A NOTE ON POLYNOMIAL ALGORITHM FOR COST COLORING OF BIPARTITE GRAPHS WITH $\Delta \leq 4$

Krzysztof Giaro and Marek Kubale

Gdańsk University of Technology, ETI Faculty
Gabriela Narutowicza 11/12
80–233 Gdańsk, Poland

e-mail: giaro@pg.edu.pl
kubale@eti.pg.gda.pl

Abstract

In the note we consider vertex coloring of a graph in which each color has an associated cost which is incurred each time the color is assigned to a vertex. The cost of coloring is the sum of costs incurred at each vertex. We show that the minimum cost coloring problem for n-vertex bipartite graph of degree $\Delta \leq 4$ can be solved in $O(n^2)$ time. This extends Jansen's result [K. Jansen, *The optimum cost chromatic partition problem*, in: Proc. CIAC'97, Lecture Notes in Comput. Sci. 1203 (1997) 25–36] for paths and cycles to subgraphs of biquartic graphs.

Keywords: bipartite graph, chromatic sum, cost coloring, NP-completeness, polynomial algorithm.

2010 Mathematics Subject Classification: 05C15.

1. Introduction

Graph coloring has many applications in scheduling. One of them is constructing class-teacher timetables. This is so because assigning time periods to classes may be viewed as proper coloring of the associated graph G with colors representing times, since every two adjacent vertices (classes) have to receive different colors to avoid conflicts. By finding the optimal schedule we usually mean a proper coloring of G with a minimum number of colors. But what we really want to minimize is the cost. Different time periods might have various corresponding costs. For example, the morning hours might be less expensive than afternoon hours (due to the cost of electricity) and much less expensive than the evening

ones (due to overtime for janitors and security personnel). The cost of each time period is the product of its weight and the number of vertices assigned to the corresponding color. Thus, when optimizing the schedule, we seek for the cost coloring of the underlying graph. We give now the basic formal definitions.

Let $G = (V, E)$ be a graph with $n = |V|$ vertices and $m = |E|$ edges. A (proper) *vertex coloring* of graph G is an assignment of colors (natural numbers) to the vertices of G in which any two adjacent vertices are assigned different colors. In other words, a k-coloring of G is a partition $(C_1, C_2, \ldots, C_k)$ of V into independent sets (some of them may be empty). We assume that for any palette of n colors we have an associated sequence $W = (w_1, \ldots, w_n)$ of color weights (costs), where w_i is a positive rational number associated with color $i = 1, \ldots, n$. Each time the color i is used on a vertex the weight w_i is accrued. Our goal is to color graph G so that a minimum total cost is attained. The *cost-chromatic number* with respect to W, denoted by $\chi_W(G)$, is the minimum number of colors necessary to produce a minimum cost coloring of G, i.e., $\Sigma_W(G) = \min_c \Sigma_i |C_i| w_i$. In the special case of cost coloring when W is the sequence of the first n natural numbers, we write $W = N = (1, \ldots, n)$. Such a minimum cost of coloring is called the *chromatic sum* of G. One can observe that the cost coloring problem is NP-hard since the ordinary chromatic number problem is NP-hard. In addition to the terminology of graph coloring we will need the notion of a cut between a source s and sink t in a digraph D. Given a digraph $D = (V, A)$, an *s–t cut* (S, T) is a partition of the vertices of D such that $s \in S$ and $t \in T$. If arcs of D have numerical weights (capacities) defined by a function c, then the *capacity* of this cut is $\sum_{e \in A \cap (S \times T)} c(e)$.

The problem of cost coloring was introduced by Supowit [12]. The first polynomial algorithm was given by Jansen [3]. He considered complements of bipartite graphs and obtained an algorithm for optimal cost coloring of such graphs in $\mathrm{O}(m\sqrt{n})$ time. Next polynomial algorithms were given by Kroon *et al.* [7]. They gave a linear time algorithm for trees and a polynomial time algorithm for interval graphs in case when there are only two different values for the color weights. The algorithm for trees was then generalized to graphs with bounded treewidth [4] and to graphs with bounded cyclicity [2]. The former class of graphs admits an algorithm running in time $\mathrm{O}(n \log^{k+1} n)$, where k is the treewidth of G, while the latter admits an algorithm taking time $\mathrm{O}(n\Delta^{3+2l} \log n)$, where l is the cyclomatic number of G and Δ denotes the maximum vertex degree. Last but not least, there are known polynomial results for the model in which we color the edges rather than vertices. For example, Zhou and Nishizeki [13] designed an algorithm for cost edge-coloring of trees in $\mathrm{O}(n\Delta^2)$ time and Cardinal *et al.* [1] considered cost edge-coloring of trees and multicycles in $\mathrm{O}(n\Delta)$ time. This means that the cost coloring problem for line graphs of trees and multicycles can also be solved in polynomial time.

In this note we focus on coloring of bipartite graphs $G = (V_1 \cup V_2, E)$. Kubicka [9] showed that for any integer $k \geq 2$ there is a bipartite graph G (a tree in fact) with a sufficiently high vertex degree such that $\chi_N(G) = k$. On the other hand, Kosowski [8] showed that for any bipartite graph with $\Delta \leq 4$ we have $\chi_N(G) \leq 3$. The smallest bipartite graph G for which $\chi_W(G) = 3$ is depicted in Figure 1. In this note, among others, we generalize this result to any sequence of weights W.
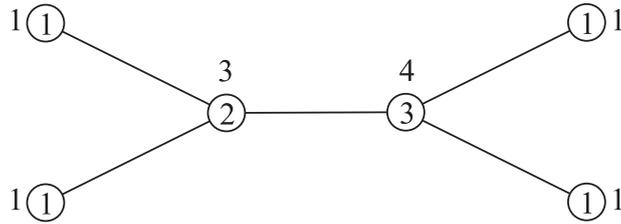

Figure 1. An example of minimum cost coloring of graph G for $W = (1, 3, 4, \dots)$; $\chi_W(G) = 3$, $\Sigma_W(G) = 11$.

The main result of this paper is showing a new polynomial solvable case of the cost coloring problem. More precisely, in the next section we generalize the sum coloring approach given in Małafiejski *et al.* [10] and obtain an $O(n^2)$ algorithm for cost coloring of bipartite graphs of degree at most 4. The algorithm is best possible in the sense that the same problem but for bipartite graphs of degree 5 becomes NP-hard [10].

2. Algorithm

In the following, by an *improper 3-coloring* of graph G we mean any 3-coloring, i.e., a partition (C_1, C_2, C_3) of its vertices such that sets C_1, C_2 are independent but vertices having color 3 may be adjacent. If G has color weights $w_1 \leq w_2 \leq w_3$ then the cost of such an improper 3-coloring (C_1, C_2, C_3) is $w_1|C_1| + w_2|C_2| + w_3|C_3|$. We have the following.

Theorem 1. *Given color weights $w_1 \leq w_2 \leq w_3$, the minimum cost improper 3-coloring of any bipartite graph $G = (V_1 \cup V_2, E)$ can be obtained in $O(mn)$ time.*

Proof. We can assume that $w_1 < w_2$ since if $w_1 = w_2$ then an optimal improper 3-coloring is $(C_1, C_2, \emptyset)$ and it is determined by 2-coloring of G. Let $l > 0$ and let V_1^*, V_2^* be the disjoint copies of $V_1(G), V_2(G)$. By v^* we denote an image of vertex $v \in V(G)$ under bijection $h \colon V_1 \cup V_2 \mapsto V_1^* \cup V_2^*$, i.e., $h(v) = v^*(h^{-1}(v^*) = v)$, similarly $h(V_i) = V_i^*$ for $i = 1, 2$. Now we define an auxiliary directed graph $D = (V(D), A(D))$ with weights on the arcs as shown in Figure 2.
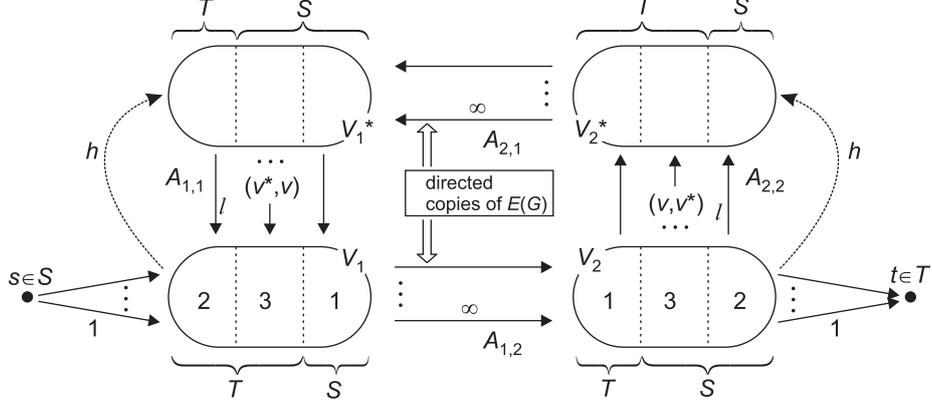
Figure 2. Digraph D with specified sets of vertices, arcs, and their weights.

$$V(D) = V(G^*) \cup V(G) \cup \{s\} \cup \{t\},$$
$$A(D) = A_{1,2} \cup A_{2,1} \cup A_{s,1} \cup A_{2,t} \cup A_{1,1} \cup A_{2,2},$$
$$w(e) = \begin{cases} 1 & \text{if } e \in A_{s,1} \cup A_{2,t}, \\ l & \text{if } e \in A_{1,1} \cup A_{2,2}, \\ \infty & \text{if } e \in A_{1,2} \cup A_{2,1}, \end{cases}$$

where

$$A_{1,2} = \{(v_1, v_2) : v_1 \in V_1 \wedge v_2 \in V_2 \wedge \{v_1, v_2\} \in E(G)\},$$
$$A_{2,1} = \{(v_2^*, v_1^*) : v_1 \in V_1 \wedge v_2 \in V_2 \wedge \{v_1, v_2\} \in E(G)\},$$
$$A_{s,1} = \{s\} \times V_1,$$
$$A_{2,t} = V_2 \times \{t\},$$
$$A_{1,1} = \{(v_1^*, v_1) : v_1 \in V_1\},$$
$$A_{2,2} = \{(v_2, v_2^*) : v_2 \in V_2\}.$$

With a given improper 3-coloring (C_1, C_2, C_3) of G we associate an s–t cut (S, T) as follows (see Figure 2).

$$S = \{s\} \cup (C_1 \cap V_1) \cup h(C_1 \cap V_1) \cup (V_2 \setminus C_1) \cup h(V_2 \cap C_2) \cup h(V_1 \cap C_3),$$

$$T = \{t\} \cup (C_1 \cap V_2) \cup h(C_1 \cap V_2) \cup (V_1 \setminus C_1) \cup h(V_1 \cap C_2) \cup h(V_2 \cap C_3).$$

Since sets C_1 and C_2 are independent, the weights on arcs $A(D) \cap (S \times T)$ are all less than infinity. More precisely, there are arcs of weight 1, namely, $\{s\} \times ((V_1 \cap C_2) \cup (V_1 \cap C_3))$ and $((V_2 \cap C_2) \cup (V_2 \cap C_3)) \times \{t\}$, and of weight l, namely $\{(v^*, v) : v \in V_1 \cap C_3\}$ and $\{(v, v^*) : v \in V_2 \cap C_3\}$.

Thus, the capacity of such an s–t cut equals

(1) $$|C_2| + (l+1)|C_3|.$$

On the other hand, any s–t cut (S,T) of finite capacity can be transformed to an s–t cut associated with some improper 3-coloring of G. We may assume that $T \cap V_1^* \subseteq h(T \cap V_1)$, because even if some $v^* \in T \cap V_1^*$ fulfills $v \in S \cap V_1$, then we can move it to partition S without increasing the capacity of the cut. Analogously, we can assume that $h(T \cap V_2) \subseteq T \cap V_2^*$, because any $v^* \in S \cap V_2^*$ such that $v \in T \cap V_2$ can be moved to partition T. In that case (S,T) is a cut associated with improper 3-coloring of the form

(2)
$$C_1 = (S \cap V_1) \cup (T \cap V_2),$$
$$C_2 = h^{-1}((S \cap V_2^*) \cup (T \cap V_1^*)),$$
$$C_3 = V(G) \setminus (C_1 \cup C_2),$$

because the finiteness of the capacity for (S,T) implies $A(D) \cap ((S \cap V_1) \times (T \cap V_2)) = \emptyset$ and $A(D) \cap ((S \cap V_2^*) \times (T \cap V_1^*)) = \emptyset$, so C_1, C_2 are in fact independent sets.

Thus, by finding a minimum s–t cut and transforming it to the associated s–t cut, we obtain an improper 3-coloring (C_1, C_2, C_3) of G according to (2), which is the cheapest for weights $w_1 = 0, w_2 = 1, w_3 = 1 + l$ (cf. (1)). However, any improper coloring which is optimal for weights $w_1 < w_2 \le w_3$ is also optimal for weights cw_1, cw_2, cw_3, where $c > 0$, as well as for weights w_1+c, w_2+c, w_3+c, for any constant c. This is so because adding to all weights a constant c increases the cost of improper 3-coloring by cn. Assuming in our construction that $l = \frac{w_3 - w_2}{w_2 - w_1}$ one can easily see that we can obtain in polynomial time a coloring (C_1, C_2, C_3), which is optimal for color weights $0, 1, 1 + l = \frac{w_3 - w_1}{w_2 - w_1}$ and, consequently, for color weights $0, w_2 - w_1, w_3 - w_1$ and w_1, w_2, w_3.

Finding a minimum cut in a network D can be done in $O(|A(D)||V(D)|)$ time (see King $et\ al.$ [6], Orlin [11]). Since in our case $|A(D)| = 2|E(G)| + 2|V(G)| = 2m + 2n = O(m+n)$ and $|V(D)| = 2|V(G)| + 2 = 2n + 2 = O(n)$, the complexity $O(mn)$ follows. ∎

Corollary 2. *Given color weights $(w_1, w_2, \ldots, w_n)$ and a bipartite graph G on n vertices with $\Delta \le 4$, the cost coloring problem can be solved in time $O(n^2)$. In this case there always exists an optimal solution using at most three cheapest colors only.*

Proof. Without loss of generality, we assume that $w_1 \le w_2 \le \cdots \le w_n$. An algorithm from [8] takes $O(n^2)$ time to transform any improper 3-coloring of G with $\Delta \le 4$ into a proper one without increasing its cost. By applying it to an optimal improper 3-coloring, whose cost is at most the cost of any proper

coloring, we arrive at the cheapest possible 3-coloring of G. Since the construction described in the proof of Theorem 1 can also be performed in $O(n^2)$ time for the graphs under consideration, the thesis of the corollary follows.   ∎

3.   Final Remarks

The reader may wonder what if a bipartite graph G is of degree $\Delta \geq 5$. We know that the cost coloring problem is then NP-hard. However, the following simple algorithm is near-optimal.

1. Let S_1 be a cheapest (C_1, C_2)-coloring of G.

2. Let S_2 be a 3-coloring obtained by assigning color C_1 to a largest independent set I in G and by 2-coloring of $V(G) \setminus I$ in the best way using C_2, C_3.

3. Return solution S as the better of S_1 and S_2.

Jansen [5] has proved that the above algorithm runs in $O\big((m+n)n^{0.5}\big)$ time to produce a solution of value at most $\sqrt{n}$ times $\Sigma_W(G)$. Moreover, his algorithm is asymptotically the best possible since the cost coloring problem is hard to approximate with ratio $O\big(n^{0.5-\epsilon}\big)$ for any $\epsilon > 0$, unless $P = NP$ [5].

References

[1] J. Cardinal, V. Ravelomanana and M. Valencia-Pabon, *Minimum sum edge colorings of multicycles*, Discrete Appl. Math. **158** (2010) 1216–1223.
doi:10.1016/j.dam.2009.04.020

[2] K. Giaro and M. Kubale, *Efficient list cost coloring of vertices and/or edges of bounded cyclicity graphs*, Discuss. Math. Graph Theory **29** (2009) 361–376.
doi:10.7151/dmgt.1452

[3] K. Jansen, *Complexity results for the optimum cost chromatic partition problem*, Forschungsbericht, Trier University (1996) 96–41.

[4] K. Jansen, *The optimum cost chromatic partition problem*, in: Proc. CIAC'97, Lecture Notes in Comput. Sci. **1203** (1997) 25–36.
doi:10.1007/3-540-62592-5_58

[5] K. Jansen, *Approximation results for the optimum cost chromatic partition problem*, J. Algorithms **34** (2000) 54–89.
doi:10.1006/jagm.1999.1022

[6] V. King, S. Rao and R. Tarjan, *A faster deterministic maximum flow algorithm*, J. Algorithms **17** (1994) 447–474.
doi:10.1006/jagm.1994.1044

[7] L.G. Kroon, A. Sen, H. Deng and A. Roy, *The optimal cost chromatic partition problem for trees and interval graphs*, in: Proc. WG'96 IWGTCCS, Lecture Notes in Comput. Sci. **1197** (1997) 279–292.
doi:10.1007/3-540-62559-3_23

[8]   A. Kosowski, *A note on the strength and minimum color sum of bipartite graphs*, Discrete Appl. Math. **157** (2009) 2552–2554.
doi:10.1016/j.dam.2009.03.008

[9]   E. Kubicka, The Chromatic Sum of a Graph (Ph.D. Thesis, Western Michigan University, Kalamazoo, 1989).

[10] M. Małafiejski, K. Giaro, R. Janczewski and M. Kubale, *A 27/26-approximation algorithm for the chromatic sum coloring of bipartite graphs*, in: Proc. APPROX'02, Lecture Notes in Comput. Sci. **2462** (2002) 136–146.
doi:10.1007/3-540-45753-4_13

[11] J.B. Orlin, *Max flows in* O(nm) *time, or better*, in: Proc. STOC'13 (ACM, New York, 2013) 765–774.
doi:10.1145/2488608.2488705

[12] K. Supowit, *Finding a maximum planar subset of a set of nets in a channel*, IEEE Trans. Comput.-Aided Design **6** (1987) 93–94.
doi:10.1109/TCAD.1987.1270250

[13] X. Zhou and T. Nishizeki, *Algorithm for the cost edge-coloring of trees*, in: Proc. COCOON'01, Lecture Notes in Comput. Sci. **2108** (2001) 288–297.
doi:10.1007/3-540-44679-6_32