [2]).

Strong Perfect Graph Conjecture. *If a graph G is not perfect, then it contains an odd hole or an odd antihole of size 5 or more as a node-induced subgraph.*

Another formulation of the Strong Perfect Graph Conjecture states that the odd holes and the odd antiholes of size 5 or more are the only p-critical graphs.

For several classes of graphs the Strong Perfect Graph Conjecture has been validated. For example, it holds for *toroidal* graphs (Grinstead [11]), for *claw-free* graphs (Parthasarathy & Ravindra [19]), for *planar* graphs (Tucker [23]), and for *K_4-free* graphs (Tucker [24]). However, a general proof of the Strong Perfect Graph Conjecture is still awaited at this moment.

In Theorem 4 we will show that the Strong Perfect Graph Conjecture also holds for graphs $G(I)$ where I is an instance of the OCCP problem for interval graphs. In other words, the Strong Perfect Graph Conjecture holds for graphs of the form $K \times G$, where K is a clique and G is an interval graph.

Theorem 4. *If I is an instance of the OCCP problem for interval graphs, then $G(I)$ is perfect if and only if $G(I)$ does not contain an odd hole of size 5 or more as a node-induced subgraph.*

Proof. Let I be an instance of the OCCP problem for interval graphs, and let $G(I)$ be the associated intersection graph. Suppose $G(I)$ is not perfect. Then $G(I)$ contains a node-induced p-critical subgraph G'. It is well known that each node of a p-critical subgraph G' is contained in exactly $\omega(G')$ maximal cliques of G', if $\omega(G')$ denotes the size of a maximum clique of G' (Padberg [18]).

As for $c = 1, \ldots, n$ the interval graph $G_c(I)$ is perfect, G' contains at least one bridge edge of $G(I)$, say between $G_1(I)$ and $G_2(I)$. Thus the subgraphs $G' \cap G_1(I)$ and $G' \cap G_2(I)$ are not empty. Now we consider the interval p defined by $f_p = \min\{f_j | n_{j,1} \in G'\}$. According to the definition of interval p, all intervals j with $v_{j,1} \in G'$ that are overlapping interval p are also overlapping each other. It follows that all nodes in the set $\{v_{j,1} \in G' | (s_j, f_j) \cap (s_p, f_p) \neq \emptyset\}$ make up a maximum clique of G'. This clique contains only overlap edges. Note that $v_{p,1}$ is an element of this clique. Node $v_{p,1}$ is contained in at most 1 other maximum clique, namely a clique containing only bridge edges. Thus node $v_{p,1}$ is contained in at most 2 maximum cliques. As G' is p-critical, each node is contained in at least 2 maximum cliques. It follows that $v_{p,1}$ is contained in exactly 2 maximum cliques. As a consequence, $\omega(G') = 2$. The latter implies that G' is an odd hole of size 5 or more. □

The result of Theorem 4 can be improved slightly. Indeed, if the node-induced subgraph G' of $G(I)$ would be an odd hole of size 5, then G' would contain exactly 2 bridge edges and exactly 3 overlap edges. (Indeed, recall that G' can not be a subgraph of one of the graphs $G_c(I)$, since the latter are perfect). Let the corresponding intervals be called 1, 2 and 3, and let the corresponding colors be called 1 and 2. Then the nodes of G' are indicated by $v_{1,1}$, $v_{2,1}$, $v_{1,2}$, $v_{2,2}$ and $v_{3,2}$. Since there is an edge between the nodes $v_{1,1}$ and $v_{2,1}$, the intervals 1 and 2 are overlapping. But then there is also an edge between the nodes $v_{1,2}$ and $v_{2,2}$. This implies that G' contains a chord and, as a consequence, it is not an odd hole. Thus the following corollaries have been obtained.

Corollary 5. *If I is an instance of the OCCP problem for interval graphs, then $G(I)$ is perfect if and only if it does not contain an odd hole of size 7 or more as a node-induced subgraph.*

Corollary 6. *If I is an instance of the OCCP problem for interval graphs, then $M'(I)$ is perfect if and only if $G(I)$ does not contain an odd hole of size 7 or more as a node-induced subgraph.*

Proof. This result is a consequence of the following well-known result: if the zero/one matrix M is a clique matrix of its associated intersection graph G, then M is perfect if and only if G is perfect (Chvatal [3]). □

5 Final Remarks

In this paper we have studied the Optimal Cost Chromatic Partition problem for trees and interval graphs. First, we have given an algorithm based on the idea of dynamic programming for solving the OCCP problem for trees which runs in linear time.

Furthermore, we have shown that the OCCP problem for chordal graphs can be solved in polynomial time if there are only two different values for the coloring costs. This result is also valid for interval graphs, since each interval graph is a chordal graph. However, if there are at least four different values for the coloring costs, then the OCCP problem for interval graphs is NP-hard.

We have presented a formulation of the problem as an integer program with coefficient matrix $M'(I)$. The matrix $M'(I)$ is shown to be a clique matrix of its associated intersection graph $G(I)$. This implies that perfectness of the matrix $M'(I)$ is equivalent to perfectness of the graph $G(I)$. Thus, if $G(I)$ is perfect, then the problem can be solved by linear programming. We have shown that $G(I)$ is perfect if and only if it does not contain an odd hole of size 7 or more as a node induced subgraph. Thereby we have proved that the Strong Perfect Graph Conjecture is valid for graphs of the form $K \times G$, where K is a clique and G is an interval graph.

References

1. E.M. Arkin, and E.L. Silverberg. Scheduling jobs with fixed start and finish times. *Discrete Applied Mathematics*, 18 (1987), 1–8.
2. C. Berge. Färbung von Graphen deren sämtliche bzw. deren ungerade Kreise starr sind. *Wiss. Z. Martin-Luther Univ. Halle-Wittenberg.* Math.-Natur. Reihe, 114 (1961).
3. V. Chvatal. On certain polytopes associated with graphs. *J. Comb. Theory*, B-18 (1975) 138–154.
4. V.R. Dondeti, and H. Emmons. Job scheduling with processors of two types. *Operations Research*, 40 (1992) S76–S85.
5. V.R. Dondeti, and H. Emmons. Algorithms for preemptive scheduling of different classes of processors to do jobs with fixed times. *European Journal of Operational Research*, 70 (1993) 316–326.
6. M. Fischetti, S. Martello, and P. Toth. The Fixed Job Schedule Problem with spread time constraints. *Operations Research*, 6 (1987) 849–858.
7. M. Fischetti, S. Martello, and P. Toth. The Fixed Job Schedule Problem with working time constraints. *Operations Research*, 3 (1989) 395–403.
8. M. Fischetti, S. Martello and P. Toth. Approximation algorithms for Fixed Job Schedule Problems. *Operations Research*, 40 (1992) S96–S108.
9. M.R. Garey, and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness.* Freeman, San Fransisco, 1979.
10. M.R. Garey, D.S. Johnson, G.L. Miller and C.H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal of Alg. Disc. Meth*, 1 (1980) 216-227.
11. C. Grinstead. The perfect graph conjecture for toroidal graphs. *Annals of Discrete Mathematics*, 21 (1984) 97–101.
12. M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, (1988). Springer Verlag.
13. A.W.J. Kolen, and L.G. Kroon. On the computational complexity of (maximum) class scheduling. *European Journal of Operational Research*, 54 (1991), 23–38.
14. A.W.J. Kolen, and L.G. Kroon. Licence Class Design: complexity and algorithms. *European Journal of Operational Research*, 63 (1992), 432–444.
15. A.W.J. Kolen, and L.G. Kroon. On the computational complexity of (maximum) shift class scheduling. *European Journal of Operational Research*, 64 (1993), 138–151.
16. A.W.J. Kolen, and L.G. Kroon. Analysis of shift class design problems. *European Journal of Operational Research*, 79 (1994), 417–430.
17. E. Kubicka, and A. Schwenk. Introduction to chromatic sums. *Proc. ACM Computer Science Conference*, (1989).
18. M.W. Padberg. Perfect Zero-One matrices. *Mathematical Programming*, 6 (1974) 180–196.

19. K.R. Parthasarathy, and G. Ravindra. The Strong Perfect Graph Conjecture is true for $K_{1,3}$-free graphs. *J. Comb. Theory*, B-21 (1976) 212–223.
20. A. Sen, H. Deng, and S. Guha. On a graph partition problem with an application to VLSI layout. *Information Processing Letters*, 43 (1991) 87–94.
21. A. Sen, H. Deng, and A. Roy. On a graph partition problem with application to multiprocessor scheduling. TR-92-020, Department of Computer Science and Engineering, Arizona State University.
22. K.J. Supowit. Finding a maximum planar subset of a set of nets in a channel. *IEEE Trans. on Computer Aided Design*, CAD 6, 1 (1987) 93–94.
23. A. Tucker. The Strong Perfect Graph Conjecture for planar graphs. *Canad. J. Math*, 25 (1973) 103–114.
24. A. Tucker. The validity of the perfect graph conjecture for K_4-free graphs. *Annals of Discrete Mathematics*, 21 (1984) 149–157.
25. M. Yanakakis, and F. Gavril. The maximum k-colorable subgraph problem for chordal graphs. *Information Processing Letters*, 24 (1987), 133–137.