Algorithm for the Cost Edge-Coloring of Trees

Xiao Zhou* and Takao Nishizeki**

Graduate School of Information Sciences, Tohoku University
Aoba-yama 05, Sendai, 980-8579, Japan.

Abstract. Let C be a set of colors, and let ω be a cost function which assigns a real number $\omega(c)$ to each color c in C. An edge-coloring of a graph G is to color all the edges of G so that any two adjacent edges are colored with different colors. In this paper we give an efficient algorithm to find an optimal edge-coloring of a given tree T, that is, an edge-coloring f of T such that the sum of costs $\omega(f(e))$ of colors $f(e)$ assigned to all edges e is minimum among all edge-colorings of T. The algorithm takes time $O(n\Delta^2)$ if n is the number of vertices and Δ is the maximum degree of T.
Keywords: cost edge-coloring, tree, bipartite graph, matching

1 Introduction

A *vertex-coloring of a graph* $G = (V, E)$ is to color all the vertices of G so that any two adjacent vertices are colored with different colors. We denote by n the number of vertices in G. Let $C = \{c_1, c_2, \cdots, c_m\}$ be a set of colors, and assume that $m = |C|$ is sufficiently large, say $m = n$. Let $\omega : C \to \mathbf{R}$ be a cost function which assigns a real number $\omega(c) \in \mathbf{R}$ to each color $c \in C$. The *cost vertex-coloring problem* is to find a vertex-coloring $f : V \to C$ of G such that $\sum_{v \in V} \omega(f(v))$ is as small as possible. One can observe that the cost vertex-coloring problem is NP-hard since the ordinary vertex-coloring problem is NP-hard. The cost vertex-coloring problem remains NP-hard for interval graphs, but the problem can be solved in linear time for trees [10]. The "vertex-chromatic sum problem" in [11, 12] is merely the cost vertex-coloring problem for a special case where $\omega(c_i) = i$ for each index $i \geq 1$. Jansen gives a polynomial-time algorithm to solve the vertex-chromatic sum problem for graphs of bounded treewidth [6].

In this paper we deal with the cost edge-coloring problem. An *edge-coloring* $f : E \to C$ *of a graph* $G = (V, E)$ is to color all the edges of G so that any two adjacent edges are

* E-Mail: zhou@ecei.tohoku.ac.jp
** E-Mail: nishi@ecei.tohoku.ac.jp

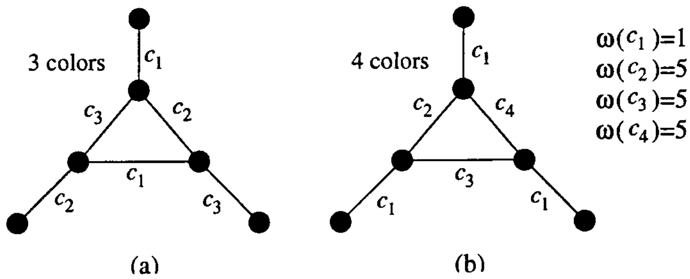

Fig. 1. (a) Edge-coloring with $\chi'(G)$ colors and (b) optimal edge-coloring with $\chi'(G)+1$ colors.

colored with different colors. The minimum number of colors required for edge-colorings of G is called the *chromatic index*, and is denoted by $\chi'(G)$. An ordinary edge-coloring problem is to compute the chromatic index $\chi'(G)$ of a given graph G and find an edge-coloring of G with $\chi'(G)$ colors. On the other hand, the *cost edge-coloring problem* is to find an *optimal edge-coloring* of G, that is, an edge-coloring f such that $\sum_{e\in E}\omega(f(e))$ is minimum among all edge-colorings of G. An optimal edge-coloring does not always use the minimum number $\chi'(G)$ of colors. For example, suppose that $\omega(c_1)=1$ and $\omega(c_i)=5$ for each index $i\geq 2$, then the graph G with $\chi'(G)=3$ in Figure 1(a) can be uniquely colored with the three cheapest colors c_1, c_2 and c_3 as in Figure 1(a), but the edge-coloring is not optimal; an optimal edge-coloring of G uses the four cheapest colors c_1, c_2, c_3 and c_4 as depicted in Figure 1(b). However, it is known that any simple graph G has an optimal edge-coloring using $\Delta(G)$ or $\Delta(G)+1$ colors, and any bipartite graph G and hence any tree has an optimal edge-coloring using $\Delta(G)$ $(=\chi'(G))$ colors, where $\Delta(G)$ is the maximum degree of G [5]. The *edge-chromatic sum problem* introduced by Giaro and Kubale [4] is merely the cost edge-coloring problem for a special case where $\omega(c_i)=i$ for each index $i\geq 1$. The cost edge-coloring problem has a natural application in scheduling theory. Consider the scheduling of biprocessor tasks of unit execution time on dedicated machines. An example of such tasks is the file transfer problem in a computer network in which each file engages two corresponding nodes, sender and receiver, simultaneously [3]. Another example is the biprocessor diagnostic problem in which links execute concurrently the same test for a fault tolerant multiprocessor system [7]. These problems can be modeled by a graph G in which machines correspond to the vertices and tasks correspond to the edges. An edge-coloring of G corresponds to a schedule, where the edges colored with color $c_i \in C$ represent the collection of tasks that are executed in the ith time slot. For each i, if a task is executed in the ith time slot, then it takes the cost $\omega(c_i)$. The goal is to find a schedule that minimizes the total cost. This corresponds to the cost edge-coloring problem.

One can observe that the cost edge-coloring problem is NP-hard since the ordinary edge-coloring problem is NP-hard. Furthermore, the cost edge-coloring problem remains NP-hard for bipartite graph, because the edge-chromatic sum problem is NP-hard for bipartite graphs [4]. When restricted to trees, many NP-hard problems can be efficiently solved in polynomial time [1, 2, 14, 15]. Giaro and Kubale recently give an algorithm to solve the edge-chromatic sum problem for trees T in time $O(n\Delta^{3.5}\log n)$, where n is the number of vertices and Δ is the maximum degree of T [4].

In this paper we give an algorithm to solve the cost edge-coloring problem for trees T in time $O(n\Delta^2)$. The algorithm hence runs in linear time if Δ is a fixed constant. The algorithm takes time $O(n\Delta^{1.5}\log(nN_\omega))$ if all the color costs $\omega(c)$ are integers in the range $[-N_\omega, N_\omega]$. The edge-chromatic sum problem is a special case of the cost edge-coloring problem, in which $N_\omega = \Delta$ for trees. Our algorithm can therefore solve the edge-chromatic sum problem for trees in time $O(n\Delta^{1.5}\log n)$. The time-complexity $O(n\Delta^{1.5}\log n)$ is better than the time-complexity $O(n\Delta^{3.5}\log n)$ of Giaro and Kubale's algorithm [4]. An early version of the paper was presented at [16].

2 Preliminaries

In this section we define some terms and present easy observations. Let $T = (V, E)$ denote a tree with vertex set V and edge set E. We will use notions such as *root, child, descendant* and *leaf* in their usual meaning. We denote by n the number of vertices in T. T is a "free tree," but we choose for the sake of convenience an arbitrary vertex as a root r, and regard T as a "rooted tree." The *degree* $deg(v)$ of vertex v is the number of edges incident to v. We denote the maximum degree of T by $\Delta(T)$ or simply by Δ. We denote by $d(v)$ the number of children of v in T. Then $d(v) = deg(v)$ if $v = r$, and $d(v) = deg(v) - 1$ otherwise. An edge joining vertices u and v is denoted by (u, v). For a vertex v of T, the subtree of T induced by v and all descendants of v is called *the subtree of* T *rooted at* v, and is denoted by T_v. Clearly, $T = T_r$ for the root r.

We assume for the notational sake of convenience that the color set C contains n colors, and write $C = \{c_1, c_2, \cdots, c_n\}$. An *edge-coloring* $f : E \to C$ of a tree T is to color all edges of T by colors in C so that any two adjacent edges are colored with different colors. Let $\omega : C \to \mathbf{R}$, where $\mathbf{R}$ is the set of real numbers. One may assume with loss of generality that ω is non-decreasing, that is, $\omega(c_i) \le \omega(c_{i+1})$ for any index i, $1 \le i < n$. The *cost* $\omega(f)$ *of an edge-coloring* f of a tree $T = (V, E)$ is defined as follows:

$$\omega(f) = \sum_{e \in E} \omega(f(e)).$$

An edge-coloring f of T is called an *optimal* one if $\omega(f)$ is minimum among all edge-colorings of T. The *cost edge-coloring problem* is to find an optimal edge-coloring of a given tree T. The cost of an optimal edge-coloring of tree T is called the *minimum cost of tree* T, and is denoted by $\omega(T)$.

Let f be an edge-coloring of T. For each vertex v of T, let $C(f, v)$ be the set of all colors that are assigned to edges incident to v, that is,

$$C(f, v) = \{f(e) \mid e \text{ is an edge incident to } v \text{ in } T\}.$$

We say that a color $c \in C$ is *missing at* v if $c \notin C(f, v)$. Let $\text{Miss}(f, v)$ be the set of all colors missing at v, that is,

$$\text{Miss}(f, v) = C - C(f, v).$$

Interchanging colors in an alternating path is one of the standard techniques of an ordinary edge-coloring, which we also use in the paper. Let f be an edge-coloring of a tree $T = (V, E)$, let c_i and c_j be any two colors in C, and let $T(c_i, c_j)$ be the subgraph of T induced by all the edges colored with c_i and c_j. Since T is a tree, each connected component of $T(c_i, c_j)$ is a path of length one or more, whose edges are colored alternately with c_i and c_j. We call such a path a *c_ic_j-alternating path.* A vertex $v \in V$ is an end of a c_ic_j-alternating path if and only if exactly one of c_i and c_j is missing at v. If v is an end of a c_ic_j-alternating path, then the path is denoted by $P(v; c_i, c_j)$. The set of all edges in $P(v; c_i, c_j)$ is sometimes simply denoted by $P(v; c_i, c_j)$. Interchanging colors c_i and c_j in $P(v; c_i, c_j)$, one can obtain another edge-coloring f' of T. We then have the following lemma on f'.

Lemma 1. *If $c_i \in \text{Miss}(f,v)$, $c_j \in C(f,v)$, and $i < j$, then $\omega(f') \le \omega(f)$, $c_i \in C(f',v)$, $c_j \in \text{Miss}(f',v)$, and $C(f',v) \cup \{c_j\} = C(f,v) \cup \{c_i\}$.*

Proof. The colors of all edges assigned by f are the same as those by f' except for the edges in $P(v; c_i, c_j)$. The first edge in $P(v; c_i, c_j)$ is colored with c_j by f, but it is colored with c_i by f'. We therefore have $c_i \in C(f',v)$, $c_j \in \text{Miss}(f',v)$, and $C(f',v) \cup \{c_j\} = C(f,v) \cup \{c_i\}$. Since $\omega(c_i) \le \omega(c_j)$ and $P(v; c_i, c_j)$ starts with an edge colored with c_j, we have

$$\sum_{e \in P(v; c_i, c_j)} \omega(f'(e)) \le \sum_{e \in P(v; c_i, c_j)} \omega(f(e)).$$

We thus have $\omega(f') \le \omega(f)$. $\mathcal{Q.E.D.}$

3 Algorithm

In this section we prove the following theorem.

Theorem 1. *An optimal edge-coloring of a tree T can be found in time $O(n\Delta^2)$ where n is the number of vertices and Δ is the maximum degree of T.*

In the remainder of this setion we prove Theorem 1. In Section 3.1 we give an algorithm to compute the minimum cost $\omega(T)$ of a given tree T, and in Section 3.2 we give an algorithm to find an optimal edge-coloring of T.

3.1 Computing the minimum cost $\omega(T)$

A "dynamic programming method" is a standard one to solve a combinatorial problem on trees. We also use it, and compute the minimum cost $\omega(T)$ of a given tree T by the "bottom-up tree computation." Let v be an internal vertex of T, and let $v_1, v_2, \cdots, v_{d(v)}$ be the children of v in T. (See Figure 2.) Then one can observe that the minimum cost $\omega(T_v)$ of the subtree T_v rooted at v cannot be computed directly from the minimum costs $\omega(T_{v_j})$ of all the subtrees T_{v_j}, $1 \le j \le d(v)$.

Our first idea is to introduce a new parameter $\omega(T_v, i)$ defined for each vertex v of T and each color $c_i \in C$ as follows:

$$\omega(T_v, i) = \min\{\omega(f) \mid f \text{ is an edge-coloring of } T_v \text{ and } c_i \in \text{Miss}(f,v)\}.$$

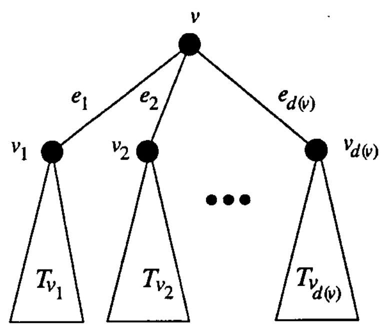

Fig. 2. Subtree T_v rooted at v.

Since T has exactly $n-1$ edges and $|C| = n$, T_v has an edge-coloring f in which c_i is not used by f and hence $c_i \in \text{Miss}(f, v)$. Thus $\omega(T_v, i)$ is well-defined. Clearly,

$$\omega(T_v) = \min_{1 \le i \le n} \omega(T_v, i).$$

We compute the values $\omega(T_v, i)$ for all indices i, $1 \le i \le n$, from leaves to root r. Thus the DP table for each vertex v consists of the n values $\omega(T_v, i)$, $1 \le i \le n$, and has a size n.

The second idea is to reduce the size from n to $d(v) + 1$. We first have the following two lemmas on the properties of $\omega(T_v, i)$.

Lemma 2. *For any vertex v of tree T, $\omega(T_v, i)$ is non-increasing with i, that is, $\omega(T_v, i) \ge \omega(T_v, i+1)$ for each index i, $1 \le i \le n-1$.*

Proof. Let f be an edge-coloring of T_v such that $c_i \in \text{Miss}(f, v)$ and $\omega(f) = \omega(T_v, i)$. If $c_{i+1} \in \text{Miss}(f, v)$, then the definition of $\omega(T_v, i+1)$ implies that $\omega(T_v, i) = \omega(f) \ge \omega(T_v, i+1)$. One may thus assume that $c_{i+1} \in C(f, v)$. Then, interchanging colors c_i and c_{i+1} in $P(v; c_i, c_{i+1})$, we obtain another edge-coloring f' of T. By Lemma 1 $\omega(f) \ge \omega(f')$ and $c_{i+1} \in \text{Miss}(f', v)$. Therefore by the definition we have

$$\omega(T_v, i) = \omega(f) \ge \omega(f') \ge \omega(T_v, i+1).$$

Q.E.D.

Lemma 3. *For any vertex v of tree T,*

$$\omega(T_v, d(v)+1) = \omega(T_v, d(v)+2) = \cdots = \omega(T_v, n).$$

Proof. Let i be any index such that $d(v)+2 \leq i \leq n$. Then by Lemma 2 $\omega(T_v, d(v)+1) \geq \omega(T_v, i)$. It therefore suffices to prove that $\omega(T_v, d(v)+1) \leq \omega(T_v, i)$.

Let f be any edge-coloring of T_v such that $c_i \in \mathrm{Miss}(f, v)$ and $\omega(f) = \omega(T_v, i)$. Since $|C(f, v)| = d(v)$, at least one of the $d(v)+1$ colors $c_1, c_2, \cdots, c_{d(v)+1}$ is missing at v. Let c_j be any of them, then $1 \leq j \leq d(v)+1$ and $c_j \in \mathrm{Miss}(f, v)$. By the definition of $\omega(T_v, j)$ we have

$$\omega(T_v, j) \leq \omega(f) = \omega(T_v, i). \tag{1}$$

Since $j \leq d(v)+1$, by Lemma 2

$$\omega(T_v, d(v)+1) \leq \omega(T_v, j). \tag{2}$$

Thus by Eqs. (1) and (2) we have $\omega(T_v, d(v)+1) \leq \omega(T_v, i)$. $\mathcal{Q.E.D.}$

From Lemmas 2 and 3 we immediately have the following lemma.

Lemma 4. *For any vertex v, $\omega(T_v) = \omega(T_v, d(v)+1)$. In particular, $\omega(T) = \omega(T_r, d(r)+1)$.*

Thus our DP table for each vertex v consists of only the $d(v)+1$ values $\omega(T_v, i)$, $1 \leq i \leq d(v)+1$, and hence the size is $d(v)+1$. All the values $\omega(T_v, i)$ for i, $d(v)+2 \leq i \leq n$, are equal to $\omega(T_v, d(v)+1)$.

For each leaf v of T, the DP table for v consists of a single value $\omega(T_v, 1) = 0$. Note that $d(v) = 0$ and T_v has no edge.

Thus it suffices to show how to construct the DP table for an internal vertex v from the DP tables for all the children $v_1, v_2, \cdots, v_{d(v)}$ of v. (See Figure 2.) We first notice that, for each i, $1 \leq i \leq d(v)+1$, T_v has an edge-coloring f such that $\omega(f) = \omega(T_v, i)$, $c_i \in \mathrm{Miss}(f, v)$, and $C(f, v)$ consists of the first $d(v)$ colors in C other than c_i, as in the following lemma.

Lemma 5. *For any vertex v of T and any index i, $1 \leq i \leq d(v)+1$, T_v has an edge-coloring f such that $\omega(f) = \omega(T_v, i)$, $c_i \in \mathrm{Miss}(f, v)$, and $C(f, v) = D_i$, where $D_i = \{c_k \mid 1 \leq k \leq d(v)+1, k \neq i\}$.*

Proof. Let f be any edge-coloring of T_v such that

(i) $\omega(f) = \omega(T_v, i)$; and

(ii) $c_i \in \text{Miss}(f, v)$.

Let

$$\rho(f) = |C(f, v) \cap D_i|. \tag{3}$$

Then clearly, $|C(f, v)| = |D_i| = d(v)$ and $\rho(f) \leq d(v)$. It suffices to prove that, among the edge-colorings of T_v satisfying (i) and (ii), there is an edge-coloring f such that $\rho(f) = d(v)$.

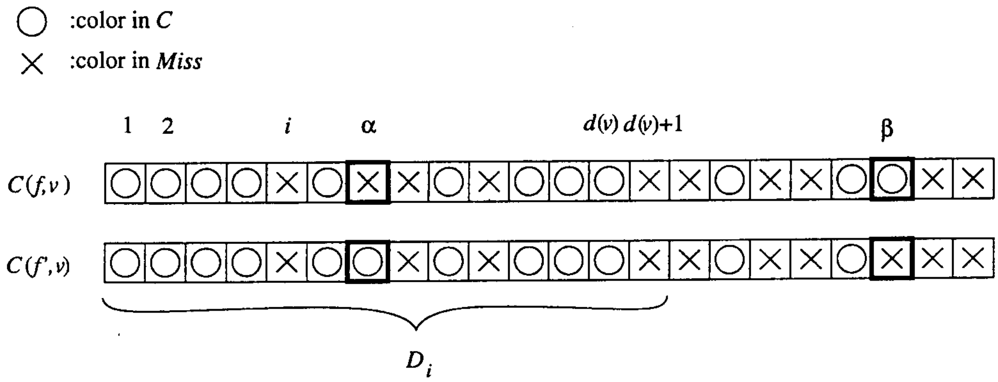

Fig. 3. $C(f, v)$ and $C(f', v)$.

Assume for a contradiction that f is an edge-coloring of T_v having the maximum $\rho(f)$ among all edge-colorings satisfying (i) and (ii), but $\rho(f) < d(v)$. We then show that there is another edge-coloring f' of T_v such that f' satisfies (i) and (ii), but $\rho(f') = \rho(f) + 1$, contrary to the assumption.

We first construct f'. Since $|C(f, v)| = |D_i| = d(v)$ and $\rho(f) < d(v)$, we have $D_i - C(f, v) \neq \emptyset$ and $C(f, v) - D_i \neq \emptyset$. Let c_α be any color in $D_i - C(f, v)$, and let c_β be any color in $C(f, v) - D_i$. Since $c_\alpha \in \text{Miss}(f, v)$ and $c_\beta \in C(f, v)$, interchanging colors c_α and c_β in $P(v; c_\alpha, c_\beta)$, we obtain another edge-coloring f' of T_v. (See Figure 3.)

We then show that f' satisfies (i) and (ii). Since $\alpha, \beta \neq i$, $c_\alpha \in D_i$ and $c_\beta \notin D_i$, we have $\alpha < \beta$ and hence by Lemma 1 $\omega(f') \leq \omega(f)$. Therefore by (i) we have

$$\omega(f') \leq \omega(f) = \omega(T_v, i). \tag{4}$$

Since $c_i \in \text{Miss}(f, v)$ and $i \neq \alpha, \beta$, we have $c_i \in \text{Miss}(f', v)$. Therefore the definition of $\omega(T_v, i)$ implies

$$\omega(T_v, i) \leq \omega(f'). \tag{5}$$

By Eqs. (4) and (5) we have $\omega(f') = \omega(T_v, i)$. Thus f' satisfies (i) and (ii).

We finally show that $\rho(f') = \rho(f) + 1$. Since $c_\alpha \in D_i$, $c_\beta \notin D_i$, and $C(f', v) \cup \{c_\beta\} = C(f, v) \cup \{c_\alpha\}$ by Lemma 1, we have

$$\begin{aligned} C(f', v) \cap D_i &= (C(f', v) \cup \{c_\beta\}) \cap D_i \\ &= (C(f, v) \cup \{c_\alpha\}) \cap D_i \\ &= (C(f, v) \cap D_i) \cup \{c_\alpha\}. \end{aligned} \tag{6}$$

Since $c_\alpha \notin C(f, v)$, by Eq. (6) we have

$$\begin{aligned} \rho(f') &= |C(f', v) \cap D_i| \\ &= |C(f, v) \cap D_i| + |\{c_\alpha\}| \\ &= \rho(f) + 1. \end{aligned}$$

Q.E.D.

We then compute the value $\omega(T_v, i)$ for an index i, $1 \leq i \leq d(v) + 1$, from the $d(v) \times (d(v)+1)$ values $\omega(T_{v_j}, k)$ for all indices j, $1 \leq j \leq d(v)$, and all indices k, $1 \leq k \leq d(v)+1$. Let J be the set of indices j of all children v_j of v, that is, $J = \{j \mid 1 \leq j \leq d(v)\}$. Let $e_j = (v, v_j)$ for each $j \in J$, as illustrated in Figure 2. Let $1 \leq i \leq d(v) + 1$, and let K_i be the set of indices k of all colors c_k in D_i, that is,

$$K_i = \{k \mid 1 \leq k \leq d(v) + 1, k \neq i\}.$$

Suppose that f is an edge-coloring of T_v such that $\omega(f) = \omega(T_v, i)$, $c_i \in \text{Miss}(f, v)$, and $C(f, v) = D_i$. Lemma 5 implies that T_v has such an edge-coloring f. For each $j \in J$, let f_j be the restriction of f to the edges in T_{v_j}, that is, f_j is an edge-coloring of T_{v_j} such that $f_j(e) = f(e)$ for every edge e in T_{v_j}. Then clearly

$$\omega(f) = \omega(T_v, i) = \sum_{j \in J} \omega(f_j) + \omega(D_i), \tag{7}$$

where

$$\omega(D_i) = \sum_{c \in D_i} \omega(c).$$

Note that the value $\omega(D_i)$ depends on i, but does not depend on the edge-coloring f. For each $j \in J$, we denote by j' the index such that $c_{j'} = f(e_j) \in D_i$. Since f is an edge-coloring of T_v, all colors $f(e_j)$ are different from each other. Therefore the mapping $\varphi : J \to K_i$ such that $\varphi(j) = j' \in K_i$ for each $j \in J$ is a bijection. Since $c_{j'} \in \text{Miss}(f_j, v_j)$, the definition of $\omega(T_{v_j}, j')$ implies that $\omega(f_j) \geq \omega(T_{v_j}, j')$. However, we have

$$\omega(f_j) = \omega(T_{v_j}, j') \tag{8}$$

as follows: if $\omega(f_j) > \omega(T_{v_j}, j')$, then T_{v_j} has an edge-coloring f'_j such that $c_{j'} \in \text{Miss}(f'_j, v_j)$ and $\omega(f'_j) = \omega(T_{v_j}, j') < \omega(f_j)$, and hence the edge-coloring f' of T_v obtained from f by replacing f_j with f'_j satisfies $c_i \in \text{Miss}(f', v)$ and $C(f', v) = D_i$ but $\omega(f') < \omega(f) = \omega(T_v, i)$, a contradiction. By Eqs. (7) and (8) we have

$$\omega(f) = \omega(T_v, i) = \sum_{j \in J} \omega(T_{v_j}, j') + \omega(D_i). \tag{9}$$

Suppose conversely that $\varphi : J \to K_i$ is any bijection. For each $j \in J$, we denote $\varphi(j)$ by j', that is, $j' = \varphi(j)$, and let g_j be any edge-coloring of T_{v_j} such that $\omega(g_j) = \omega(T_{v_j}, j')$ and $c_{j'} \in \text{Miss}(g_j, v_j)$. Let g be an edge-coloring of T_v extended from all g_j, $j \in J$, as follows:

$$g(e) = \begin{cases} c_{j'} & \text{if } e = (v, v_j) \text{ and } j \in J; \\ g_j(e) & \text{if } e \text{ is an edge of } T_{v_j} \text{ and } j \in J. \end{cases}$$

Then $c_i \in \text{Miss}(g, v)$, $C(g, v) = D_i$, and

$$\begin{aligned} \omega(g) &= \sum_{j \in J} \omega(g_j) + \omega(D_i) \\ &= \sum_{j \in J} \omega(T_{v_j}, j') + \omega(D_i). \end{aligned} \tag{10}$$

Let

$$b(i) = \min_{\varphi} \sum_{j \in J} \omega(T_{v_j}, j'), \tag{11}$$

where the minimum is taken over all bijections $\varphi : J \to K_i$. Then one can know from Eqs. (9) and (10) that $\omega(g) = \omega(T_v, i)$ if and only if φ attains the minimum value $b(i)$.

We thus have the following lemma.

Lemma 6. *For any internal vertex v of T and any index i, $1 \le i \le d(v) + 1$,*

$$\omega(T_v, i) = b(i) + \omega(D_i).$$

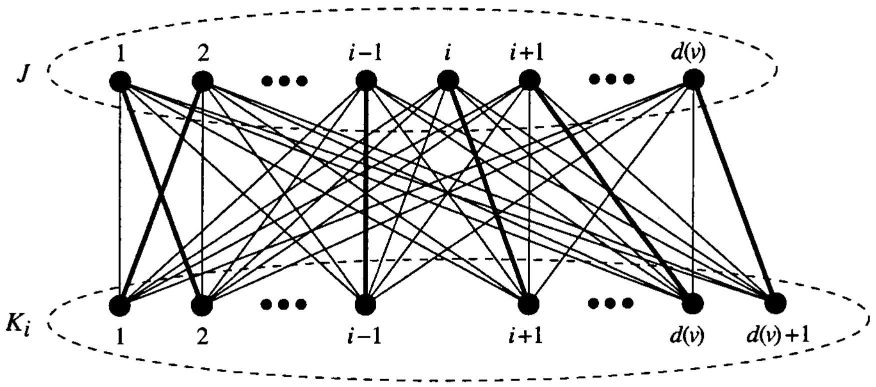

Fig. 4. Complete bipartite graph $B(i) = (J, K_i; J \times K_i)$.

One can easily compute the values $\omega(D_i)$, $1 \le i \le d(v) + 1$, in time $O(d(v))$. We shall therefore show how to compute $b(i)$.

The problem of computing $b(i)$ can be reduced to the minimum weight perfect matching problem for a complete bipartite graph, as follows. Let $B(i) = (J, K_i; J \times K_i)$ be a complete bipartite graph with vertex sets J and K_i. (See Figure 4.) Note that $|J| = |K_i| = d(v)$. Let $h : J \times K_i \to \mathbf{R}$ be a weight function such that $h(e) = \omega(T_{v_j}, k)$ for each edge $e = (j, k)$, $j \in J$ and $k \in K_i$. The minimum weight perfect matching problem is to find a perfect matching M such that the sum of weights of the edges in M is as small as possible [13]. A perfect matching in $B(i)$ is indicated by thick lines in Figure 4. Clearly, $b(i)$ is equal to the weight of the minimum perfect matching in $B(i)$.

The minimum weight perfect matching problem can be solved in time $O(d^3(v))$ for a single graph $B(i)$ [13], and hence the value $b(i)$ for an index i, $1 \le i \le d(v) + 1$, can be computed in time $O(d^3(v))$. Therefore the $d(v) + 1$ values $b(i)$ for all indices i, $1 \le i \le d(v) + 1$, can be computed total in time $O(d^4(v))$.

The third idea is to compute all the $d(v)+1$ values $b(i)$, $1 \le i \le d(v)+1$, together. Kao *et al.* recently showed that if a given complete bipartite graph B has d vertices then one can solve in time $O(d^3)$ the minimum weight perfect matching problems for all d complete bipartite graphs obtained from B by deleting each of the d vertices [8, 9]. Let $K = \{1, 2, \cdots, d(v)+1\}$, let $B = (J, K; J \times K)$ be a complete bipartite graph, and let $h(e) = \omega(T_{v_j}, k)$ for each edge $e = (j, k)$, $j \in J$ and $k \in K$. Applying their algorithm to B, we can compute all the values $b(i)$, $1 \le i \le d(v)+1$, in time $O(d^3(v))$. We thus have the following lemma.

Lemma 7. *For any internal vertex v of T, one can construct the DP table for v from the DP tables for all the children of v in time $O(d^3(v))$.*

We can thus compute the minimum cost $\omega(T) = \omega(T_r, d(r)+1)$ in the DP table for root r in time

$$\begin{aligned} O\left(\sum_{v \in V} d^3(v)\right) &= O\left(\left(\sum_{v \in V} d(v)\right) \Delta^2\right) \\ &= O(n\Delta^2). \end{aligned}$$

Clearly, the total size of all DP tables is $\sum_{v \in V}(d(v)+1) = O(n)$. We do not store the bijection φ found at each internal vertex except the root r.

3.2 Finding an optimal edge-coloring

We next show how to find an optimal edge-coloring f of T. Although we compute the DP tables from leaves to the root, we decide the colors of edges from the root r to leaves, as follows.

When we computed

$$\omega(T) = \omega(T_r, d(r)+1) = b(d(r)+1) + \omega(D_{d(r)+1}),$$

we found a bijection $\varphi : J \to K_{d(r)+1}$ attaining the minimum value $b(d(r)+1)$ in Eq. (11), where $J = K_{d(r)+1} = \{j \mid 1 \le j \le d(r)\}$. For each j, $1 \le j \le d(r)$, we color the edge e_j joining r and its jth child r_j with $c_{\varphi(j)} \in D_{d(r)+1} = \{c_1, c_2, \cdots, c_{d(r)}\}$. We hence have $C(f, r) = \{c_1, c_2, \cdots, c_{deg(r)}\}$ since $d(r) = deg(r)$.

We may thus assume that, for an internal vertex v, the edge joining v and its parent u has been colored with a color $c_i \in C$ for some index i, $1 \le i \le deg(u)$. Let $i' = i$ if $i \le d(v) + 1$, and let $i' = d(v) + 1$ if $i \ge d(v) + 2$. We find again the bijection $\varphi : J \to K_{i'}$ attaining the minimum value $b(i')$ in Eq. (11). For each j, $1 \le j \le d(v)$, we color the edge $e_j = (v, v_j)$ with $c_{\varphi(j)} \in D_{i'}$. We hence have $C(f, v) = D_{i'} \cup \{c_i\} \subseteq \{c_1, c_2, \cdots, c_{deg(v)}\} \cup \{c_i\}$ since $d(v) + 1 = deg(v)$.

Thus we can correctly find an optimal edge-coloring of T, and the coloring uses the first $\Delta(T)$ cheapest colors $c_1, c_2, \cdots, c_\Delta$ in C. Clearly, the algorithm takes time $O(n\Delta^2)$. This completes a proof of Theorem 1.

Kao *et al.* also show that if a given complete bipartite graph B has d vertices and all edge weights are integers in the range $[-N, N]$ then one can solve in time $O(d^{2.5} \log(dN))$ the minimum weight perfect matching problems for all d complete bipartite graphs obtained from B by deleting each of the d vertices [8, 9]. Using their algorithm, one can find an optimal edge-coloring of tree T in time $O(n\Delta^{1.5} \log(nN_\omega))$ for the case where all the color costs are integers in the range $[-N_\omega, N_\omega]$, because $d \le 2\Delta + 1 \le 2n$ and $N \le nN_\omega$ in this case. Our algorithm thus solves the edge-chromatic sum problem for trees in time $O(n\Delta^{1.5} \log n)$, because $N_\omega = \Delta$ for the problem.

Acknowledgments

We thank Tak-Wah Lam and Takeshi Tokuyama for pointing out some references.

References

1. S. Arnborg and J. Lagergren. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.
2. R.B. Borie, R.G. Parker, and C.A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7:555–581, 1992.
3. E.G. Coffman, M.R. Garey, D.S. Johnson, and A.S. LaPaugh. Scheduling file transfers. *SIAM J. Comput.*, 14:744–780, 1985.

4. K. Giaro and M. Kubale. Edge-chromatic sum of trees and bounded cyclicity graphs. *Information Processing Letters*, 75:65–69, 2000.
5. H. Hajiabolhassan, M.L. Mehrabadi and R. Tusserkani. Minimal coloring and strength of graphs. *Discrete Mathematics*, 215:265–270, 2000.
6. K. Jansen. The optimum cost chromatic partition problem. In *Proc. CIAC'97, Lecture Notes in Computer Science, Springer-Verlag*, 1203:25–36, 1997.
7. H. Krawczyk and M. Kubale. An approximation algorithm for diagnostic test scheduling in multicomputer systems. *IEEE Trans. Comput.*, 34:869–872, 1985.
8. M.Y. Kao, T.W. Lam, W.K. Sung and H.F. Ting. All-cavity maximum matchings. In *Proc. ISAAC'97, Lecture Notes in Computer Science, Springer-Verlag*, 1350:364–373, 1997.
9. M.Y. Kao, T.W. Lam, W.K. Sung and H.F. Ting. Cavity matchings, label compressions, and evolutionary trees. *SIAM J. Comp.*, 30(2):602–624, 2000.
10. 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 International Workshop on Graph Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, Springer*, 1197:279–292, 1997.
11. M. Kubale. *Introduction to Computational Complexity and Algorithmic Graph Coloring.* Gdańskie Towarzystwo Naukowe, Gdańsk, Poland, 1998.
12. E. Kubicka. *The Chromatic Sum of a Graph.* Ph.D. Thesis, Western Michigan University, 1989.
13. C.H. Padadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall, Englewood Cliffs, NJ, 1982.
14. J. A. Telle and A. Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math.*, 10:529–550, 1997.
15. X. Zhou, S. Nakano, and T. Nishizeki. Edge-coloring partial k-trees. *Journal of Algorithms*, 21:598–617, 1996.
16. X. Zhou and T. Nishizeki. Algorithm for the cost edge-coloring of trees. In *Proc. of the 7th Annual International Conference on Computing and Combinatorics, Lect. Notes in Computer Science, Springer*, 2108:288-297, 2001.