

 Open access • Journal Article • DOI:10.1007/S10951-006-0003-7

Time slot scheduling of compatible jobs — Source link

Marc Demange, D. de Werra, Jérôme Monnot, V. Th. Paschos

Institutions: ESSEC Business School, École Polytechnique Fédérale de Lausanne, Paris Dauphine University

Published on: 01 Apr 2007 - Journal of Scheduling (Kluwer Academic Publishers-Plenum Publishers)

Topics: Bipartite graph, Edge coloring and Open-shop scheduling

Related papers:

- [Weighted coloring: further complexity and approximability results](#)
- [Batch processing with interval graph compatibilities between tasks](#)
- [Approximation algorithms for the max-coloring problem](#)
- [Weighted coloring on planar, bipartite and split graphs: Complexity and approximation](#)
- [Buffer minimization using max-coloring](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/time-slot-scheduling-of-compatible-jobs-16zwxm61m3>

CAHIER DU LAMSADE
182

Septembre 2001

Time slot scheduling of compatible jobs

M. Demange, D. de Werra, J. Monnot, V. Th. Paschos

Table of Contents

Résumé	ii
Abstract	ii
1 Introduction	1
2 General properties	2
3 Approximating weighted coloring in general graphs	3
3.1 Standard approximation	3
3.2 Differential approximation	5
3.3 A general negative result	7
4 The bipartite case and some related cases	7
4.1 The complexity of time slot scheduling	7
4.2 On the approximability of time slot scheduling in bipartite graphs	11
4.2.1 Standard approximation	11
4.2.2 Differential approximation	12
4.3 The split graphs	13
5 An edge coloring model	15
5.1 Complexity results	15
5.2 On the approximation of the ETSS in bipartite graphs	19
5.2.1 Standard approximation	19
5.2.2 Differential approximation	21
6 Cographs	22
References	24

Ordonnement des tâches compatibles avec fenêtres de temps

Résumé

Nous introduisons une version pondérée de la coloration des graphes motivée par une classe de problèmes d'ordonnement : chaque nœud v d'un graphe G est associé à une tâche (d'une durée $w(v)$) à exécuter. Les arêtes représentent des contraintes de non-simultanéité (incompatibilités). Il s'agit alors d'associer chaque tâche à une fenêtre de temps de sorte que les opérations assignées à chaque fenêtre soient compatibles entre elles. La longueur d'une fenêtre est la durée maximale des tâches qui lui sont associées. Le nombre k de fenêtres de temps utilisées doit être déterminé au mieux. Il s'agit alors de trouver une k -coloration $S = (S_1, \dots, S_k)$ de G minimisant la somme $w(S_1) + \dots + w(S_k)$ où $w(S_i) = \max\{w(v) : v \in S_i\}$. Nous étudions les propriétés des solutions optimales et présentons des résultats de complexité et d'approximation. Certains de ces résultats mettent en jeu des méthodes heuristiques. Nous montrons que le problème de décision associé est **NP**-complet, même dans le cas des graphes bipartis, de leurs graphes aux arêtes ou encore pour les graphes *split*.

Mots-clé : Coloration pondérée; Ordonnement chromatique; Approximations; Coloration d'arêtes; Ordonnement par lot.

Time slot scheduling of compatible jobs

Abstract

A version of weighted coloring of a graph is introduced which is motivated by some types of scheduling problem: each node v of a graph G corresponds to some operation to be processed (with a processing time $w(v)$), edges represent non-simultaneity requirements (incompatibilities). We have to assign each operation to one time slot in such a way that in each time slot all operations assigned to this slot are compatible; the length of a time slot will be the maximum of the processing times of its operations. The number k of time slots to be used has to be determined as well. So we have to find a k -coloring $S = (S_1, \dots, S_k)$ of G such that $w(S_1) + \dots + w(S_k)$ is minimized where $w(S_i) = \max\{w(v) : v \in S_i\}$. Properties of optimal solutions are discussed, complexity and approximability results are presented. Heuristic methods are given for establishing some of these results. The associated decision problems are shown to be **NP**-complete for bipartite graphs and for line-graphs of bipartite graphs and for split graphs.

Keywords: Weighted coloring; Chromatic scheduling; Approximations; Edge coloring; Batch scheduling.

1 Introduction

In this paper we shall consider some variations of a chromatic scheduling problem which can be expressed as a kind of weighted coloring problem. More precisely the nodes v of a graph $G = (V, E)$ will have non-negative weights $w(v)$. Then, considering that these weights correspond to processing times of the operations (jobs) associated to the nodes, we will try to find schedules with a minimum total completion time.

As usual we will have to take into account some incompatibility constraints between the jobs; these will be represented by the edges of a graph G .

A k -coloring of $G = (V, E)$ is a partition $\mathcal{S} = (S_1, \dots, S_k)$ of the node set V of G into stable sets S_i . In terms of scheduling, it amounts to assigning each job v to a time-slot (or period) i in such a way that no two jobs u, v assigned to the same time slot i are incompatible; in other words $u, v \in S_i$ implies that u and v are not incompatible, i.e., S_i is a stable set for each i . In our situation, the lengths of the time slots $1, 2, \dots, k$ are not given in advance; assuming that the jobs scheduled in time slot i may be processed simultaneously, the amount of time needed will be given by $w(S_i) = \max\{w(v) : v \in S_i\}$. As a consequence, the total amount of time needed to complete all jobs will be:

$$C(\mathcal{S}) = \sum_{i=1}^k w(S_i) \quad (1)$$

where $\mathcal{S} = (S_1, \dots, S_k)$. The problem then amounts to finding for a weighted graph $G_w = (V, E, w)$ (each node v has a positive weight $w(v)$) a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ such that $C(\mathcal{S})$ defined by (1) is minimum.

As we shall see the number k of time slots is not always given in advance; it may turn out that if $\chi(G)$ is the chromatic number of G , we may have to use $k > \chi(G)$ time slots. So we shall admit in our formulation that some of the subsets S_i be possibly empty. We shall furthermore assume that all weights are positive integer numbers.

After establishing approximation results for the weighted coloring problem in general graphs, we examine some special cases of the scheduling problem introduced above which we call *Time Slot Scheduling* (or TSS); complexity issues as well as approximability will be discussed. Heuristic methods will be given for deriving some of these results.

Our problem TSS is related to the *batch scheduling* problem which has been studied by several authors (see for instance Potts and Kovalyov ([25]) for a survey, or Potts et al. ([26]) for a special case). In the papers on batch scheduling, there are usually incompatibility constraints between operations belonging to a same job, or precedence constraints. The general case of incompatibility requirements represented by an arbitrary graph is formulated in Boudhar and Finke ([7]), where they consider the complement of our graph: edges indicate compatibilities and they partition the node set into cliques. On the other hand, several types of requirements are introduced, like sequencing constraints or limitations in the size of a batch. Our approach is characterized by the underlying graph theoretical formulation. We shall therefore prefer the term *time slot scheduling* to specify the weighted graph coloring approach which will be used in the paper.

In Bodlaender et al. ([5]) incompatibilities between operations are defined in a different way; two operations are called *incompatible* if they cannot be assigned to the same processor, while in our model two *incompatible* operations cannot be assigned to the same time slot (but they could be processed on the same machine). In Bodlaender et al. ([5]) the k -coloring gives the assignment of operations to k machines, while in our model it gives an assignment of operations to k time slots. The graphs are not the same; they are not even complementary, since two jobs which are to be processed in different time slots (incompatible in our sense) may or may not be processed

on the same machine. In the formulation of Bodlaender et al. ([5]) the objective is to determine a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ such that $\max_{i \leq k} w(S_i)$ is minimum where $w(S_i) = \sum_{v \in S_i} w(v)$.

Finally in the special case where the graph G is the line graph $L(H)$ of a complete bipartite graph $H = K_{n,n}$ we have a known problem of decomposition of traffic matrices which arises in satellite communication (see Burkard ([8]) and Rendl ([27]) for more references). In the case where $G = L(H)$ is the line graph of an arbitrary multigraph H and where all weights are equal, the problem $\text{TSS}(L(H))$ amounts to finding the *chromatic index* of H , i.e., the smallest number of edges in an edge coloring of H ; this problem is difficult (see Holyer ([18])). The problem of Bodlaender et al. ([5]) consists in finding an edge k -coloring (M_1, \dots, M_k) (with k given) such that $\max\{|M_i| \mid i = 1, \dots, k\}$ is minimum. It is known that such a k -coloring (with $-1 \leq |M_i| - |M_j| \leq 1$ for $i, j \leq k$) can be found in polynomial time (see for instance de Werra ([11])).

There are other types of weighted coloring problems in the literature: see Gerke ([17]), de Werra and Gay ([14]) for applications to frequency assignment and Tuza ([29]) for a survey of coloring with constraints. One should in particular mention the case of the so called *sum coloring problem*; in that situation the weight assigned to a node is not given but it depends on the color assigned to the node. More precisely, one wants to determine a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ with a minimum cost $C(\mathcal{S}) = \sum_{i=1}^k w(S_i)$ where $w(S_i) = i|S_i|$; this amounts to defining $w(S_i) = \sum_{v \in S_i} w(v)$ where $w(v) = i$ if node $v \in S_i$. Properties of these colorings together with complexity results can be found for instance in Nicoloso et al. ([24]), Bar-Noy et al. ([2]) and Jansen ([19]). For an introduction to complexity with an orientation towards scheduling, we refer the reader to Blazewicz et al. ([4]). For graph theoretical terms not defined here, the reader is referred to Berge ([3]).

2 General properties

Before examining the variations on TSS, we shall give a general property which will be needed later. We recall the reader that a subset $S \subseteq V$ is *maximum* with respect to a property P if its cardinality is maximum among all subsets which have property P . We shall say that S is *maximal* with respect to P if S has property P but any subset of V which properly contains S does not have property P . A maximum subset S is maximal but the converse is not true. Now consider an instance of TSS given by a weighted graph $G = (V, E, w)$; we want to find a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ of (the nodes of) G such that $C(\mathcal{S})$ defined by (1) is minimum. Now k is not given in advance; we may choose $k = |V|$ and accept to have some subsets $S_i = \emptyset$. For practical purposes it will be convenient to use the following.

Proposition 1. *Given a weighted graph $G = (V, E, w)$, there exists a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ with $C(\mathcal{S}) = \min\{C(\tau) : \tau \text{ is a } |V|\text{-coloring of } G\}$ where for $i = 1, \dots, k$ we have:*

- (i) $S_i \neq \emptyset$;
- (ii) S_i is a maximal stable set in $G \setminus S_1 \setminus \dots \setminus S_{i-1}$;
- (iii) $w(S_1) \geq w(S_2) \geq \dots \geq w(S_k)$;

Proof. Assume that we have obtained a p -coloring $\mathcal{S} = (S_1, \dots, S_p)$ which minimize $C(\mathcal{S})$ over all possible colorings of G . By reordering the subsets S_i and removing the empty sets we get a k -coloring \mathcal{S} satisfying (i) and (iii). If S_1 is not maximal we may find in some S_l ($l \geq 2$) some elements which can be added to S_1 in order to make it maximal; this will not increase $w(S_1)$ because (iii) holds. We may repeat this for S_2, \dots, S_{k-1} and obtain the required k -coloring. ■

Let us denote by $\Delta(G)$ the maximum degree of the nodes in G .

Corollary 1. *For a weighted graph $G = (V, E, w)$, there exists always a k -coloring \mathcal{S} which minimizes $C(\mathcal{S})$ and which satisfies $k \leq \Delta(G) + 1$*

Proof. Assume that we have a k -coloring \mathcal{S} which minimizes $C(\mathcal{S})$ and where each S_i is maximal in $G \setminus S_1 \setminus \dots \setminus S_{i-1}$. Suppose $S_l \neq \emptyset$ (with $l > \Delta(G) + 1$). Then let $v \in S_l$; since v has at most $\Delta(G)$ neighbors in G there is at least one stable set S_r ($r \leq \Delta(G) + 1$) containing no neighbor of v ; hence $S_r \cup \{v\}$ is a stable set of $G \setminus S_1 \setminus \dots \setminus S_{r-1}$; this contradicts the fact that S_r was maximal in this graph. ■

Corollary 2. *If a weighted graph $G = (V, E, w)$ is the line-graph $L(G)$ of a multigraph H , then there exists always a k -coloring \mathcal{S} of the nodes of G which minimizes $C(\mathcal{S})$ and satisfies $k \leq 2\Delta(H) - 1$*

Proof. Each edge $e = [u, v]$ of H corresponds to a node e of $L(H)$ which belongs to two maximal cliques K_u, K_v whose nodes correspond to the edges of H adjacent to u (for K_u) and to v (for K_v).

Since $\max\{|K_u|, |K_v|\} \leq \Delta(H)$, we have: $d_{L(H)}(e) = (|K_u| - 1) + (|K_v| - 1) \leq 2\Delta(H) - 2 = \Delta(L(H))$. The result follows from Corollary 1. ■

Notice that if $G = L(H)$ for some H , then the node coloring problem in G is equivalent to an edge coloring problem in H where the weights are assigned to the edges of H . The stable sets S_i in G become matchings M_i in H . We shall consider later the case where H is bipartite.

Remark 1. We can easily show that in Corollary 1 we have $k \leq p(\omega(G) - 1) + 1$ where $\omega(G)$ is the maximum cardinality of a clique in G and p is the maximum number of (maximal) cliques in which one node of G is contained. If G is a line-graph $L(H)$, then $p = 2$ and $\omega(G) = \Delta(G)$, so Corollary 2 follows. ■

Remark 2. In fact, it follows from Proposition 1 that the number k of colors in an optimal k -coloring $C(\mathcal{S})$ can be bounded above by any bound on the chromatic number which is derived by a sequential coloring algorithm which gives maximal stable sets in the subgraph generated by the colored nodes. In particular the bounds of Welsh-Powell and of Matula are valid for k (see for instance de Werra ([12])). ■

3 Approximating weighted coloring in general graphs

3.1 Standard approximation

In this section, we establish approximability results for the weighted coloring problem defined in section 1. We use two approximation-quality criteria called in what follows *standard approximation ratio* and *differential approximation ratio*, respectively. Consider an instance I of an **NP**-hard optimization problem Π and a polynomial time approximation algorithm **A** solving Π ; we will denote by $\text{worst}(I)$, $\text{val}_A(I)$ and $\text{opt}(I)$ the values of the worst solution of I , of the approximated one (provided by **A** when running on I), and the optimal one for I , respectively. If Π is a maximization (resp., minimization) problem, the value $\text{worst}(I)$ is in fact the optimal solution of a maximization problem Π' having the same objective function and the same constraint set as Π . Let us note that computation of the solution realizing $\text{worst}(I)$ can be easy for some **NP**-hard problems (this is the case of graph coloring) but for other ones (for example, for traveling salesman, or for optimum satisfiability, or for minimum maximal independent set) this computation is **NP**-hard. Commonly, the quality of an approximation algorithm for Π is expressed by the ratio (called standard in what follows) $\rho_A(I) = \text{val}_A(I)/\text{opt}(I)$, and the quantity $\rho_A = \inf\{r : \rho_A(I) < r, I \text{ instance of } \Pi\}$ (resp.,

$\rho_A = \sup\{r : \rho_A(I) > r, I \text{ instance of } \Pi\}$, if Π is a maximization problem) constitutes the approximation ratio of A for Π . On the other hand, the differential approximation ratio measures how the value of an approximate solution is placed in the interval between $\text{worst}(I)$ and $\text{opt}(I)$. More formally, it is defined as $\delta_A(I) = |\text{worst}(I) - \text{val}_A(I)|/|\text{worst}(I) - \text{opt}(I)|$. The quantity $\delta_A = \sup\{r : \delta_A(I) > r, I \text{ instance of } \Pi\}$ is the differential approximation ratio of A for Π . A very optimistic configuration for both standard and differential approximations is the one where an algorithm achieves ratios bounded below by $1 - \epsilon$ ($1 + \epsilon$ for the standard approximation for minimization problems), for any $\epsilon > 0$. We call such algorithms *polynomial time approximation schemes*. The complexities of such schemes may be polynomial or exponential in $1/\epsilon$ (they are always polynomial in the sizes of the instances). A polynomial time approximation scheme with complexity also polynomial in $1/\epsilon$ is called *fully polynomial time approximation scheme*.

The approximation result presented in this section is based upon the so-called *master-slave approximation strategy*. Consider an **NP**-hard minimization covering graph-problem consisting in covering the nodes of the input graph G , of order n , by subgraphs G' verifying a certain property π . Most of these problems can be approximated by the following strategy:

- (a) find a maximum subgraph G' of G verifying π ;
- (b) delete $V(G')$ from V ;
- (c) repeat steps (a) and (b) in the remaining graph until $V = \emptyset$.

The maximization problem solved at step (a) is called the *slave*, while the original minimization problem is called the *master*. These terms are due to Simon ([28]) who points out the fact that if the slave problem is polynomial then the master problem is approximable within $O(\log n)$. A classical example of master-slave approximation is given by Johnson (in [21], algorithm D3 at the end of section 7 devoted to graph coloring). At each iteration this algorithm computes a maximum stable set of the remaining graph, it colors its nodes with a new color and removes them from the graph. The master problem in this case is the minimum graph coloring, while the slave problem is the maximum stable set. The following result will be used in the sequel.

Proposition 2. ([1]) *In the master-slave approximation game for weighted problems, if the weighted slave problem is approximable within ratio ρ , then the weighted master problem is approximable within ratio $\log n/\rho$.*

For our problem, the (maximization) slave problem, denoted by `SLAVE_WC`, consists of *determining a stable S^* maximizing quantity $|S|/w(S)$, over any stable set S , where $w(S) = \max\{w(v) : v \in S\}$* . Consequently, the overall algorithm `W_COLOR` we devise for weighted coloring can be outlined as follows:

- (1) solve `SLAVE_WC` in G ; let \hat{S} be the solution obtained; set $V = V \setminus \hat{S}$, $G = G[V]$;
- (2) color the nodes of \hat{S} with a new color;
- (3) repeat steps (1) and (2) until all the nodes of the input graph are colored.

Lemma 1. *`SLAVE_WC` is approximable within $O(\log^2 n/n)$ in polynomial time.*

Proof. We show in what follows that `SLAVE_WC` is equi-approximable with the maximum stable set problem. Consider the following algorithm, called `SLAVE_WC` in the sequel:

1. rank the nodes of V in nonincreasing weight-order; let L the list obtained;
2. for any $v \in L$ do:

- (a) set $V_v = \{u \in L : w(u) > w(v)\}$, $V = V \setminus (V_v \cup (\{v\} \cup \Gamma(v)))$, $G = G[V]$;
- (b) run the maximum stable algorithm of [6] on G ; let S_v be the stable set computed; store set $S^v = S_v \cup \{v\}$ as candidate= solution for SLAVE_WC;
- (c) return to the original graph G ;

3. among the sets stored in step (2b), choose one, denoted by \hat{S} , maximizing quantity $|S^v|/w(v)$.

It is easy to see that since the algorithm of [6] runs in polynomial time, algorithm SLAVE_WC does so. On the other hand, for any node v , S^v is indeed a stable set containing v ; moreover, since all the nodes heavier than v have been removed from G , $w(S^v) = w(v)$.

Denote by S^* an optimal solution of SLAVE_WC in G , and set $v^* = \operatorname{argmax}\{w(v) : v \in S^*\}$. If there exist more than one v^* , fix one of them. Of course v^* has been considered in step 2 of algorithm SLAVE_WC and set S^{v^*} has been computed and stored in step (2b). Consequently, (recall that $w(S^*) = w(v^*)$)

$$\frac{|\hat{S}|}{w(\hat{S})} \geq \frac{|S^{v^*}|}{w(v^*)} = \frac{|S^{v^*}|}{w(S^*)} \quad (2)$$

On the other hand, when S^{v^*} has been computed, the whole set S^* was present in the current graph. Denote it by G^* , denote by $S(G^*)$ a maximum stable of G^* and by ρ_S the approximation ratio of a maximum stable set algorithm. Then,

$$\rho_S = \frac{|S^{v^*}|}{|S(G^*)|} \leq \frac{|S^{v^*}|}{|S^*|} \quad (3)$$

Combining (2) and (3), we get

$$\frac{\frac{|\hat{S}|}{w(\hat{S})}}{\frac{|S^*|}{w(S^*)}} \geq \frac{\frac{|S^{v^*}|}{w(S^*)}}{\frac{|S^*|}{w(S^*)}} \geq \frac{|S^{v^*}|}{|S(G^*)|} \geq \rho_S \quad (4)$$

Taking into account that, in terms of n , the best known approximation ratio for maximum stable set problem is $O(\log^2 n/n)$ ([6]), (4) concludes the proof of the lemma. ■

Using Proposition 2 and Lemma 1, the following holds for algorithm W_COLOR and the weighted coloring problem.

Proposition 3. *The weighted coloring problem can be approximately solved in polynomial time within approximation ratio $O(n/\log n)$.*

3.2 Differential approximation

We present in this section a polynomial time approximation algorithm guaranteeing a differential approximation ratio bounded below by a fixed constant. Consider a graph $G = (V, E, w)$, where w is the vector of the node-weights of G . Then, our algorithm, denoted by DW_COLOR works as follows:

- [a] construct an edge-weighted graph $\bar{G} = (V, E', w')$ where \bar{G} is the complement of G and for any $e = [v, u] \in E'$, $w'(e) = \min\{w(u), w(v)\}$;
- [b] compute a maximum-weight matching M^* of \bar{G} ;
- [c] color the endpoints of any edge of M^* with a new color;

[d] color every exposed node of V (with respect to M^*) with a new color.

Obviously, algorithm `DW_COLOR` is polynomial (since the computation of a maximum-weight matching can be performed in $O(n^{2.5})$ ([10])) and computes a feasible coloring for G . Indeed, the solution computed is a collection of stable sets of size 2 and of singletons. More precisely, this solution can be written as $\hat{\mathcal{S}}_{M^*} = \{\{v, u\} : [v, u] \in M^*\} \cup \{\{x\} : x \text{ exposed with respect to } M^*\}$.

Denote by $\mathcal{S}^* = (S_1^*, \dots, S_p^*)$ an optimal weighted coloring and by $\text{val}_{\bar{G}}(M)$ the value of any maximum weight matching M of \bar{G} . Denote also by $X(M)$ the set of nodes of G exposed with respect to M . Of course, $\text{worst}(G) = \sum_{v \in V} w(v)$ and in all,

$$\text{worst}(G) = \sum_{v \in V} w(v) \quad (5)$$

$$\text{opt}(G) = \sum_{v \in \mathcal{S}^*} w(v) \quad (6)$$

$$\text{val}_{\bar{G}}(M) = \sum_{[v,u] \in M} \min \{w(u), w(v)\} \quad (7)$$

Lemma 2. *For any maximum weight matching M of G , $\text{worst}(G) - \text{val}_{\bar{G}}(M)$ is the value of a feasible weighted coloring \mathcal{S}_M of G where $\mathcal{S}_M = \{\{v, u\} : [v, u] \in M\} \cup \{\{x\} : x \in X(M)\}$.*

Proof. Fix a maximum weight matching M in G . Then, from (5) and (7), $\text{worst}(G) - \text{val}_{\bar{G}}(M) = \sum_{[v,u] \in M} \max\{w(u), w(v)\} + \sum_{x \in X(M)} w(x)$. But this is exactly the value of the weighted coloring \mathcal{S} claimed in the statement of the lemma. ■

Corollary 3. $\text{val}_{\text{DW_COLOR}}(G) = \text{worst}(G) - \text{val}_{\bar{G}}(M^*)$.

Lemma 3. *The value of the coloring $\hat{\mathcal{S}}$ constructed by algorithm `DW_COLOR` is not larger than the value of any coloring \mathcal{S}_M , where \mathcal{S}_M is as in Lemma 2.*

Proof. Simply remark that, by construction, $\text{val}_{\bar{G}}(M^*) \geq \text{val}_{\bar{G}}(M)$. Then, Lemma 2 and Corollary 3 complete the proof. ■

Consider now the optimal solution $\mathcal{S}^* = (S_1^*, \dots, S_p^*)$ fixed above and consider the following (deteriorated) coloring induced by \mathcal{S}^* :

- for any S_i^* , $i = 1, \dots, p$, order its nodes in nonincreasing weight order; let $L = (v_1^i, \dots, v_{|S_i^*|}^i)$ be the ordered list obtained (v_1^i is then a maximum-weight node);
- starting from v_1^i , color any two consecutive nodes of L with a new color ($\lfloor |S_i^*|/2 \rfloor$ colors have so been used, $i = 1, \dots, p$);
- if $|S_i^*|$ is odd, $i = 1, \dots, p$, then color the last node of L (a minimum-weight one) with a new color.

It is easy to see that the above procedure transforms every feasible weighted coloring of G into another feasible one. Let us denote by $\mathcal{S}' = \{S_1^*, \dots, S_p^*\}$ the coloring obtained from \mathcal{S}^* and by $\text{val}(\mathcal{S}')$ its value. Then, the following holds.

Lemma 4. $\text{val}(\mathcal{S}') \leq (\text{worst}(G) + \text{opt}(G))/2$.

Proof. Consider, for $i = 1, \dots, p$, $S_i^* \in \mathcal{S}^*$, $S_i^* = \{v_1^i, \dots, v_{|S_i^*|}^i\}$, $v_1^i \geq \dots \geq v_{|S_i^*|}^i$ and assume (in order to be as general as possible) $|S_i^*|$ odd. The contribution of S_i^* in $\text{val}(\mathcal{S}')$ is then

$$\text{val}(\mathcal{S}') \Big|_{S_i^*} = w(v_1^i) + w(v_3^i) + \dots + w(v_{|S_i^*-2}^i) + w(v_{|S_i^*|}^i) \quad (8)$$

(the term $w(v_{|S_i^*|}^i)$ is the one corresponding to the singleton) while its contribution in $\text{opt}(G)$ is $w(v_1^i)$. Remark that, since the nodes of S_i^* are ordered in nonincreasing weight order, then, for $j = 3, 5, \dots, |S_i^*|$,

$$w(v_j) \leq \frac{w(v_{j-1}) + w(v_j)}{2} \quad (9)$$

Using (9) for $j = 3, 5, \dots, |S_i^*|$, (8) can be rewritten as

$$\text{val}(\mathcal{S}^{*'})|_{S_i^*} \leq w(v_1^i) + \sum_{j=1}^{|S_i^*|} \frac{w(v_{j-1}) + w(v_j)}{2} = \frac{w(v_1^i)}{2} + \sum_{j=1}^{|S_i^*|} \frac{w(v_{j-1}) + w(v_j)}{2} \quad (10)$$

But the first term in the last equality of (10) is the half of the contribution of S_i^* in $\text{opt}(G)$ (recall that S_i^* is a stable set of the optimal coloring) while the second term is the half of the contribution of S_i^* in $\text{worst}(G)$. Consequently, summing (10) for $S_i^* \in \mathcal{S}^*$, one gets the result claimed. ■

Some easy algebra shows that $\text{val}(\mathcal{S}^{*'}) \leq (\text{worst}(G) + \text{opt}(G))/2$ implies $(\text{worst}(G) - \text{val}(\mathcal{S}^{*'}))/(\text{worst}(G) - \text{opt}(G)) \geq 1/2$. Hence, using Lemmata 2 to 4, the following result holds.

Proposition 4. *The approximation ratio achieved by DW_COLOR is bounded below by $1/2$. This bound is tight.*

Proof. In order to prove the tightness claimed, one can consider a graph G_m consisting of a matching of size m (in other words, G_m is a collection of m disjoint edges) the endpoints of its edges having weight 1. Then, $\text{opt}(G) = 2$, $\text{worst}(G) = 2m$, while $\text{val}_{DW_COLOR} = m$. ■

3.3 A general negative result

We show here that the difference between the values of the solution computed by any polynomial time approximation algorithm and the optimal one is unbounded for TSS.

Consider any class \mathcal{G}' of graphs and a node-weighted graph $G \in \mathcal{G}'$. Denote by w , the vector of the (integer) weights of G . Construct then a node weighted graph G' identical to G with node-weight vector $w' = (c+1)w$. Obviously, every solution \mathcal{S} for G of value $\text{val}(G)$ remains feasible for G' and its value becomes $\text{val}(G') = (c+1)\text{val}(G)$. Assume a polynomial time approximation algorithm A guaranteeing, for some fixed integer $c > 1$ and for every node-weighted graph G , $\text{val}_A(G) - \text{opt}_{TSS}(G) \leq c$. Then, $\text{val}_A(G') - \text{opt}_{TSS}(G') \leq c$ implies $\text{val}_A(G) \leq \text{opt}_{TSS}(G) + c/(c+1)$. Since node-weights are assumed integer, the last inequality implies $\text{val}_A(G) = \text{opt}_{TSS}(G)$ and the following result holds.

Proposition 5. *Unless $P = NP$, for any $c \in \mathbb{N}$, $c \geq 1$, no polynomial time algorithm can compute a solution of TSS in any class of graphs such that the difference between its value and the optimal value is bounded above by c .*

Finally, it can be easily proved that, in general graphs, TSS cannot be solved by polynomial time approximation scheme.

4 The bipartite case and some related cases

4.1 The complexity of time slot scheduling

In this section $G = (V, E, w)$ will be a weighted bipartite graph where L (resp. R) is the “left set” (resp. “right set”) of nodes and each edge has one endpoint in L and the other in R . An instance of TSS is given by a bipartite weighted graph G with a positive integer q . Let $\text{TSS}(G, q)$ be the following problem: does there exist a k -coloring \mathcal{S} of G with $C(\mathcal{S}) \leq q$?

Proposition 6. *TSS(G, q) is NP-complete in the strong sense even if G is a bipartite graph.*

Proof. We shall use a reduction from 1-PrExt (see Bodlaender et al. ([5])) which is the following: “given a bipartite graph $G = (V, E)$ with $|V| \geq 3$ and three nodes v_1, v_2, v_3 , does there exist a 3-coloring (S_1, S_2, S_3) of (the nodes of) G such that $v_i \in S_i$ for $i = 1, 2, 3$?”. 1-PrExt was shown to be NP-complete in Bodlaender et al. ([5]). Consider an instance of 1-PrExt given by a bipartite graph and specific nodes v_1, v_2, v_3 . It is immediate to see that we may assume $\{v_1, v_2, v_3\} \subseteq L$ (otherwise if $1 \leq |\{v_1, v_2, v_3\} \cap L| \leq 2$ the problem is easy). We introduce three new nodes u_1, u_2, u_3 in R and edges $[v_i, u_i]$ for $i \neq j$ and $1 \leq i, j \leq 3$. In the new bipartite graph G' we associate weights $w(u_i) = w(v_i) = 2^{3-i}$ for $i = 1, 2, 3$ and $w(v) = 1$ for every other node v in G' . Then we set $q = 7$ and we consider problem TSS($G', q = 7$). There exists a k -coloring \mathcal{S} of G' for some $k \leq \Delta(G') + 1$ with $C(\mathcal{S}) \leq 7$ if and only if there exists a 3-coloring (S_1, S_2, S_3) of G with $v_i \in S_i$, $i = 1, 2, 3$. This can be seen as follows: suppose we have a 3-coloring (S_1, S_2, S_3) of G with $v_i \in S_i$, $i = 1, 2, 3$; then we get a 3-coloring \mathcal{S}' of the nodes of G' by setting $S'_i = S_i \cup \{u_i\}$ for $i = 1, 2, 3$. For \mathcal{S}' we have $C(\mathcal{S}') = w(S_1) + w(S_2) + w(S_3) = 1 + 2 + 4 = 7$.

Conversely assume G' has a k -coloring $\mathcal{S}' = (S'_1, \dots, S'_k)$ with $C(\mathcal{S}') \leq 7$ for some $k \leq \Delta(G') + 1$. By inspection we notice that the only way of having a k -coloring with $C(\mathcal{S}') \leq q = 7$ is to have $\{u_1, v_1\} \subseteq S'_1$; similarly we deduce $\{u_2, v_2\} \subseteq S'_2$ otherwise $C(\mathcal{S}') \geq 8$ and as a consequence $\{u_3, v_3\} \subseteq S'_3$. But then we must have $k = 3$. So we obtain a 3-coloring of G' by setting $S_i = S'_i \setminus \{u_i\}$ for $i = 1, 2, 3$. ■

Remark 3. The above proof shows that the problem TSS(G, q) is NP-complete for a bipartite graph G as soon as the weights $w(v)$ can take three different values. As a consequence TSS(G, q) is also NP-complete if G is a comparability graph (i.e., a graph whose edges can be transitively oriented) see Berge ([3]). Moreover, the result also holds for bipartite graphs of fixed maximum degree greater than 12. ■

Proposition 7. *If $G = (V, E, w)$ is a bipartite weighted graph where the weights $w(v)$ are 1 or t ($t > 1$) then one can construct a k -coloring \mathcal{S} minimizing $C(\mathcal{S})$ in polynomial time.*

Proof. We shall show that an optimal k -coloring \mathcal{S} can be found in polynomial time; it will satisfy $k \leq 3$ and $C(\mathcal{S}) \leq 2t$.

Let $V(t)$ (resp., $V(1)$) be the set of nodes v with $w(v) = t$ (resp., $w(v) = 1$). We can assume $|V(t) \cap L| \geq |V(t) \cap R|$ where L is the left set and R the right set of nodes of G .

If $V(t)$ is not stable, then $\mathcal{S} = (L, R)$ is an optimal coloring with $C(\mathcal{S}) = 2t$ since there is at least one edge $[u, v]$ of G with $u, v \in V(t)$.

If $V(t)$ is stable, then we have two cases:

- if $V(t) \cap R = \emptyset$, then $\mathcal{S} = (L, R)$ is an optimal coloring with $C(\mathcal{S}) = t + 1$;
- if $V(t) \cap R \neq \emptyset$ then $\mathcal{S} = (V(t), L \setminus V(t), R \setminus V(t))$ is an optimal coloring with $C(\mathcal{S}) = t + 2$ when $t > 2$ and $\mathcal{S} = (L, R)$ is an optimal coloring with $C(\mathcal{S}) = 2t$ when $1 < t \leq 2$.

This can be seen as follows: there exists in G a connected component containing a chain with an odd number of edges and with both end-nodes u, v in $V(t)$. In any bicoloring of G nodes u and v cannot have the same color, so we cannot have a bicoloring \mathcal{S} with $C(\mathcal{S}) = t + 1$. Clearly, we can determine in polynomial time in which case we are. ■

Example 1. Figure 1 illustrates the different cases for a “bi-weighted” bipartite graph. Black nodes represent set $V(t)$, while white nodes nodes draw set $V(1)$. In Figure 1(a), $V(t)$ is not stable. Then, $\mathcal{S} = (L, R)$ and $c(\mathcal{S}) = 2t$. In Figure 1(b) $V(t) \cap R = \emptyset$. Then, $\mathcal{S} = (L, R)$ and $c(\mathcal{S}) = t + 1$. Finally, in Figure 1(c), $V(t) \cap R \neq \emptyset$ and set $V(t)$ is stable. Here, coloring (L, R) has cost $2t$, while coloring $(V(t), L \setminus V(t), R \setminus V(t))$ has cost $t + 2$. ■

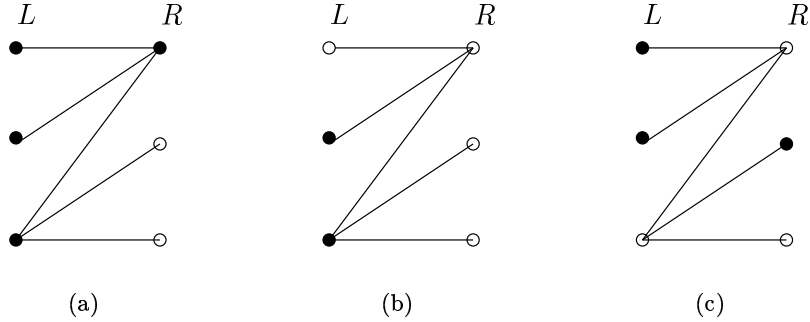


Figure 1: The possible cases when G is bipartite with weights in $\{1, t\}$.

Remark 4. If G is the complement of a bipartite graph, or more generally a graph where the maximum size $\alpha(G)$ of a stable set satisfies $\alpha(G) \leq 2$, then a k -coloring \mathcal{S} with minimum cost $C(\mathcal{S})$ can be constructed in polynomial time as shown in Boudhar and Finke ([7]): a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ will have $|S_i| \leq 2$. It can be constructed by considering \overline{G} , giving a cost $w'(e) = \min\{w(u), w(v)\}$ to each edge $e = [u, v]$ of \overline{G} and constructing a matching M with maximum cost in \overline{G} ; the edges $[u, v]$ give the sets $S_i = \{u, v\}$ with $|S_i| = 2$ and the nodes unsaturated by M will give the sets S_i with $|S_i| = 1$. This construction is identical to the one described in 3.2.

Since there is a one to one correspondence between the colorings \mathcal{S} of G and the matchings M of \overline{G} , and since we also have $C(\mathcal{S}) = \sum_{v \in V} w(v) - C'(M)$, the result follows. A maximum cost matching can be constructed in $O(n^{2.5})$ (see for instance Cook et al. ([10])).

If G is the complement of a bipartite graph, then we have to construct a maximum cost matching in a bipartite graph. The algorithm is then similar in spirit to the one given in Garey and Johnson ([16]) for solving a two-processor problem with unit processing times and resource constraints: a matching in a auxiliary graph (where nodes are operations) gives the pairs of operations to be processed simultaneously. In addition, it is shown in Boudhar and Finke ([7]) that the problem becomes difficult as soon as there are release dates (even if all weight $w(v) = 1$). ■

Example 2. Figure 2 gives an example of a graph G meeting the conditions of Remark 4. Thick edges in Figure 2(b) define a matching $M = \{[a, g], [b, e], [c, d]\}$ in \overline{G} with maximum cost $K(M) = \min\{5, 4\} + \min\{3, 2\} + \min\{2, 4\} = 8$. The corresponding coloring in G (Figure 2(a)) is $\mathcal{S} = (S_1, S_2, S_3, S_4) = (\{a, g\}, \{b, e\}, \{c, d\}, \{f\})$ with $C(\mathcal{S}) = \max\{5, 4\} + \max\{3, 2\} + \max\{2, 4\} + 3 = 15 = (\sum_{v \in V} w(v)) - K(M) = 23 - 8$. ■

More generally we can establish the following property of optimal k -colorings \mathcal{S} in a weighted graph $G = (V, E, w)$ with chromatic number $\chi(G) = q$.

Proposition 8. Let $G = (V, E, w)$ be a weighted graph and let $q = \chi(G)$ be its chromatic number. Assume $w(v) \in \{t_1, \dots, t_r\}$ with $t_1 > \dots > t_r$ for each node v . Then every k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ which minimizes $C(\mathcal{S})$ satisfies: $w(S_i) > w(S_{i+q-1})$, for any $i \leq k - q$. In particular, $k \leq 1 + r(\chi(G) - 1)$.

Proof. Assume now that $\chi(G) \geq 2$ and there is an optimum k -coloring which does not have the property; let $\mathcal{S} = (S_1, \dots, S_k)$ be this coloring and choose the smallest i such that $w(S_i) = \dots = w(S_{i+q-1}) \geq w(S_k)$. We have $i \leq k - q$ by assumption. Now $S_i \cup S_{i+1} \cup \dots \cup S_k$

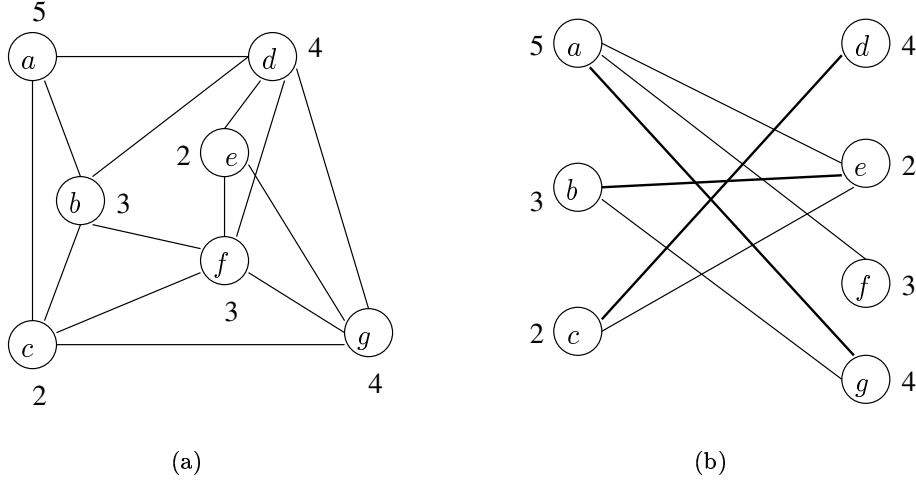


Figure 2: Construction of an optimal TSS in the complement of a bipartite graph.

generates a subgraph G' of G ; we have therefore $\chi(G') \leq \chi(G) = q$, so there exists a q -coloring $(S'_i, \dots, S'_{i+q-1})$ of G' (with $i + q - 1 < k$). Assuming $w(S'_i) \geq w(S'_{i+1}) \geq \dots \geq w(S'_{i+q-1})$ we have $w(S'_i) = w(S_i)$ and $w(S'_{i+s}) \leq w(S'_i) = w(S_{i+s})$ for $s = 1, \dots, q - 1$. Setting $S'_j = S_j$ for $j = 1, \dots, i - 1$ we get an $(i + q - 1)$ -coloring $S' = (S'_1, \dots, S'_{i+q-1})$ of G with $C(S') < C(S)$ which is a contradiction. ■

Corollary 4. *If $G = (V, E, w)$ is a weighted bipartite graph with weights $w(v) \in \{t_1, \dots, t_r\}$ with $t_1 > \dots > t_r$ then any k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ minimizing $C(\mathcal{S})$ satisfies: $w(S_1) > w(S_2) > \dots > w(S_{k-1}) \geq w(S_k)$. In particular $k \leq 1 + r$.*

Remark 5. Notice that the above result is not in contradiction with remark 3 stating that $\text{TSS}(G, q)$ is **NP**-complete for a bipartite graph G and weights $w(v)$ in $\{t_1, t_2, t_3\}$. ■

Remark 6. Proposition 8 is best possible in the following sense: given any pair of positive integers q, r there exists a weighted graph $G = (V, E, w)$ with chromatic number $\chi(G) = q$ and weights $w(v) \in \{t_1, \dots, t_r\}$ such that every optimal k -coloring \mathcal{S} has $k = 1 + r(\chi(G) - 1) = 1 + r(q - 1)$. G is constructed from a clique K_0 on q nodes with weights $w(v) = 1$ for each v . We will recursively append a clique K_j to a node v of a clique K_i ($i < j$): this means introduce $q - 1$ new nodes which form a clique K_j on q nodes with v . This will clearly keep the chromatic number equal to $\chi(K_0) = q$

The construction runs as follows: start from K_0 ; let $G_1 = K_0$ be the graph in construction; for $i = 1$ to $r - 1$ do the following: append a new clique on q nodes to each node of G_i ; give all new nodes v a weight $w(v) = 2^i$ and let G_{i+1} be the resulting weighted graph. One can verify that every optimal coloring $\mathcal{S} = (S_1, \dots, S_k)$ of G_r will have:

$$\begin{aligned}
w(S_1) &= \dots = w(S_{q-1}) &= 2^{r-1} \\
w(S_q) &= \dots = w(S_{2q-2}) &= 2^{r-2} \\
&\dots \\
w(S_{(r-2)(q-1)+1}) &= \dots = w(S_{(r-1)(q-1)}) &= 2^1 \\
w(S_{(r-1)(q-1)+1}) &= \dots = w(S_{r(q-1)}) &= w(S_{r(q-1)+1}) = 2^0 \quad \blacksquare
\end{aligned}$$

4.2 On the approximability of time slot scheduling in bipartite graphs

4.2.1 Standard approximation

4.2.1.1 A positive result

We propose in this section a polynomial time approximation algorithm achieving a constant approximation ratio for TSS in bipartite graphs. This algorithm, denoted by `BIP_WCOLOR` works as follows:

1. sort the nodes of G in nonincreasing weight order; let $\bar{L} = (v_1, v_2, \dots, v_n)$ be the list obtained;
2. starting from v_1 color the nodes of \bar{L} with color c whenever it is possible;
3. trivially color the remaining uncolored nodes with at most two new colors b and g ; store the solution obtained during steps 2 and 3;
4. trivially color the nodes of L by c and the ones of R by b ; store the solution obtained;
5. output the smallest between the solutions stored in steps 3 and 4.

Proposition 9. *Algorithm `BIP_WCOLOR` polynomially solves TSS in bipartite graphs within approximation ratio bounded above by $4/3$. This bound is tight.*

Proof. Let us first briefly discuss steps 2 and 3 of `BIP_WCOLOR`. In step 1 it tries to construct a stable set including the heaviest nodes of G and it colors them with c . Suppose that color c includes nodes of both sets L and R . Then, in step 3 color the uncolored nodes of L with say b and the uncolored ones of R with g . It is easy to see that the solution so constructed is a feasible coloring. On the other hand, if step 2 colors all the nodes of say L by r , then two colors, say c and b are sufficient to color all the nodes of G (the ones of R being colored with b).

Obviously, the weight of color c equals $w(v_1)$ (the greatest among the node weights). Suppose now that step 2 stops while a node of weight $w(v_1)/k$, for some $k > 1$, has been encountered. Then,

$$\text{opt}(G) \geq w(v_1) + \frac{w(v_1)}{k} \quad (11)$$

$$\text{val}_{\text{BIP_WCOLOR}}(G) \leq \begin{cases} w(v_1) + \frac{2w(v_1)}{k} & \text{the final solution is the one of step 3} \\ 2w(v_1) & \text{the final solution is the one of step 4} \end{cases} \quad (12)$$

Combination of (11) with the first line of (12) leads to an approximation ratio bounded above by $(k+2)/(k+1)$, while (11) and the second line of (12) lead to an approximation ratio bounded above by $2k/(k+1)$. The first of the ratios is decreasing with k while the second one is increasing and equality holds for $k = 2$. In this case the common value for both ratios is $4/3$.

Consider now a bipartite graph $B = (L, R, E)$ and set $L = \{l_1, l_2, \dots, l_{|L|}\}$ and $R = \{r_1, r_2, \dots, r_{|R|}\}$. Set also $w(l_1) = w(r_1) = 2$ and $w(l_i) = w(r_j) = 1$, $i = 2, \dots, |L|$, $j = 2, \dots, |R|$. Set finally $E = \{[l_1, r_j], [r_1, l_i] : j = 2, \dots, |R|, i = 2, \dots, |L|\}$. Then, when `BIP_WCOLOR` runs in B the two candidate solutions are either $(\{l_1, r_1\}, L \setminus \{l_1\}, R \setminus \{r_1\})$ (steps 2 and 3), or (L, R) (step 4). Both of them have cost 4. On the other hand, the optimal solution for B is $(\{l_1, r_1\}, L \setminus \{l_1\} \cup R \setminus \{r_1\})$ of total cost 3. So, ratio $4/3$ is attained for B and the proof of the proposition is complete. ■

4.2.1.2 An inapproximability result

We give in this section an inapproximability result for TSS in bipartite graphs. It provides a lower bound on the approximation ratio of every polynomial time algorithm.

As in the proof of Proposition 6, consider a bipartite graph $G = (L, R, E)$ and $\{x, y, z\} \subset L$ and construct the bipartite graph G' . Recall that $\text{opt}_{\text{TSS}}(G') = 7$ iff the answer for 1-PrExt in G is *yes*. Assume now that there exists a polynomial time approximation algorithm A achieving, for some $\epsilon_0 > 0$, an approximation ratio $(8/7) - \epsilon_0$.

- If $\text{opt}_{\text{TSS}}(G') = 7$ (the answer for 1-PrExt in G is *yes*), then $\text{val}_A(G') = 8 - 7\epsilon_0$ and, since val has to be integer, $\text{val}_A(G') = 7$;
- on the other hand, if $\text{opt}_{\text{TSS}}(G') > 7$ (the answer for 1-PrExt in G is *no*), then $\text{val}_A(G') \geq \text{opt}_{\text{TSS}}(G') \geq 8$.

Consequently, on the hypothesis that a polynomial time approximation algorithm A achieves, for some $\epsilon_0 > 0$, approximation ratio $(8/7) - \epsilon_0$ for TSS, one can in polynomial time decide on the correct answer for 1-PrExt simply by reading the value of the solution computed by A and the following proposition summarizes the above discussion.

Proposition 10. *Unless $\mathbf{P} = \mathbf{NP}$, for any $\epsilon > 0$ no polynomial time algorithm achieves an approximation ratio bounded above by $(8/7) - \epsilon$ for TSS in bipartite graphs.*

4.2.2 Differential approximation

Assume $G = (L, R, E, w)$ is a node-weighted bipartite graph, $|L| + |R| = n$, assume that nodes are ordered in nonincreasing weight order, i.e., $w(v_1) \geq w(v_2) \geq \dots \geq w(v_n)$ and, for simplicity, set $w_i = w(v_i)$, $i = 1, \dots, n$. Consider then the following algorithm, called $\mathbf{C_SCHEME}$ in what follows and run it with parameters G and a fixed constant $\epsilon > 0$:

1. set $\eta = \lceil 1/\epsilon \rceil$;
2. set $S_L = \{v_{4\eta+3}, \dots, v_n\} \cap L$;
3. set $S_R = \{v_{4\eta+3}, \dots, v_n\} \cap R$;
4. set $\tilde{\mathcal{S}}$ the best partition into stable sets of the nodes $v_1, \dots, v_{4\eta+2}$;
5. output $\hat{\mathcal{S}} = S_L \cup S_R \cup \tilde{\mathcal{S}}$.

Since η is a fixed constant, the set $\tilde{\mathcal{S}}$ of step 4 of algorithm $\mathbf{C_SCHEME}$ can be computed by an exhaustive search in constant time. Consequently, the whole complexity of $\mathbf{C_SCHEME}$ is linear in n .

Consider now an optimal solution $\mathcal{S}^* = (S_1^*, S_2^*, \dots, S_p^*)$, denote by $w_{i_1}, w_{i_2}, \dots, w_{i_p}$ the respective weights of the stable sets of \mathcal{S}^* and recall that, since nodes are assumed ordered in decreasing weight order, $w_1 = w_{i_1} \geq w_{i_2} \geq \dots \geq w_{i_p}$. Obviously,

$$\text{opt}(G) = w_{i_1} + w_{i_2} + \dots + w_{i_p} \quad (13)$$

Denote now by G' the subgraph of G induced by the node-set $\{v_1, \dots, v_{4\eta+2}\}$ and recall that $\tilde{\mathcal{S}}$ is optimal for G' . Then, the following lemma holds.

Lemma 5.

$$(i) \quad |\tilde{\mathcal{S}}| \leq 2\eta + 2.$$

(ii) $\text{val}(\tilde{\mathcal{S}}) = \text{opt}(G') \leq \text{opt}(G)$.

Proof of (i). Suppose $|\tilde{\mathcal{S}}| \geq 2\eta + 3$. Then, there exist at least three singletons, say $\{v\}$, $\{v'\}$ and $\{v''\}$, in $\tilde{\mathcal{S}}$. Obviously, the nodes of at least two of the above singletons, say v and v' , belong to the same bipartition of G . But then, one can contract stable sets $\{v\}$, and $\{v'\}$ into a single stable set, producing so a new coloring of value smaller than the one of $\tilde{\mathcal{S}}$, a contradiction.

Proof of (ii). Set $V = L \cup R$, $\mathcal{S}^{*'} = (S_1^* \cap V, S_2^* \cap V, \dots, S_p^* \cap V)$ and denote by $\text{val}(\mathcal{S}^{*'})$ the value of $\mathcal{S}^{*'}$. Clearly, $\mathcal{S}^{*'}$ is feasible for G' ; hence, $\text{opt}(G') = \text{val}(\tilde{\mathcal{S}}) \leq \text{val}(\mathcal{S}^{*'}) \leq \text{val}(\mathcal{S}^*) = \text{opt}(G)$. So, the proof of (ii) and of the lemma is complete. ■

We are now ready to prove the main result of this section, i.e., that TSS admits a polynomial time differential approximation scheme in bipartite graphs.

Proposition 11. *For any fixed $\epsilon > 0$, the differential approximation ratio of $\mathcal{C_SCHEME}$ when called with inputs G and ϵ , is bounded below by $1 - \epsilon$.*

Proof. Using (5),(13) and (i) of Lemma 5, one gets

$$\text{worst}(G') - \text{opt}(G') = \sum_{i=1}^{4\eta+2} w_i - \sum_{j=1}^{|\tilde{\mathcal{S}}|} w_{i_j} \geq (4\eta + 2 - (2\eta + 2)) w_{4\eta+2} = 2\eta w_{4\eta+2} \geq \frac{2}{\epsilon} w_{4\eta+2} \quad (14)$$

On the other hand, denoting by $w(S_L)$ and $w(S_R)$, the weights of the sets S_L and S_R , respectively, we have

$$\begin{aligned} w(S_L) &\leq w_{4\eta+2} \\ w(S_R) &\leq w_{4\eta+2} \end{aligned} \quad (15)$$

Using now (14) and (15), (ii) of Lemma 5, and the fact that $\text{worst}(G') \leq \text{worst}(G)$, we get for $\text{val}_{\mathcal{C_SCHEME}}(G)$:

$$\begin{aligned} \text{val}_{\mathcal{C_SCHEME}}(G) &= w(S_L) + w(S_R) + \text{opt}(G') \\ &\leq (1 - \epsilon)\text{opt}(G') + \epsilon \left(\text{opt}(G') + \frac{1}{\epsilon} w(S_L) + \frac{1}{\epsilon} w(S_R) \right) \\ &\leq (1 - \epsilon)\text{opt}(G') + \epsilon \left(\text{opt}(G') + \frac{2}{\epsilon} w_{4\eta+2} \right) \\ &\leq (1 - \epsilon)\text{opt}(G) + \epsilon \text{worst}(G) \end{aligned} \quad (16)$$

Starting from (16), some easy algebra leads to the result claimed and completes the proof of the proposition. ■

We note finally that with exactly the same arguments as the ones for the proof of Proposition 10 (at the beginning of the section 4.2.1.2) and taking into account that $\text{worst}(G') = n - 3 + 14 = n + 11$, one can conclude that, unless $\mathbf{P} = \mathbf{NP}$, no polynomial time algorithm can achieve differential approximation ratio greater than $1 - (1/(n + 4))$, and the following holds.

Proposition 12. *Unless $\mathbf{P} = \mathbf{NP}$, TSS cannot be solved in bipartite graphs by a fully polynomial time approximation scheme.*

4.3 The split graphs

To conclude the study of the bipartite case, we have to examine the situation of *split* graphs, i.e., graphs G in which the node set $V(G)$ can be partitioned into a stable set S and a clique K . These graphs can be considered as intermediate between bipartite graphs and complements of bipartite graphs.

Proposition 13. *TSS(G, q) is NP-complete in the strong sense if G is a split graph.*

Proof. We shall use a reduction from the following problem called Min-Cover: given a collection $\mathcal{C} = (\overline{C}_i : i \in I)$ of subsets \overline{C}_i of a set \overline{S} and a positive integer q ($q \leq |I|$) does there exist a sub-collection $\mathcal{C}' = (\overline{C}_j : j \in J)$ with $|J| \leq q$ and $\cup_{j \in J} \overline{C}_j = \overline{S}$? Min-Cover was shown to be NP-complete in [22].

We shall transform Min-Cover into TSS(G, q) where G is a split graph. Let us construct graph G as follows. Each element \overline{v} of \overline{S} becomes a node v of a stable set S ; each subset \overline{C}_i in \mathcal{C} corresponds to a node c_i of the clique K of G . The set $N(c_i)$ of neighbors of node c_i is given by: $N(c_i) = \{v : \overline{v} \in \overline{S}\} \setminus \{v : \overline{v} \in \overline{C}_i\}$. The weights are given by $w(c_i) = |I|$, $i \in I$, and $w(v) = |I| + 1, v \in S$.

Now there exists a cover $\mathcal{C}' = (\overline{C}_j : j \in J) \subset \mathcal{C}$ with $\cup_{j \in J} \overline{C}_j = \overline{S}$ and $|\mathcal{C}'| = |J| \leq q$ if and only if there exists in G a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ with $C(\mathcal{S}) \leq |I|^2 + q$. This can be seen as follows.

- Assume that we have a cover \mathcal{C}' with $|\mathcal{C}'| \leq q$; for each $\overline{v} \in \overline{S}$ let $j(\overline{v}) = \min\{j \in J : \overline{v} \in \overline{C}_j\}$; for $i = 1, \dots, |I|$ let $S_i = \{c_i\} \cup \{v : j(\overline{v}) = i\}$. This gives an $|I|$ -coloring \mathcal{S} of G : since \mathcal{C}' is a cover, every node $v \in S$ is introduced into a set S_i . Furthermore $\overline{v} \in \overline{C}_j$ implies that nodes v and c_j of G are not linked by an edge, hence each S_i is a stable set. Now $C(\mathcal{S}) = \sum_{i=1}^{|I|} w(S_i)$ and we have $w(S_i) = |I|$ if $\overline{C}_i \notin \mathcal{C}'$ and $w(S_i) \leq |I| + 1$ if $\overline{C}_i \in \mathcal{C}'$. Hence $C(\mathcal{S}) \leq |I|^2 + q$.
- Conversely let $\mathcal{S} = (S_1, \dots, S_k)$ be a k -coloring of G with $C(\mathcal{S}) \leq |I|^2 + q$. We have $q \leq |I|$ and hence $q < |I| + 1$. Since G has a clique K on $|I|$ nodes c_i satisfying $w(c_i) = |I|$ and since all nodes $v \in S$ have $w(v) = |I| + 1$, the coloring \mathcal{S} is an $|I|$ -coloring. This means that we can assume $c_i \in S_i$ for $i = 1, \dots, |I|$. Since $C(\mathcal{S}) \leq |I|^2 + q$, there are at most q subsets S_i in \mathcal{S} with $w(S_i) = |I| + 1$. For every node $v \in S$ there exists a stable set $S_{i(v)}$ which contains v . This implies that for each $v \in S$, there exists a subset $\overline{C}_{i(v)}$ in \mathcal{C} which contains \overline{v} . Let $J = \{j : j = i(v) \text{ for some } v \in S\}$; it follows that $\cup_{j \in J} \overline{C}_j = \overline{S}$. Furthermore, since $w(S_{i(v)}) = |I| + 1$ for any v and since $C(\mathcal{S}) \leq |I|^2 + q$, we have $|J| \leq q$. Hence $\mathcal{C} = (\overline{C}_j | j \in J)$ is the required cover. ■

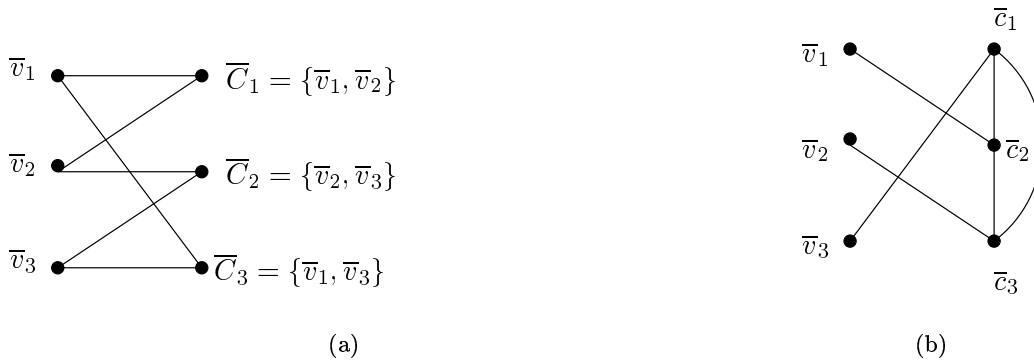


Figure 3: Reduction of Min-Cover to TSS in a split graph.

Example 3. Figure 3 illustrates the reduction of Proposition 13 with $|I| = 3$ and $q = 2$. In Figure 3(a) an instance of Min-Cover is shown: $\mathcal{C} = (\overline{C}_i | i \in I, |I| = 3)$, $\overline{C}_1 = \{\overline{v}_1, \overline{v}_2\}$,

$\overline{C}_2 = \{\overline{v}_2, \overline{v}_3\}$, $\overline{C}_3 = \{\overline{v}_1, \overline{v}_3\}$. The question is here “ $\exists \mathcal{C}' = (\overline{C}_j | j \in J)$ with $|J| \leq q = 2$?”. In Figure 3(b) the split graph constructed from the Min-Cover instance of Figure 3(a) is illustrated: $w(\overline{v}_j) = |I| + 1 = 4$, $j = 1, 2, 3$; $w(\overline{C}_i) = |I| = 3$, $i = 1, 2, 3$. Here the question is “ $\exists ak$ – coloring $\mathcal{S} = (S_1, \dots, S_k)$ with $C(\mathcal{S}) \leq |I|^2 + q = 11$?”. With respect to the TSS-instance just constructed, $S_1 = \{\overline{c}_1, \overline{v}_1, \overline{v}_2\}$, $S_2 = \{\overline{c}_2, \overline{v}_3\}$, $S_3 = \{\overline{c}_3\}$, with $w(S_1) = 4$, $w(S_2) = 4$ and $w(S_3) = 3$. So, $\mathcal{C}' = \{\overline{C}_1, \overline{C}_2\}$ with $J = \{1, 2\}$ ($|J| = 2$) and $\overline{C}_1 \cup \overline{C}_2 = \overline{S}$. ■

The proof of Proposition 13 shows that the problem is **NP**-complete even if the weights can take only two values. As observed in Boudhar and Finke ([7]), the problem is easy if all weights are 1; it amounts to finding the chromatic number of a split graph $G = (K, S, E)$. We have $\chi(G) = |K|$ where K is the maximum clique. Boudhar and Finke ([7]) mention also that when all weights are 1, the problem is easy if G is the complement of a chordal graph (no chordless cycle of size at least 4), or the complement of a circular-arc graph (intersection graph of a family of intervals on a cycle), or the complement of a comparability graph (graph having a transitive orientation of its edges). It also follows from the proof of Proposition 13 that $\text{TSS}(G, q)$ is **NP**-complete if G is a chordal graph (i.e., a graph where every cycle of length at least four has a chord), since a split graph is a chordal graph (see Berge ([3])).

5 An edge coloring model

5.1 Complexity results

If the weighted graph $G = (V, E, w)$ is a line-graph $L(H)$, then our node coloring problem becomes an edge coloring problem in a graph H where the edges e have weights $w(e)$.

Proposition 14. *TSS(G, q) is **NP**-complete in the strong sense if G is the line-graph $L(H)$ of a regular bipartite multigraph H with $\Delta(H) = 3$.*

Proof. We shall start from the following **NP**-complete problem called 2-SIM: given a bipartite regular multigraph $H = (V, E)$ and two disjoint (partial) matchings M_1^*, M_2^* , does there exist an edge 3-coloring $\mathcal{M} = (M_1, M_2, M_3)$ of H such that $M_i^* \subseteq M_i$ for $i = 1, 2$?

2-SIM was shown to be **NP**-complete in de Werra and Erschler ([13]). We shall transform 2-SIM into an edge 3-coloring problem in a bipartite multigraph H which will correspond to $\text{TSS}(L(H), q)$.

Replace each edge $e = [u, v]$ in M_2^* by edges $[u, v_e]$, $[v_e, u_e]_1$, $[v_e, u_e]_2$ and $[u_e, v]$ where u_e and v_e are new nodes and introduce $[u, v_e]$ and $[u_e, v]$ in M_2^* and $[v_e, u_e]_1$ in M_1^* . The resulting graph is still regular bipartite with degree 3. Notice that in an edge 3-coloring the edges $[u, v_e]$ and $[u_e, v]$ will have the same color (which can be given to edge $[u, v]$ of the initial graph). Notice also that in the resulting graph every edge of M_2^* is adjacent to an edge of M_1^* .

Furthermore there exists in the resulting graph an edge coloring $(\widehat{M}_1, \widehat{M}_2, \widehat{M}_3)$ with $M_i^* \subseteq \widehat{M}_i$ for $i = 1, 2$ if and only if there exists an edge coloring (M_1, M_2, M_3) in the initial graph with $M_i^* \subseteq M_i$ for $i = 1, 2$.

Let us give weights $w(e) = 2^{3-i}$ to all edges $e \in M_i^*$ for $i = 1, 2$ and weights $w(e) = 1$ to all remaining edges of H . Let \widehat{H} be the resulting weighted graph. Then, by defining the weight $w(M_i)$ of a matching M_i as the maximum of the weights of the edges in M_i , we have the following: \widehat{H} has an edge k -coloring $\widehat{\mathcal{M}} = (\widehat{M}_1, \dots, \widehat{M}_k)$ with $C(\widehat{\mathcal{M}}) = w(\widehat{M}_1) + \dots + w(\widehat{M}_k) \leq q = 7$ if and only if H has an edge 3-coloring $\mathcal{M} = (M_1, \dots, M_3)$ with $M_i^* \subseteq M_i$ ($i = 1, 2$). Clearly from an edge 3-coloring \mathcal{M} with $M_i^* \subseteq M_i$ ($i = 1, 2$) we derive an edge 3-coloring $\widehat{\mathcal{M}} = \mathcal{M}$ of \widehat{H} with $C(\widehat{\mathcal{M}}) = q = 7$.

Conversely any coloring $\widehat{\mathcal{M}}$ of \widehat{H} with $M_1^* \not\subseteq \widehat{M}_i$ for all i has $C(\widehat{\mathcal{M}}) \geq 8$. So, assume $M_1^* \subseteq \widehat{M}_1$. Again if $M_2^* \not\subseteq \widehat{M}_i$ for all $i \in \{2, 3\}$ the coloring $\widehat{\mathcal{M}}$ has $C(\widehat{\mathcal{M}}) \geq 8$. So, we have

$M_2^* \subseteq \widehat{M}_2$. Now $k = 3$, otherwise the k -coloring $\widehat{\mathcal{M}}$ cannot have $C(\widehat{\mathcal{M}}) \leq 7$. So $\mathcal{M} = \widehat{\mathcal{M}}$ is an edge 3-coloring of H which satisfies the requirements. ■

In what follows, we denote by $\text{ETSS}(G_k, q, k)$ the edge coloring version of TSS in k -regular bipartite graphs $G_k = (L, R, E)$. Note that the remark of the beginning of this section together with proposition 14 establish the following corollary.

Corollary 5. *ETSS is NP-complete even in cubic bipartite graphs.*

Proposition 15. *ETSS is strongly NP-complete for any k -regular bipartite graph with $k \geq 3$.*

Proof. In order to prove the proposition we inductively reduce $\text{ETSS}(G_{k-1}, q)$ to $\text{ETSS}(G_k, 3q)$. Consider a $(k-1)$ -regular bipartite graph $G_{k-1} = (L, R, E)$ and denote by w_{k-1} its edge-weight vector. Remark that $|L| = |R|$ and let r_i and l_i be for $i = 1, \dots, |L|$ the nodes of R and L , respectively. Construct a copy $G'_{k-1} = (L', R', E')$ of G_{k-1} ($L = L', R = R', E = E'$) and denote by r'_i and l'_i the nodes of R' and L' , respectively. For $i = 1, \dots, |L|$ link r_i with l'_i and l_i with r'_i . Set $w_k(e) = w_{k-1}(e)$ for $e \in E \cup E'$ and $w_k(e) = 2q$ for $e \in \{[r_i, l'_i], [l_i, r'_i] : i = 1, \dots, |L|\}$. Obviously, G_k is k -regular. One can easily see that there exists an edge coloring of weight q in G_{k-1} , iff there exists an edge coloring of weight $3q$ in G_k . Then Corollary 5 completes the proof of the proposition. ■

Remark 7. When the graph G is simple (no multiple edges) the results of Propositions 14 and 15 remain valid even in simple cubic bipartite graphs. Briefly, the reduction of a cubic bipartite multigraph G to a simple cubic bipartite graph B transforms the simple edges of G into simple edges in B and any multi-edge of G (any vertex of G has at most one incident multi-edge) into the gadget of figure 4. Note that in any feasible edge coloring of B , $\{\text{color}(a), \text{color}(b)\} = \{\text{color}(a'), \text{color}(b')\}$. So, one can prove that, on the one hand, 2-SIM is NP-complete in a simple cubic graph much more simply than in [15] and, on the other hand, that ETSS is NP-complete in a simple cubic graph. ■

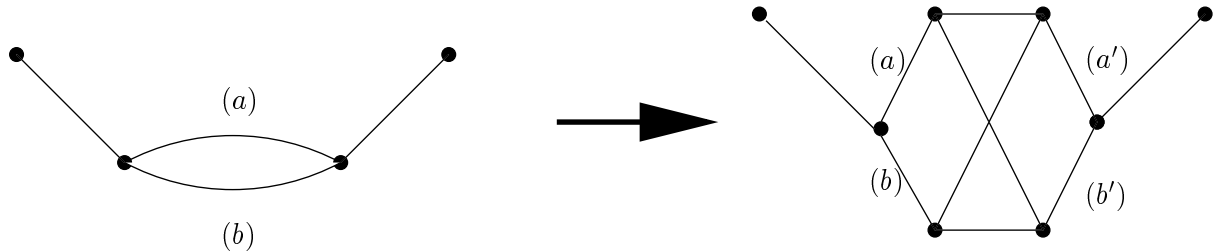


Figure 4: Transformation of a cubic bipartite multigraph G into a simple cubic bipartite graph B .

Notice that the proof of Proposition 14 shows that the problem is difficult if there are three possible value for the weights $w(e)$. We may now consider the special case where in a bipartite multigraph H every edge e has weight $w(e) \in \{1, t\}$ where $t > 1$ is an integer. We have to find an edge k -coloring $\widehat{\mathcal{M}}$ such that $C(\widehat{\mathcal{M}})$ is minimum.

Proposition 16. *TSS($L(H), q$) can be solved in polynomial time if H is bipartite with weights $w(e) \in \{1, t\}$ on the edges.*

Proof. Let H be a bipartite multigraph and let $E(s)$ be the set of edges e with weight $w(e) = s$ for $s = 1, t$. Let $\Delta(s)$ be the maximum degree of the partial graph $H(s)$ generated by the edges

in $E(s)$ for $s = 1, t$. Clearly if $\Delta(t) = \Delta(H)$ (maximum degree in H), then any edge $\Delta(H)$ -coloring \mathcal{M} of H will give $C(\mathcal{M}) = \Delta(H)t$ which is minimum in this case; such a coloring exists from the theorem of König (see Berge ([3])).

So we may assume that $\Delta(t) < \Delta(H)$. We shall try to find an edge k -coloring $\mathcal{M} = (M_1 \dots, M_k)$ with $w(M_1) = \dots = w(M_{\Delta(t)}) = t$ and $w(M_{\Delta(t)+1}) = \dots = w(M_k) = 1$ with k as small as possible. Such a k can be found by using network flow techniques.

Construct a network $N(r)$ as follows: remove from H all edges in $E(t)$ and replace each edge $[u, v]$ in $E(1)$ by an arc $\vec{e} = (u, v)$ with capacity $c(\vec{e}) = 1$ and lower bound of flow $l(\vec{e}) = 0$; here r is a nonnegative integer. Introduce a source s_0 with arc (s_0, u) for each $u \in L$ which is adjacent in H to at least one edge of $E(1)$; set $l(s_0, u) = d_{H(1)}(u) - r$ and $c(s_0, u) = \Delta(t) - d_{H(t)}(u)$. In the same way, introduce a sink t_0 with arc (v, t_0) from each node v of R which is adjacent in H to at least one edge of $E(1)$; set $l(v, t_0) = d_{H(1)}(v) - r$ and $c(v, t_0) = \Delta(t) - d_{H(t)}(v)$.

From the construction of $N(r)$, we see that if for some positive r there exists an (integral) compatible flow f from s_0 to t_0 , it defines a subset $E^*(f) \subset E(1)$ (by taking all edges $[u, v]$ such that $f(u, v) = 1$ in $N(r)$) such that:

- the partial graph H^* of H defined by the edges of $E(t) \cup E^*(f)$ has $\Delta(H^*) = \Delta(t)$;
- the partial graph \widehat{H} of H defined by the remaining edges has $\Delta(\widehat{H}) \leq r$.

We have to find the smallest possible r for which $N(r)$ contains a feasible flow (notice that for r large there exists a compatible flow; take for instance the flow of value 0 on each arc). So, we can always start from $r = \Delta(H)$ and decrease it stepwise. Such an r will give us an edge $(\Delta(H(t)) + r)$ -coloring \mathcal{M} such that $C(\mathcal{M}) = \Delta(H(t))t + r$.

But such a coloring \mathcal{M} may not be of minimum cost. We have to examine also edge k -colorings $\mathcal{M} = (M_1, \dots, M_k)$ where $w(M_i) = t$ for the first $\Delta(H(t)) + \ell$ matchings and minimize the number r of matchings M_j with $w(M_j) = 1$. This can be done by the network flow algorithm described above by increasing the capacity of all arcs (s_0, u) and (v, t_0) by ℓ units. We will have to do this for $\ell = 0$ to $\Delta(H) - \Delta(H(t))$.

We will keep the edge k -coloring \mathcal{M} with the smallest cost $C(\mathcal{M})$. ■

Example 4. Figure 5 illustrates the above algorithm. In Figure 5(a), an edge “biweighted” bipartite graph H is drawn with $\Delta(H) = 3$ and $\Delta(H(t)) = 2$. The thick edges represent set $E(t)$, while the thin ones represent $E(1)$. In Figure 5(b), the network $N(r = 1)$ for constructing E^* with $\ell = 0$ is shown. In the network $N(r = 1)$, $\bar{\ell}, \bar{c}$ indicates a lower bound $\bar{\ell}$ of flow and a capacity \bar{c} . Moreover, no feasible flow exists in $N(r)$. In Figure 5(c), the network $N(r = 2)$ for constructing E^* with $\ell = 0$ is shown; the flow values are given in parentheses when non zero. Here, as one can see, there exists a feasible flow in the network; $E^*(f) = \{[b, d], [c, \bar{f}]\}$. Figure 5(d) illustrates the construction of the $\Delta(H(t)) + \ell$ matchings M_i with $w(M_i) = t$ for $\ell = 0$; $M_1 = \{[b, d], [a, e], [c, \bar{f}]\}$, $M_2 = \{[a, d], [c, e]\}$ ($w(M_1) = w(M_2) = t$). Figure 5(e), illustrates the construction of the r matchings M_i with $w(M_i) = 1$: $M_3 = \{[a, \bar{f}]\}$, $M_4 = \{[c, \bar{f}]\}$ ($w(M_3) = w(M_4) = 1$); this gives $C(M_1, M_2, M_3, M_4) = 2t + 2$. Figure 5(f) shows the construction of E^* with $\ell = 1 = \Delta(H) - \Delta(H(t))$. As one can see, with $r = 0$ one gets a flow $f(u, v) = 1$ for any $[u, v] \in E(1)$; so, $E^* = E \setminus E(t)$. Finally, in Figure 5(g) the edge coloring \mathcal{S} resulting from the construction of Figure 5(f) is illustrated with $C(\mathcal{S}) = 3t$. For any of the three colors M_1, M_2, M_3 , $w(M_i) = t$. ■

Remark 8. In Rendl ([27]) it is shown that $\text{TSS}(G, q)$ is **NP**-complete if G is the line graph $L(H)$ of a complete bipartite graph $K_{n,n}$; the nodes of $L(H)$ have degree $2n - 2$. The interest of the above proof is to deal with the case of degrees at most 3. In addition Rendl ([27]) states Proposition 16 for the special case of the line graph of $K_{n,n}$. ■

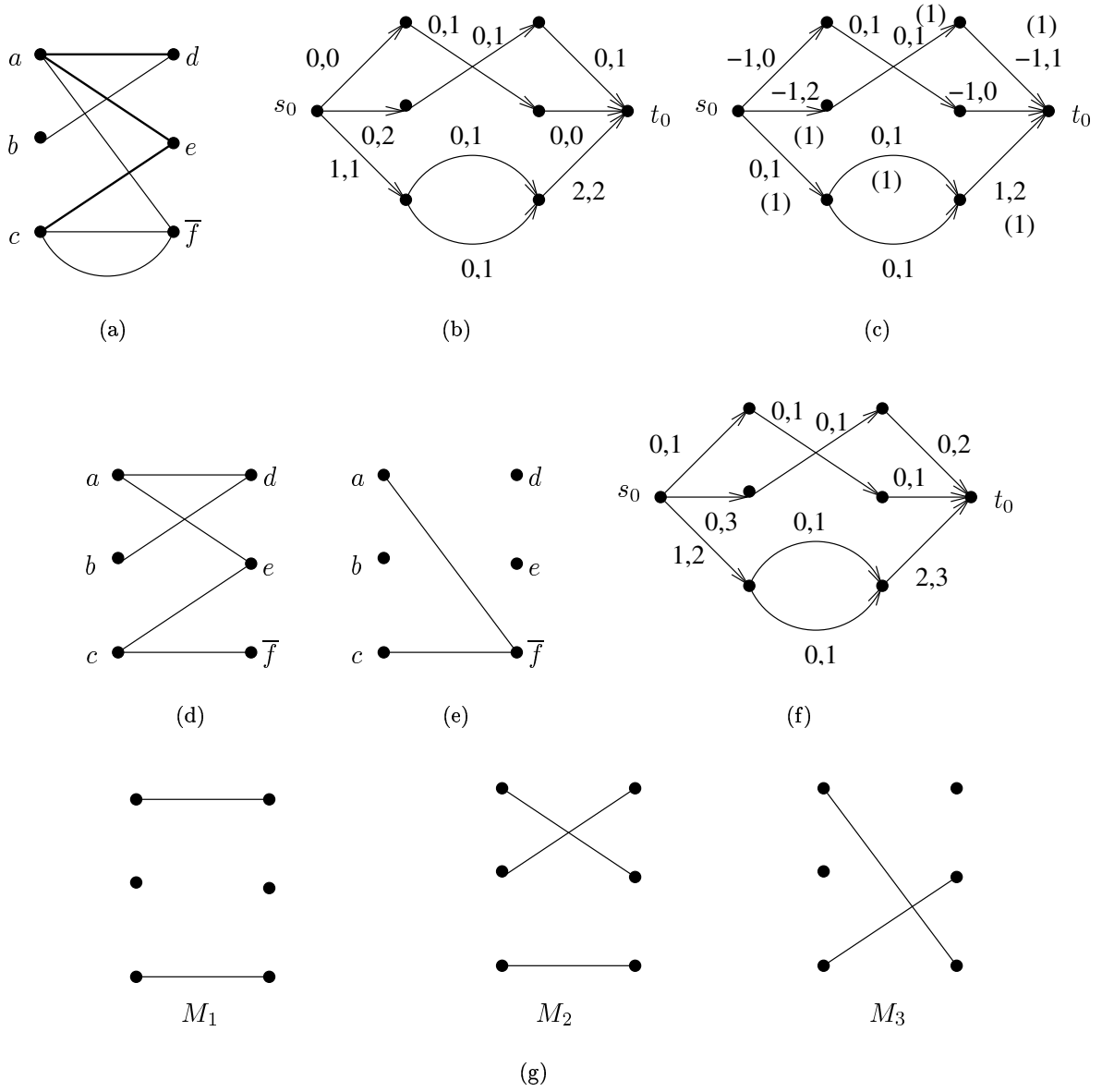


Figure 5: An illustration of the algorithm of Proposition 16.

Remark 9. When $G = L(H)$ is the line-graph of a bipartite graph, then every optimal k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ satisfies $k \leq 1 + r(\chi(G) - 1) = 1 + r(\chi(L(H)) - 1)$ if the weights of the nodes of G can take r different values. For $r = 1$, we have obviously $k = \Delta(H)$ from the theorem of König (see Berge [3]) and for $r = 2$, we get $k \leq 2\Delta(H) - 1$; according to Corollary 4, this bound is valid also for any $r \geq 2$. Moreover the bound of Corollary 4 is best possible: for any s , there exists a tree H with $\Delta(H) = s$ and such that any optimal edge k -coloring $\mathcal{M} = (M_1, \dots, M_k)$ of H has $k = 2\Delta(H) - 1$. H is constructed as follows: start from a graph containing a node v_0 , then for $i = 1, \dots, s$ do the following:

- Introduce a node v_1 and an edge $e_i = [v_0, v_i]$ with weight $w(e_i) = 1$;
- For $j = 1, \dots, s-1$ introduce a node v_{ij} and an edge $e_{ij} = [v_i, v_{ij}]$ with weight $w(e_{ij}) = s+1$.

Now H is a tree with $\Delta(H) = s$; every optimal edge k -coloring $\mathcal{M} = (M_1, \dots, M_k)$ has $k = 2\Delta(H) - 1$. Consider an edge $(2s - p)$ -coloring with $2 \leq p \leq s$. There must be at least s different colors on the edges $[v_i, v_{ij}]$; if there are only $s - 1$, then we need s additional colors for the edges $[v_0, v_i]$ and this would give an edge $(2s - 1)$ -coloring. So the cost of such an edge $(2s - p)$ -coloring is at least $s(s + 1) = s^2 + s$. But the k -coloring defined by:

$$\begin{aligned} M_j &= \{[v_i, v_{ij}] : i = 1, \dots, s\} & \text{for } j = 1, \dots, s - 1 \\ M_{s-1+i} &= \{[v_0, v_i]\} & \text{for } i = 1, \dots, s \end{aligned}$$

has cost $(s + 1)(s - 1) + s = s^2 + s - 1$. ■

5.2 On the approximation of the ETSS in bipartite graphs

5.2.1 Standard approximation

Remark first that, by König's theorem ([23]), the optimal solution of the (unweighted) edge covering achieves standard approximation ratio Δ for ETSS, for any $\Delta \geq 3$, where Δ is the maximum degree of the input graph G .

In what follows in this section, we restrict ourselves to bipartite graphs of maximum degree $\Delta = 3$. Extension of the results presented for $\Delta \geq 4$ is, as we will see, immediate. We are given a bipartite graph $G = (L, R, E)$, with $E = \{e_1, \dots, e_{|E|}\}$; denote by w the edge-weight vector and, for $E' \subseteq E$, by $G[E']$ the partial subgraph of G induced by E' , and consider the following algorithm EW_COLOR:

1. rank edges of G in decreasing weight order; set $E = \{e_1, e_2, \dots, e_{|E|}\}$;
2. set $M_1 = M_2 = \dots = M_{|E|} = \emptyset$ ($M_j, j = 1, \dots, |E|$, are the edge colors) ;
3. for $i = 1$ to $|E|$ do
 - (a) set $j_0 = \min\{j = 1, \dots, |E| : M_j \cup \{e_i\} \text{ is a matching}\}$;
 - (b) set $M_{j_0} = M_{j_0} \cup \{e_i\}$;
4. set $\mathcal{S}_1 = (M_1, \dots, M_r)$ the list of the non-empty matchings of $(M_1, M_2, \dots, M_{|E|})$;
5. set $E_j = \{e_1, \dots, e_j\}$;
6. set $E' = \operatorname{argmax}\{E_j : G[E_j] \text{ has maximum degree at most } 2\}$;
7. compute an optimal 2-coloring (M'_1, M'_2) , such that $e_1 \in M'_1$, for E' ;
8. complete (M'_1, M'_2) by running steps 1 to 3 in $G \setminus G[E']$;

9. set $\mathcal{S}_2 = (M'_1, M'_2, \dots, M'_{r'})$ the edge coloring computed in steps 8 and 9;
10. output $\mathcal{S} = \operatorname{argmin}\{\operatorname{val}(\mathcal{S}_1), \operatorname{val}(\mathcal{S}_2)\}$.

Lemma 6. $r \leq 5, r' \leq 5$.

Proof. Suppose that one of r, r' , say $r \geq 6$ and denote by $e = [u, v]$ the first edge of M_6 . Since e cannot be added in any of M_1, \dots, M_5 , there exists at least one edge incident to e in each one of these matchings; in other words there exist at least 5 edges adjacent to e , one in any M_i , $i = 1, \dots, 5$. The fact that $\Delta(G) = 3$ means that at most two of these edges are adjacent to u and at most two of them adjacent to v . Consequently, on the hypothesis that $\Delta(G) = 3$, at most four matchings cannot accept e and the proof of the lemma is complete. ■

Proposition 17. *EW_COLOR achieves approximation ratio $5/3$ in $O(|E| \log |E|)$.*

Proof. Following Lemma 6 one can set $\mathcal{S}_1 = (M_1, \dots, M_5)$, $\mathcal{S}_2 = (M'_1, \dots, M'_5)$ (some of the M_i , or M'_i , $i = 1, \dots, 5$ may be empty). Fix an optimal solution \mathcal{S}^* and denote by M_1^*, M_2^*, M_3^* the three largest matchings of \mathcal{S}^* . Finally, denote by e_{ij}, e'_{ij} and e_{ij}^* ; the first edges of M_j, M'_j , $j = 1, \dots, 5$ and $M_j^*, j = 1, 2, 3$, respectively. Then,

$$\operatorname{val}_{\text{EW_COLOR}}(\mathcal{S}_1) \leq \sum_{j=1}^5 w(e_{ij}) \quad (17)$$

$$\operatorname{val}_{\text{EW_COLOR}}(\mathcal{S}_2) \leq \sum_{j=1}^5 w(e'_{ij}) \quad (18)$$

$$\operatorname{opt}(G) \geq \sum_{j=1}^3 w(e_{ij}^*) \quad (19)$$

We now consider two cases, namely, $w(e_{i_3}^*) \geq w(e_{i_3})$ and $w(e_{i_3}^*) < w(e_{i_3})$.

For case $w(e_{i_3}^*) \geq w(e_{i_3})$, remark first that

$$w(e_{i_1}^*) = w(e_{i_1}) \quad (20)$$

$$w(e_{i_2}^*) \geq w(e_{i_2}) \quad (21)$$

In fact, (21) holds because, ad contrario, $E_{i_2} = \{e_{i_1}, \dots, e_{i_2}\}$ (see step 5 of algorithm EW_COLOR) would be a matching and then e_{i_2} would be introduced (by steps 3a and 3b) in M_1 . Combination of (17), (19), (20) and (21) leads to

$$\operatorname{val}_{\text{EW_COLOR}}(\mathcal{S}_1) \leq \frac{5}{3} \operatorname{opt}(G) \quad (22)$$

Case $w(e_{i_3}^*) < w(e_{i_3})$ implies, in particular, that $G[E_{i_3}]$ (defined as in step 5 of algorithm EW_COLOR) has maximum degree 2. In this case, step 7 colors the edges of $G[E_{i_3}]$ with 2 colors; consequently, $E_{i_3} \subseteq E'$ (step 6). Furthermore, since the edges of $E' \setminus E_{i_3}$ are lighter than the ones of E_{i_3} , we can extend the coloring of $G[E_{i_3}]$ to a coloring of $G[E']$ without augmenting the total cost; so,

$$w(e_{i_2}^*) = w(e'_{i_2}) \quad (23)$$

On the other hand, $G[E' \cup \{e'_{i_3}\}]$ is of maximum degree 3, hence $e_{i_3}^* \in E' \cup \{e'_{i_3}\}$ and

$$w(e_{i_3}^*) \geq w(e'_{i_3}) \quad (24)$$

Combination of (18), (19), (23) and (24) leads

$$\text{val}_{\text{EW_COLOR}}(\mathcal{S}_2) \leq \frac{5}{3} \text{opt}(G) \quad (25)$$

and (22) and (25) conclude the proof of the proposition. ■

Remark 10. The same analysis as the one in the proof of Proposition 17 concludes that ETSS is approximable within standard approximation ratio bounded above by $(2\Delta - 1)/3$, for any $\Delta \geq 3$. ■

Proposition 18. *Unless $\mathbf{P} = \mathbf{NP}$, for any $\epsilon > 0$ no polynomial time algorithm achieves approximation ratio bounded above by $(2^k/(2^k - 1)) - \epsilon$, even in k -regular bipartite graphs.*

Proof. From the proofs of Propositions 14 and 15, one can see that ETSS in regular bipartite graphs of degree at least 3 is \mathbf{NP} -complete whenever the optimal value of the instance is at most $2^k - 1$. ■

5.2.2 Differential approximation

As previously we first assume $G = (L, R, E)$ is a bipartite graph of maximum degree $\Delta = 3$ and with edge-weight vector w , and consider the following algorithm, denoted by `EC_SCHEME` in what follows:

- (a) set $k = \lceil 1/\epsilon \rceil$;
- (b) rank the edges in E in decreasing-weight order; set $E = \{e_1, \dots, e_{|E|}\}$;
- (c) set $E' = \{e_1, e_2, \dots, e_{3k+5}\}$;
- (d) optimally color $G[E']$;
- (e) greedily complete the edge coloring of step (d) in order to color E with at most three colors (in other words we omit weights and color the unweighted version of G).

Proposition 19. *Algorithm `EC_SCHEME` is a polynomial time differential approximation scheme for ETSS.*

Proof. Set $\mathcal{S}^* = \{M_1^*, \dots, M_r^*\}$ be an optimal solution of $G[E']$ (step (d)). By Lemma 6, $r \leq 5$. So,

$$\text{worst}(G[E']) - \text{opt}(G[E']) \geq 3kw(e_{3k+5}) \geq \frac{3}{\epsilon}w(e_{3k+5}) \quad (26)$$

$$\text{val}_{\text{EC_SCHEME}}(G[E']) \leq \text{opt}(G[E']) + w(e_{3k+6}) + w(e_{3k+7}) + w(e_{3k+8}) \quad (27)$$

After some easy algebra and taking into account that edges in E are ranked in decreasing weight order, (27) gives

$$\begin{aligned} \text{val}_{\text{EC_SCHEME}}(G[E']) &\leq (1 - \epsilon)\text{opt}(G[E']) + \epsilon \left(\text{opt}(G[E']) + \frac{3}{\epsilon}w(e_{3k+5}) \right) \\ &\leq (1 - \epsilon)\text{opt}(G) + \epsilon \text{worst}(G) \end{aligned} \quad (28)$$

where in (28) one uses (26). This completes the proof of the proposition. ■

Remark 11. One can easily see that the result of Proposition 19 holds also for any fixed $\Delta > 3$ and for any graph (not necessarily bipartite). ■

6 Cographs

The case of cographs (or equivalently graphs containing no induced chain P_4 on four nodes) has to be mentioned. These graphs, also called P_4 -free graphs, are a subclass of the perfectly ordered graphs introduced in Chvátal ([9]); for the perfectly ordered graphs, an order θ on the node set V can be defined in such a way that for any induced subgraph G' of the original graph G the *greedy sequential algorithm* (GSC) based on the order θ' induced by θ on the nodes of G' gives a minimum coloring of G' (i.e., a coloring in exactly $\chi(G')$ colors). Here the GSC algorithm based on an order θ consists in examining consecutively the nodes as they occur in θ and coloring them with the smallest possible color. As observed in de Werra ([12]) a graph G is a cograph if and only if for all induced subgraphs G' of G the GSC based on any order θ gives a coloring of G' in $\chi(G')$ colors.

Proposition 20. *If $G = (V, E, w)$ is a weighted cograph, then all k -colorings $\mathcal{S} = (S_1, \dots, S_k)$ minimizing $C(\mathcal{S})$ satisfy $k = \chi(G)$.*

Proof. Assume there is a k' -coloring $\mathcal{S}' = (S'_1, \dots, S'_{k'})$ which minimizes $C(\mathcal{S}')$ and for which $k' > \chi(G)$. We can order the nodes of G by taking consecutively the nodes of S'_1 , those of S'_2 and so on. Using the resulting order θ we can apply the GSC algorithm which will produce a k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ with $k = \chi(G)$; it will satisfy $w(S_1) \geq w(S_2) \geq \dots \geq w(S_k)$. Assume also that $w(S'_1) \geq w(S'_2) \geq \dots \geq w(S'_{k'})$. Indeed each node $v \in S'_j$ will satisfy $v \in S_i$ with $i \leq j$ after recoloring. So we have $w(S'_j \setminus \{v\}) \leq w(S'_j) \leq w(S'_i)$ and since $w(v) \leq w(S'_j) \leq w(S'_i)$ we also have $w(S'_i \cup \{v\}) = w(S'_i)$; so we will finally have $w(S_i) \leq w(S'_i)$ for $i = 1, \dots, k$. Now $S_{k+1} = \emptyset$, so $0 = w(S_{k+1}) < w(S'_{k+1})$ because all weights are positive. Hence $C(\mathcal{S}) < C(\mathcal{S}')$ which is a contradiction. ■

We can now show that there is a polynomial algorithm which constructs a k -coloring \mathcal{S} with minimum $C(\mathcal{S})$; such a result can be expected from graphs like cographs for which several generally difficult coloring problems are easier, see Jansen and Scheffler ([20]) for instance.

Proposition 21. *Let $G = (V, E, w)$ be a weighted cograph. Then the k -coloring \mathcal{S} constructed by the GSC algorithm based on any order θ where $u < v$ (u before v in θ) implies $w(u) \geq w(v)$ minimizes $C(\mathcal{S})$.*

Proof. Let $t_1 > t_2 > \dots > t_r$ be the values taken by the weights $w(v)$ in G . Every k -coloring $\mathcal{S} = (S_1, \dots, S_k)$ of G with $k = \chi(G)$ and $w(S_1) \geq w(S_2) \geq \dots \geq w(S_k)$ satisfies:

$$w(S_i) \geq \max \{t_s : \omega(G(s)) \geq i\}$$

where $\omega(H)$ denotes the maximum size of a clique in a graph H and $G(s)$ is the subgraph generated by all nodes v with $w(v) \geq t_s$. Indeed any such k -coloring will have the first $\omega(G(1))$ sets S_i with $w(S_i) = t_1$; also the first $\omega(G(2))$ sets S_i will have $w(S_i) \geq t_2$ and generally the first $\omega(G(s))$ sets S_i will have $w(S_i) \geq t_s$.

Now consider the k -coloring $\overline{\mathcal{S}} = (\overline{S}_1, \dots, \overline{S}_k)$ obtained by applying the GSC algorithm based on any order θ with nonincreasing weights. Let $p(s)$ be the largest color given to a node v with $w(v) = t_s$; let v_0 be such a node. Since cographs are perfectly ordered graphs (with respect to our order θ), it follows by considering the subgraph G' of G generated by v_0 and all its predecessors in θ that there is in G' a clique $K \ni v_0$ with $K \cap \overline{S}_i \neq \emptyset$ for $i = 1, \dots, p(s)$. Clearly $w(\overline{v}) \geq t_s$ for every node $\overline{v} \in K$. This means that $p(s) \leq \omega(G(s))$. Now $\overline{\mathcal{S}}$ (where we have $w(\overline{S}_1) \geq \dots \geq w(\overline{S}_k)$) satisfies $w(\overline{S}_i) = \max\{t_s : p(s) \geq i\}$; but then $w(\overline{S}_i) \leq \max\{t_s : \omega(G(s)) \geq i\}$ and we have $w(\overline{S}_i) = \max\{t_s : \omega(G(s)) \geq i\}$ for $i = 1, \dots, k$. So $\overline{\mathcal{S}}$ is a k -coloring with minimum cost. ■

Remark 12. The above proof shows in fact that if we are given a perfectly ordered graph G and if the order θ of nonincreasing weights is such that the GSC algorithm gives a minimum coloring (i.e., a k -coloring with $k = \chi(G)$), then one can find a k -coloring \mathcal{S} which minimizes $C(\mathcal{S})$. For cographs, this condition was satisfied since any order θ could be chosen to construct a minimum coloring. ■

Remark 13. Proposition 21 is best possible in the following sense. If G is simply a P_4 , then we may have no optimal k -coloring \mathcal{S} with $k = \chi(G)$.

Consider for instance the weighted graph G with nodes a, b, c, d having $w(a) = w(d) = 3$, $w(b) = w(c) = 1$; the unique coloring \mathcal{S} with minimum cost $C(\mathcal{S})$ is $(\{a, d\}, \{b\}, \{c\})$ with cost 5 and $k = 3$.

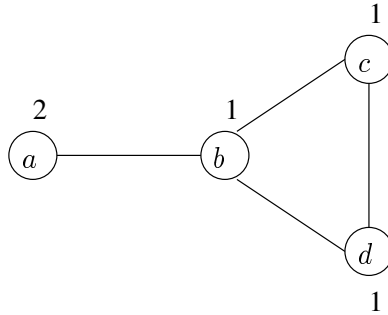


Figure 6: A cograph where $\min\{c(\mathcal{S})\} > \max\{w(K)\}$.

In addition even if there is an optimum k -coloring with $k = \chi(G)$, it may not exist a solution where, as in the proof of Proposition 21, the number $p(1)$ of subsets S_i with $w(S_i) = t_1 = \max\{w(v) : v \in V\}$ is equal to the chromatic number $\chi(G(1))$ of the subgraph generated by all nodes v with $w(v) = t_1$. For instance, if we change the weights $w(b), w(c)$ to 2, then the optimal k -coloring is $(\{a, c\}, \{b, d\})$ with $k = 2 = \chi(G)$, but $G(1)$ is the subgraph generated by $\{a, d\}$ and $\chi(G(1)) = 1$, while the optimal coloring \mathcal{S} has $2 > \chi(G(1)) = 1$ subsets S_1, S_2 with $w(S_1) = w(S_2) = 3 = t_1$.

Finally, it should be observed that a trivial lower bound on the minimum cost of a coloring is simply the maximum weight $w(K)$ of a clique K in the graph G considered, i.e., the sum of the weights of the nodes in K . In spite of the many properties of cographs, one should notice that such a bound may not always be attained as shown in Figure 6. Here, $\max\{w(K)\} = 3$, $\mathcal{S} = (\{a, c\}, \{b\}, \{d\})$ with $c(\mathcal{S}) = 2 + 1 + 1 = 4$. For any k -coloring \mathcal{S}' with $k \geq 3$, $c(\mathcal{S}') \geq 4$. ■

Acknowledgments. Part of this research was carried out while the second author was visiting the CERMSEM (CEntre de Recherches de Mathématiques, Statistique et Economie Mathématique) at the University of Paris 1 (Panthéon-Sorbonne) in January and February 2001. The support of this institution is gratefully acknowledged.

References

- [1] L. Alfandari and V. T. Paschos. Master-slave strategy and polynomial approximation. *Comput. Opt. Appl.*, 16:231–245, 2000.
- [2] A. Bar-Noy, M. Bellare, M. M. Halldórsson, H. Shachnai, and T. Tamir. On chromatic sums and distributed resource allocation. *Inform. and Comput.*, 140:183–202, 1988.
- [3] C. Berge. *Graphs and hypergraphs*. North Holland, Amsterdam, 1973.
- [4] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Scheduling Computer and Manufacturing Processes*. Springer Verlag, Berlin, 1996.
- [5] H. L. Bodlaender, K. Jansen, and G. J. Woeginger. Scheduling with incompatible jobs. *Discrete Appl. Math.*, 55:219–232, 1994.
- [6] B. B. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, 1992.
- [7] M. Boudhar and G. Finke. Scheduling on a batch machine with job compatibilities. *JORB-EL*, 2001. To appear.
- [8] R. E. Burkard. On the decomposition of traffic matrices arising in communication satellites. *Computing*, 35:99–112, 1985.
- [9] V. Chvátal. Perfectly ordered graphs. In C. Berge and V. Chvátal, editors, *Topics on Perfect Graphs*, volume 21 of *Annals of Discrete Math.*, pages 253–277.
- [10] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. J. Wiley, 1998.
- [11] D. de Werra. On some combinatorial problems arising in scheduling. *Canad. Operational Research Society Journal*, 8:165–175, 1970.
- [12] D. de Werra. Heuristics for graph coloring. *Computing*, 7:191–208, 1990.
- [13] D. de Werra and J. Erschler. Open shop scheduling with some additional constraints. *Graphs and Combinatorics*, 12:81–93, 1996.
- [14] D. de Werra and Y. Gay. Chromatic scheduling and frequency assignment. *Discrete Appl. Math.*, 49:165–174, 1994.
- [15] J. Fiala. The NP-completeness of the edge precoloring extension problem on bipartite graphs. Unpublished result.
- [16] M. R. Garey and D. S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM J. Comput.*, 4:397–411, 1975.
- [17] S. Gerke. Colouring weighted bipartite graphs with a co-site constraint. *Discrete Math.*, 224:125–138, 2000.
- [18] I. Holyer. The NP-completeness of edge-colouring. *SIAM J. Comput.*, 10:718–720, 1981.
- [19] K. Jansen. Approximation results for the optimum cost chromatic partition problem. In P. M. Pardalos and D. Du, editors, *Theoretical Computer Science, Network Design: Connectivity and Facilities Location*, pages 143–168. 1997.

- [20] K. Jansen and P. Scheffler. Generalized coloring for tree-like graphs. *Discrete Appl. Math.*, 75:135–155, 1997.
- [21] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.
- [22] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of computer computations*, pages 85–103. Plenum Press, New York, 1972.
- [23] D. König. Über graphen und ihrer anwendung auf determinantentheorie und mengenlehre. *Math. Ann.*, 77:453–465, 1916.
- [24] S. Nicoloso, M. Sarrafzadeh, and X. Song. On the sum coloring problem on interval graphs. *Algorithmica*, 23:109–126, 1999.
- [25] C. N. Potts and M. Y. Kovalyov. Scheduling with batching: a review. *European J. Oper. Res.*, 120:228–249, 2000.
- [26] C. N. Potts, V. A. Strusevich, and T. Tautenhahn. Scheduling batches with simultaneous job processing for two-machine shop problems. *J. of Scheduling*, 4:25–51, 2001.
- [27] F. Rendl. On the complexity of decomposing matrices arising in satellite communication. *Oper. Res. Lett.*, 4:5–8, 1985.
- [28] H. U. Simon. On approximate solutions for combinatorial optimization problems. *SIAM J. Disc. Math.*, 3(2):294–310, 1990.
- [29] Z. Tuza. Graph colorings with local constraints-a survey. *Discussiones Mathematicae Graph Theory*, 17:161–228, 1997.