Weighted node coloring: when stable sets are expensive
(Extended abstract)

M. Demange[1], D. de Werra[2], J. Monnot[3], and V. Th. Paschos[3]

[1] ESSEC, France, `demange@essec.fr`
[2] Ecole Polytechnique Fédérale de Lausanne, Switzerland `dewerra@dma.epfl.ch`
[3] LAMSADE, Université Paris-Dauphine, Paris, France,
`{monnot,paschos}@lamsade.dauphine.fr`

Abstract. A version of weighted coloring of a graph is introduced: each node v of a graph $G = (V, E)$ is provided with a positive integer weight $w(v)$ and the weight of a stable set S of G is $w(S) = \max\{w(v) : v \in V \cap S\}$. A k-coloring $\mathcal{S} = (S_1, \ldots, S_k)$ of G is a partition of V into k stable sets $S_1, \ldots, S_k$ and the weight of $\mathcal{S}$ is $w(S_1) + \ldots + w(S_k)$. The objective then is to find a coloring $\mathcal{S} = (S_1, \ldots, S_k)$ of G such that $w(S_1) + \ldots + w(S_k)$ is minimized. Weighted node coloring is **NP**-hard for general graphs (as generalization of the node coloring problem). We prove here that the associated decision problems are **NP**-complete for bipartite graphs, for line-graphs of bipartite graphs and for split graphs. We present approximation results for general graphs. For the other families of graphs dealt, properties of optimal solutions are discussed and complexity and approximability results are presented.

1 Introduction

A k-coloring of $G = (V, E)$ is a partition $\mathcal{S} = (S_1, \ldots, S_k)$ of the node set V of G into stable sets S_i. The objective is here to determine a node coloring minimizing k. A natural generalization of the problem, denoted by WC in what follows, is the one where a strictly positive integer weight $w(v)$ is considered for any node $v \in V$, and where the weight of stable set S of G is $w(S) = \max\{w(v) : v \in S\}$. Then, the objective is to determine $\mathcal{S} = (S_1, \ldots, S_k)$ a node coloring of G minimizing the quantity $\sum_{i=1}^{k} w(S_i)$. This problem is easily shown **NP**-hard; it suffices to consider $w(v) = 1$, $\forall v \in V$ and WC becomes the classical node coloring problem.

In [1] we show that WC is not a toy problem. In terms of scheduling, a weighted node coloring 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 some time slot i are incompatible. In our situation, the lengths of the time slots $1, 2, \ldots, k$ 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 $\text{val}(\mathcal{S}) = \sum_{i=1}^{k} w(S_i)$. where $\mathcal{S} = (S_1, \ldots, S_k)$. The problem then amounts to finding for a weighted graph $G_w = (V, E, w)$ a coloring $\mathcal{S} = (S_1, \ldots, S_k)$ such that $\text{val}(\mathcal{S})$ is minimum. This problem is related to the *batch scheduling* problem which has been studied by several authors (see for instance [2] for a survey, or [3] 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 [4], 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.

After establishing approximation results for the weighted coloring problem in general graphs, we examine some special cases, dealing with bipartite graphs, split graphs and cographs. We also study the weighted edge coloring problem in bipartite graphs. For all these cases, complexity issues as well as approximability will be discussed. For graph theoretical terms not defined here, the reader is referred to [5].

2 General properties

The following proposition describes a general property which will be needed later.

Proposition 1. *Consider an instance of WC given by a weighted graph* $G = (V, E, w)$ *and a coloring* $\mathcal{S}'$. *We can always construct in polynomial time a* k*-coloring* $\mathcal{S} = (S_1, \ldots, S_k)$ *verifying* $\text{val}(\mathcal{S}) \leqslant \text{val}(\mathcal{S}')$ *and* $k \leqslant \Delta(G) + 1$.

Proof (Sketch). Set $\mathcal{S}' = (S'_1, S'_2, \ldots)$ and rank the S'_i's in decreasing weight order. Take S_i (the ith component of the coloring $\mathcal{S}$) such that $S_i \supseteq S'_i$ is a maximal stable set in $G \setminus S'_1 \setminus \ldots \setminus S'_{i-1}$. □

In particular, this result holds for an optimal weighted node coloring of G. If H is the line-graph of G, denoted by $L(G)$, we have the following.

Corollary 1. *If* $G = L(H)$, *then the solution* $\mathcal{S}$ *of Proposition 1 verifies* $k \leqslant 2\Delta(H) - 1$.

We can easily show that in Corollary 1 we have $k \leqslant p(\omega(G) - 1) + 1$ 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 1 follows. Also, it follows from Proposition 1 that the number k of colors in an optimal k-coloring $\text{val}(\mathcal{S})$ 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, [6]).

We can also establish the following property of optimal k-colorings $\mathcal{S}$ in a weighted graph $G=(V,E,w)$ for $w(v)\in\{t_1,\ldots,t_r\}$ with $t_1>\ldots>t_r$ for each node v.

Proposition 2. *Let* $G=(V,E,w)$ *be a* r*-valued weighted graph and let* $q=\chi(G)$ *be its chromatic number. Then every optimal* k*-coloring* $\mathcal{S}^*=(S_1^*,\ldots,S_k^*)$ *satisfies:* $w(S_i^*)>w(S_{i+q-1}^*)$*, for any* $i\leqslant k-q$*. In particular,* $k\leqslant 1+r(q-1)$*. This bound is tight.*

Proof (Sketch). Assume that there exists an index $i\leqslant k-q$ such that $w(S_i^*)=\ldots=w(S_{i+q-1}^*)$; $S_i^*\cup\ldots\cup S_k^*$ induces a subgraph G' verifying $\chi(G')\leqslant\chi(G)$. Thus, we can change sets $S_i^*,\ldots,S_k^*$ by q other sets to obtain a $q+i-1$-coloring with a lower cost, a contradiction. □

3 Approximating weighted coloring in general graphs

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 $\mathrm{worst}(I)$, $\mathrm{val}_{\mathtt{A}}(I)$ and $\mathrm{opt}(I)$ 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 $\mathrm{worst}(I)$ is in fact the optimal solution of a minimization (resp., maximization) problem Π' having the same objective function and the same constraint set as Π. Let us note that computation of the solution realizing $\mathrm{worst}(I)$ 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) $\rho_{\mathtt{A}}(I)=\mathrm{val}_{\mathtt{A}}(I)/\mathrm{opt}(I)$. On the other hand, the differential approximation ratio measures how the value of an approximate solution is placed in the interval between $\mathrm{worst}(I)$ and $\mathrm{opt}(I)$. More formally, it is defined as $\delta_{\mathtt{A}}(I)=|\mathrm{worst}(I)-\mathrm{val}_{\mathtt{A}}(I)|/|\mathrm{worst}(I)-\mathrm{opt}(I)|$. 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). When they are polynomial in $1/\epsilon$ the schemes are called *fully polynomial time approximation scheme.*

The standard 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; 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 [7] 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 for graph coloring, using maximum stable set as slave problem, is given in [8].

Proposition 3. *([9]) In the master-slave approximation game for weighted problems, if the weighted slave problem is approximable within ratio $\rho(n)$, then the weighted master problem is approximable within standard ratio $O(\log n/\rho(n))$.*

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; repeat steps (1) and (2) until all the nodes of the input graph are colored.

Lemma 1. *SLAVE_ WC is approximable in polynomial time within standard ratio $O(\log^2 n/n)$.*

Proof (Sketch). 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: (2a) set $V_v = \{u \in L : w(u) > w(v)\}$, $V = V \setminus (V_v \cup (\{v\} \cup \Gamma(v)))$, $G = G[V]$; (2b) run the maximum stable algorithm of [10] on G; let S_v be the stable set computed; store set $S^v = S_v \cup \{v\}$ as candidate solution for SLAVE_WC; (2c) return to the original graph G; (3) among the sets stored in step (2b), choose one, denoted by $\hat{S}$, maximizing quantity $|S^v|/w(v)$. We prove in [1] that algorithm `SLAVE_WC` achieves, for problem SLAVE_WC the same ratio, $O(\log^2 n/n)$, as the algorithm of [10], called in step (2b) for stable set, this ratio being the best known, in terms of n for the latter problem. □

Using Proposition 3 and Lemma 1, the following holds for algorithm `W_COLOR`.

Proposition 4. *The weighted coloring problem can be approximately solved in polynomial time within standard approximation ratio $O(n/\log n)$.*

We now deal with differential approximation and present 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. The solution computed `DW_COLOR` is a collection of stable sets of size 2 and of singletons.

Proposition 5. *The differential approximation ratio achieved by* `DW_COLOR` *is bounded below by 1/2. This bound is tight.*

Proof (Sketch). Denote by $\mathcal{S}^* = (S_1^*, \ldots, S_p^*)$ an optimal weighted coloring and by $\mathrm{val}_{\bar{G}}(M)$ the value of any maximum weight matching M of $\bar{G}$. For any $G[S_i^*]$, consider a maximum weight matching M_i', set $M' = \cup_{i=1}^{p} M_i'$ and apply steps [b] to [d] of `DW_COLOR` starting from M'; denote by $\mathcal{S}'$ the coloring so obtained. Then, $\mathrm{val}(\mathcal{S}') = \mathrm{worst}(G) - \mathrm{val}_{\bar{G}}(M') \leqslant (\mathrm{worst}(G) + \mathrm{opt}(G))/2$. Finally, since $\mathrm{val}_{\bar{G}}(M^*) \geqslant \mathrm{val}_{\bar{G}}(M')$, the result claimed is easily deduced. The tightness of the ratio is proved in [1] by considering an 1-valued graph G_m induced by a matching of size m. □

Note that algorithm `DW_COLOR` computes an optimal solution when $\alpha(G) \leqslant 2$.

We finish this section by two inapproximability results. Consider any class $\mathcal{G}'$ of graphs and a node-weighted graph $G \in \mathcal{G}'$ and suppose that WC is **NP**-complete for any $G \in \mathcal{G}'$. Then, the following holds.

Proposition 6. *For any class $\mathcal{G}'$ of node-weighted graphs: if WC($\mathcal{G}'$) is **NP**-complete, then, unless **P** = **NP**, for any $c \in \mathbb{N}$, $c \geqslant 1$, no polynomial time algorithm can compute a solution of WC in any class of graphs such that the difference between its value and the optimal value is bounded above by c; furthermore, if WC($\mathcal{G}'$) is strongly **NP**-complete, then, unless **P** = **NP**, WC($\mathcal{G}'$) cannot be solved neither by a standard nor by a differential fully polynomial time approximation scheme.*

4 The bipartite case and some related cases

4.1 The bipartite graphs

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 WC is given by a bipartite weighted graph G with a positive integer q. Let $\mathrm{WC}(G, q)$ be the following problem: does there exist a coloring $\mathcal{S}$ of G with $\mathrm{val}(\mathcal{S}) \leqslant q$?

Proposition 7. *$\mathrm{WC}(G, q)$ is **NP**-complete in the strong sense even if G is a bipartite graph of maximum degree at most 14.*

Proof (Sketch). We use a reduction from 1-PrExt ([11]): "given a bipartite graph $G = (V, E)$ with $|V| \geqslant 3$ 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 $i = 1, 2, 3$?" 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 $\{v_1, v_2, v_3\} \subseteq L$. We introduce three new nodes u_1, u_2, u_3 in R and edges $[v_i, u_j]$ for $i \neq j$ and $1 \leqslant i, j \leqslant 3$. 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 $\mathrm{WC}(G', 7)$. There exists a coloring $\mathcal{S}$ of G' with $\mathrm{val}(\mathcal{S}') \leqslant 7$ if and only if there exists a 3-coloring (S_1, S_2, S_3) of G with $v_i \in S_i$, $i = 1, 2, 3$. □

As a consequence of Proposition 7, WC is also **NP**-complete if G is a comparability graph (i.e., a graph whose edges can be transitively oriented, see [5]).

Proposition 8. *If* $G = (V, E, w)$ *is a bipartite weighted graph with bivalued weights, then one can construct an optimal* k*-coloring* $\mathcal{S}$ *in polynomial time.*

Proof (Sketch). By Proposition 2, an optimal solution is either a 2- or a 3-coloring. In the former case we can construct it by a greedy algorithm. For the latter case, if any optimal solution is a 3-coloring, then the set $V_{\max}$ of the maximum-weight nodes is stable (if not, there exists an optimal 2-coloring) and $\mathcal{S} = (V_{\max}, L \setminus V_{\max}, R \setminus V_{\max})$. □

We now propose a polynomial time approximation algorithm achieving a constant standard approximation ratio for WC 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) optimally color the remaining uncolored nodes with at most two new colors b and g following the bipartition of G; store the solution obtained during steps (2) and (3); (4) compute a minimum-weight 2-coloring; store the solution obtained; (5) output the smallest between the solutions stored in steps (3) and (4).

As the bicoloring of a of a connected bipartite graph is unique, a minimum-weight 2-coloring is simply the unique bipartition of V. If the graph is not connected, then a minimum-weight 2-coloring can be easily computed by taking care of assigning the same color to all the heaviest color-classes of the connected components of G. In what follows, we denote by $w_{\max}$ (resp., $w_{\min}$) the largest (resp., smallest) node weight.

Proposition 9. *`BIP_WCOLOR` polynomially solves WC in bipartite graphs within standard approximation ratio bounded above by* $4r_w/(3r_w + 2)$*, where* $r_w = w_{\max}/w_{\min}$*. This bound is tight.*

Proof (Sketch). Obviously, the weight of color c equals $w_{\max}$. Suppose now that step (2) stops while a node of weight $w_{\max}/t$, for some $t > 1$, has been encountered. Then, $\text{opt}(G) \geqslant w_{\max} + (w_{\max}/t) + w_{\min}$ (otherwise, the optimal solution for WC on G would be a 2-coloring). On the other hand, $\text{val}_{\texttt{BIP\_WCOLOR}}(G) \leqslant w_{\max}(t + 2)/t$ if the final solution is the one of step (3) and $\text{val}_{\texttt{BIP\_WCOLOR}}(G) \leqslant 2w_{\max}$ if the final solution is the one of step (4). Combination of the expressions above and some algebra show that the common value for both ratios is $4r_w/(3r_w + 2) \leqslant 4/3$. Tightness is shown in [1]. □

In the proof of Proposition 7, one can see that WC is **NP**-complete when $w_{\max} = 4$ and $w_{\min} = 1$. Here, algorithm `BIP_WCOLOR` yields ratio 7/8 and this ratio is the best possible. So the following holds.

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

We now deal with the differential approximation of WC in bipartite graphs. Consider the following algorithm, called `C_SCHEME` in what follows and run it with parameters G and a fixed constant $\epsilon > 0$: (a) rank the nodes of G in non-increasing weight and set $w_i = w(v_i)$, $i = 1, \ldots, n$; (b) set $\eta = \lceil 1/\epsilon \rceil$; set $S_L = \{v_{4\eta+3}, \ldots, v_n\} \cap L$; set $S_R = \{v_{4\eta+3}, \ldots, v_n\} \cap R$; (c) set $\tilde{\mathcal{S}}$ the best partition into stable sets of the nodes $v_1, \ldots, v_{4\eta+2}$; (d) output $\hat{S} = S_L \cup S_R \cup \tilde{\mathcal{S}}$.

Since η is a fixed constant, the whole complexity of `C_SCHEME` is linear in n. Denote now by G' the subgraph of G induced by the node-set $\{v_1, \ldots, v_{4\eta+2}\}$ and recall that $\tilde{\mathcal{S}}$ is optimal for G'.

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

Proof (Sketch). We can easily see that $|\tilde{\mathcal{S}}| \leqslant 2\eta + 2$ and $\mathrm{val}(\tilde{\mathcal{S}}) = \mathrm{opt}(G') \leqslant \mathrm{opt}(G)$ (the relative proof is given in [1]). Then, $\epsilon(\mathrm{worst}(G') - \mathrm{opt}(G')) \geqslant 2w_{4\eta+2}$. Moreover, $\mathrm{opt}(G') \leqslant \mathrm{opt}(G)$. Hence, $\mathrm{val}_{\texttt{C\_SCHEME}}(G) \leqslant \mathrm{opt}(G') + 2w_{4\eta+2} \leqslant (1-\epsilon)\mathrm{opt}(G') + \epsilon\mathrm{worst}(G') \leqslant (1-\epsilon)\mathrm{opt}(G) + \epsilon\mathrm{worst}(G)$. □

4.2 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. In this last case, WC is polynomial ($\alpha(G) \leqslant 2$, cf., Proposition 5).

Proposition 12. *WC is **NP**-complete in the strong sense if G is a split graph.*

Proof (Sketch). The reduction is from the Min-Set-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 $\cup_{j\in J}\overline{C}_j = \overline{S}$?

Let us construct a split 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 $\mathcal{S} = (S_1, \ldots, S_k)$ with $\mathrm{val}(\mathcal{S}) \leq |I|^2 + q$. □

The proof of Proposition 12 shows that the problem is **NP**-complete even if the weights can take only two values. It also follows from this proof that WC(G, q) is **NP**-complete if G is a chordal graph, since a split graph is a chordal graph ([5]).

5 An edge coloring model

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 13. *WC 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 (Sketch). We shall start from the following **NP**-complete problem called 2-SIM ([12]): given a bipartite regular multigraph $H = (V, E)$ and two disjoint (partial) matchings M_1^*, M_2^*, does there exist an edge 3-coloring (M_1, M_2, M_3) of H such that $M_i^* \subseteq M_i$ for $i = 1, 2$?

Replace any 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. 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 $\widehat{\mathcal{M}} = (\widehat{M_1}, \ldots, \widehat{M_k})$ with $\text{val}(\widehat{\mathcal{M}}) = w(\widehat{M_1}) + \ldots + w(\widehat{M_k}) \leqslant 7$ if and only if H has an edge 3-coloring $\mathcal{M} = (M_1, \ldots, M_3)$ with $M_i^* \subseteq M_i$ $(i = 1, 2)$. □

In what follows, we denote by $\text{EWC}(G_k, q)$ the edge coloring version of WC in k-regular bipartite graphs $G_k = (L, R, E)$.

Proposition 14. *EWC is strongly **NP**-complete in k-regular bipartite graphs with $k \geqslant 3$.*

Proof (Sketch). The proof is by induction. For $k = 3$, we use Proposition 13 and the gadget of figure 1 showing how one can transform a cubic bipartite multigraph G to a simple cubic bipartite graph B. Note that in any feasible edge coloring of B, $\{\text{color}(a), \text{color}(b)\} = \{\text{color}(a'), \text{color}(b')\}$.

Suppose that strong **NP**-completeness is true for $k-1$. We use the following reduction from $\text{EWC}(G_{k-1}, q)$ to $\text{EWC}(G_k, 3q)$. Consider a $(k-1)$-regular bipartite graph $G_{k-1} = (L, R, E)$ and denote by w_{k-1} is edge-weight vector. Remark that $|L| = |R|$ and let r_i and l_i be for $i = 1, \ldots, |L|$ 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 $i = 1, \ldots, |L|$ 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 $e \in \{[r_i, l'_i], [l_i, r'_i] : i = 1, \ldots, |L|\}$. Obviously, G_k is k-regular. Then, there exists an edge coloring of weight q in G_{k-1}, iff there exists an edge coloring of weight $3q$ in G_k. □

We now study the special case where edge weights are bivalued.

Proposition 15. *$\text{WC}(L(H), q)$ can be solved in polynomial time if H is bipartite with weights $w(e) \in \{a, b\}$ on the edges.*

Proof (Sketch). In order to simplify the sketch, suppose $a = 1$ and $b = t$. Starting from H, we construct a network N and solve a particular flow problem. 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 $\Delta(t) = \Delta(H)$, then any edge $\Delta(H)$-coloring of H is optimal. Construct

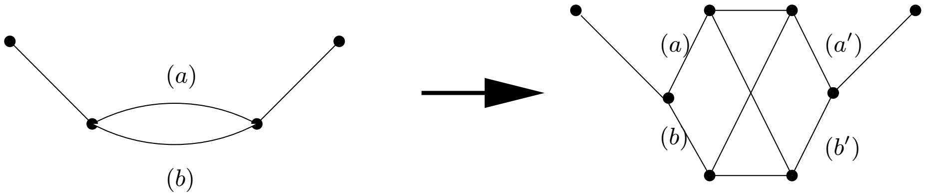

Fig. 1. Transformation of a cubic bipartite multigraph G into a simple cubic bipartite graph B.

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 $\overrightarrow{e}=(u,v)$ with capacity $c(\overrightarrow{e})=1$ and lower bound of flow $l(\overrightarrow{e})=0$; 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 $c(v,t_0)=\Delta(t)-d_{H(t)}(v)$. We have to find the smallest possible r for which $N(r)$ contains a feasible flow. Such an r will give us an edge $(\Delta(H(t))+r)$-coloring $\mathcal{M}$ such that $\mathrm{val}(\mathcal{M})=\Delta(H(t))t+r$. But such a coloring $\mathcal{M}$ may not be of minimum cost. We have to examine also edge k-colorings $\mathcal{M}=(M_1,\ldots,M_k)$ 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 $\Delta(H)-\Delta(H(t))$. □

In [13] it is shown that WC 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 fixed degrees, for any fixed constant. In addition [13] states Proposition 15 for the special case of the line graph of $K_{n,n}$.

We now deal with the approximation of EWC. Remark first that, by König's theorem ([14]), the optimal solution of the (unweighted) edge covering achieves standard approximation ratio Δ for EWC, for any $\Delta\geqslant 3$, 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 $\Delta=3$. We are given a bipartite graph G; 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`, when we assume that the set $E=\{e_1,e_2,\ldots,e_{|E|}\}$ of edges of G is ranked in decreasing weight order and, for any $j\in\{1,\ldots,|E|\}$, we set $E_j=\{e_1,\ldots,e_j\}$: (1) set $M_1^1=M_2^1=\ldots=M_{|E|}^1=\emptyset$; (2) for $i=1$ to $|E|$ do: set $j_0=\min\{j=1,\ldots|E|:M_j^1\cup\{e_i\}$ is a matching$\}$; set $M_{j_0}^1=M_{j_0}^1\cup\{e_i\}$; (3) set $\mathcal{S}_1=(M_1^1,\ldots,M_{r_1}^1)$ the list of the non-empty matchings of $(M_1^1,M_2^1,\ldots,M_{|E|}^1)$; set $k_0=\max\{j:$

$G[E_j]$ has maximum degree at most 2}; (4) for $\ell = 2$ to k_0 do: (4a) compute an optimal 2-coloring (M_1^ℓ, M_2^ℓ) for $G[E_\ell]$; (4b) complete (M_1^ℓ, M_2^ℓ) by running steps (1) to (2) in $G \setminus G[E_\ell]$; (4c) set $\mathcal{S}_\ell = (M_1^\ell, M_2^\ell, \ldots, M_{r_\ell}^\ell)$ the edge coloring computed in steps (4a) and (4b); (5) output $\mathcal{S} = \text{argmin}\{\text{val}(\mathcal{S}_\ell) : \ell = 1, \ldots, k_0\}$.

Any set $\mathcal{S}_\ell$ computed by algorithm `EW_COLOR` verify Corollary 1; hence, $r_\ell \leqslant 5$.

Proposition 16. *`EW_COLOR` achieves standard approximation ratio 5/3 in polynomial time. This ratio is tight.*

Proof (Sketch). Following the remark just above on the value of r_ℓ, one can set $\mathcal{S}_\ell = (M_1^\ell, \ldots, M_5^\ell)$, (some of the M_i^ℓ, $i = 1, \ldots, 5$ may be empty). Fix an optimal solution $\mathcal{S}^*$ and denote by M_1^*, M_2^*, M_3^* the three largest matchings of $\mathcal{S}^*$. Set $i_3^* = \min\{j : e_j \in M_3^*\}$. By construction, $G[E_{i_3^*-1}]$ has maximum degree at most 2 and hence $w(M_1^{i_3^*-1}) + w(M_2^{i_3^*-1}) \leqslant w(M_1^*) + w(M_2^*)$ and $w(M_3^{i_3^*-1}) + w(M_4^{i_3^*-1}) + w(M_5^{i_3^*-1}) \leqslant 3w(M_3^*)$. We so finally obtain $\text{val}_{\text{EW\_COLOR}}(\mathcal{S}) \leqslant \text{val}(\mathcal{S}_{i_3^*-1}) \leqslant 5\text{opt}(G)/3$. The proof of the tightness is shown in [1]. □

The same analysis as the one in the proof of Proposition 16 concludes that EWC is approximable within standard approximation ratio bounded above by $(2\Delta - 1)/3$, for any $\Delta \geqslant 3$.

Proposition 17. *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 (Sketch). From the proofs of Propositions 13 and 14, where, in the latter, we change cost $w_k(e)$ to $2\max\{w_{k-1}(e)\}$ (this case remains **NP**-complete), one can see that EWC in regular bipartite graphs of degree at least k is **NP**-complete whenever the optimal value of the instance is at most $2^k - 1$. □

We now give a differential approximation result for EWC. 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: set $k = \lceil 1/\epsilon \rceil$; rank the edges in E in decreasing-weight order; set $E = \{e_1, \ldots, e_{|E|}\}$; set $E' = \{e_1, e_2, \ldots, e_{3k+5}\}$; optimally color $G[E']$ and greedily complete the edge coloring of step obtained in order to color E with at most three colors (in other words, omit weights and color the unweighted version of G).

Proposition 18. *Algorithm `EC_SCHEME` is a polynomial time differential approximation scheme for EWC.*

Proof (Sketch). Let $(M_1^*, \ldots, M_r^*)$ be an optimal solution of $G[E']$. By Corollary 1, we can suppose $r \leqslant 5$. So, $\text{worst}(G[E']) - \text{opt}(G[E']) \geqslant 3kw(e_{3k+5}) \geqslant 3w(e_{3k+5})/\epsilon$, and $\text{val}_{\texttt{EC\_SCHEME}}(G) \leqslant \text{opt}(G[E']) + w(e_{3k+6}) + w(e_{3k+7}) + w(e_{3k+8})$. After some algebra and taking into account that edges in E are ranked in decreasing weight order, $\text{val}_{\texttt{EC\_SCHEME}}(G) \leqslant (1-\epsilon)\text{opt}(G) + \epsilon\text{worst}(G)$. □

One can easily see that the result of Proposition 18 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 [15]; 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'. 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 [6], a graph G is a cograph if and only if for all induced subgraphs G' of G the `GSC` based upon any order θ gives a coloring of G' in $\chi(G')$ colors.

Lemma 2. *If $G = (V, E, w)$ is a weighted cograph, then all optimal colorings $\mathcal{S} = (S_1, \ldots, S_k)$ satisfy $k = \chi(G)$.*

Proof (Sketch). Assume there exists an optimal k'-coloring $\mathcal{S}' = (S'_1, \ldots, S'_{k'})$ with $k' > \chi(G)$. 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 $\mathcal{S} = (S_1, \ldots, S_k)$ with $k = \chi(G)$ (we have ordered $\mathcal{S}$ and $\mathcal{S}'$ by non-increasing costs). Each node $v \in S'_j$ will satisfy $v \in S_i$ with $i \leqslant j$ after applying `GSC`. Thus, we have $w(S'_i \cup \{v\}) = w(S'_i)$ and $S_{k+1} = \emptyset$, a contradiction. □

We can now show that there is a polynomial algorithm which constructs an optimal k-coloring $\mathcal{S}$; such a result can be expected from graphs like cographs for which several generally difficult coloring problems are easier ([16]).

Proposition 19. *Let $G = (V, E, w)$ be a a weighted cograph. Then, the k-coloring $\mathcal{S}$, constructed by the `GSC` algorithm based upon any order θ where $u < v$ (u before v in θ) implies $w(u) \geqslant w(v)$, is optimal.*

Proof (Sketch). Let $t_1 > t_2 > \ldots > t_r$ be the values taken by the weights $w(v)$ in G. Every k-coloring $\mathcal{S} = (S_1, \ldots, S_k)$ of G with $k = \chi(G)$ and $w(S_1) \geqslant w(S_2) \geqslant \ldots \geqslant w(S_k)$ satisfies: $w(S_i) \geqslant \max\{t_s : \omega(G(s)) \geqslant i\}$ 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 $w(v) \geqslant t_s$. 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 $w(S_i) \geqslant t_2$ and generally the first $\omega(G(s))$ sets S_i will have $w(S_i) \geqslant t_s$.

Now consider the k-coloring $\overline{\mathcal{S}} = (\overline{S}_1, \ldots, \overline{S}_k)$ 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, 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 $i = 1, \ldots, p(s)$. So, $\overline{\mathcal{S}}$ satisfies $w(\overline{S}_i) = \max\{t_s : \omega(G(s)) \geqslant i\}$ and thus $\overline{\mathcal{S}}$ is an optimal coloring. □

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

Proposition 19 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 $k = \chi(G)$.

References

1. Demange, M., de Werra, D., Monnot, J., Paschos, V.T.: Time slot scheduling of compatible jobs. Cahier du LAMSADE 182, LAMSADE, Université Paris-Dauphine (2001) Available on www_address: `http://www.lamsade.dauphine.fr/cahdoc.html#cahiers`.
2. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: a review. European J. Oper. Res. **120** (2000) 228–249
3. Potts, C.N., Strusevich, V.A., Tautenhahn, T.: Scheduling batches with simultaneous job processing for two-machine shop problems. J. of Scheduling **4** (2001) 25–51
4. Boudhar, M., Finke, G.: Scheduling on a batch machine with job compatibilities. JORBEL (2001) To appear.
5. Berge, C.: Graphs and hypergraphs. North Holland, Amsterdam (1973)
6. de Werra, D.: Heuristics for graph coloring. Computing **7** (1990) 191–208
7. Simon, H.U.: On approximate solutions for combinatorial optimization problems. SIAM J. Disc. Math. **3** (1990) 294–310
8. Johnson, D.S.: Approximation algorithms for combinatorial problems. J. Comput. System Sci. **9** (1974) 256–278
9. Alfandari, L., Paschos, V.T.: Master-slave strategy and polynomial approximation. Comput. Opti. Appl. **16** (2000) 231–245
10. Boppana, B.B., Halldórsson, M.M.: Approximating maximum independent sets by excluding subgraphs. BIT **32** (1992) 180–196
11. Bodlaender, H.L., Jansen, K., Woeginger, G.J.: Scheduling with incompatible jobs. Discrete Appl. Math. **55** (1994) 219–232
12. de Werra, D., Erschler, J.: Open shop scheduling with some additional constraints. Graphs and Combinatorics **12** (1996) 81–93
13. Rendl, F.: On the complexity of decomposing matrices arising in satellite communication. Oper. Res. Lett. **4** (1985) 5–8
14. König, D.: Über graphen und iher anwendung auf determinantentheorie und mengenlehre. Math. Ann. **77** (1916) 453–465
15. Chvàtal, V.: Perfectly ordered graphs. In Berge, C., Chvàtal, V., eds.: Topics on Perfect Graphs. (Volume 21 of Annals of Discrete Math.) 253–277
16. Jansen, K., Scheffler, P.: Generalized coloring for tree-like graphs. Discrete Appl. Math. **75** (1997) 135–155