IEICE TRANS. INF. & SYST., VOL.E94–D, NO.2 FEBRUARY 2011

PAPER Special Section on Foundations of Computer Science — Mathematical Foundations and Applications of Algorithms and Computer Science —

Minimum Cost Edge-Colorings of Trees Can Be Reduced to Matchings

Takehiro ITO[†a)], **Naoki SAKAMOTO**[†b)], ***Nonmembers***, **Xiao ZHOU**[†c)], ***Member***, ***and*** **Takao NISHIZEKI**[††d)], ***Fellow***

SUMMARY Let C be a set of colors, and let $\omega(c)$ be an integer cost assigned to a 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 C. The cost $\omega(f)$ of an edge-coloring f of G is the sum of costs $\omega(f(e))$ of colors $f(e)$ assigned to all edges e in G. An edge-coloring f of G is optimal if $\omega(f)$ is minimum among all edge-colorings of G. In this paper, we show that the problem of finding an optimal edge-coloring of a tree T can be simply reduced in polynomial time to the minimum weight perfect matching problem for a new bipartite graph constructed from T. The reduction immediately yields an efficient simple algorithm to find an optimal edge-coloring of T in time $O(n^{1.5}\Delta\log(nN_\omega))$, where n is the number of vertices in T, Δ is the maximum degree of T, and N_ω is the maximum absolute cost $|\omega(c)|$ of colors c in C. We then show that our result can be extended for multitrees.

key words: *algorithm, cost edge-coloring, multitree, perfect matching, tree*

1. Introduction

Let $G = (V, E)$ be a graph with vertex set V and edge set E, and let C be a set of colors. An *edge-coloring* of G is to color all the edges in E so that any two adjacent edges are colored with different colors in C. The minimum number of colors required for edge-colorings of G is called the *chromatic index*, and is denoted by $\chi'(G)$. It is well-known that $\Delta(G) \le \chi'(G) \le \Delta(G) + 1$ for every simple graph G and that $\chi'(G) = \Delta(G)$ for every bipartite (multi)graph G, where $\Delta(G)$ is the maximum degree of G [9]. The ordinary *edge-coloring problem* is to compute the chromatic index $\chi'(G)$ of a given graph G and find an edge-coloring of G using $\chi'(G)$ colors. Let ω be a cost function which assigns an integer $\omega(c)$ to each color $c \in C$, then 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 $i \ge 2$, then the graph G with $\chi'(G) = 3$ in Fig. 1 (a) can be uniquely colored with the three cheapest colors c_1, c_2 and c_3 as in Fig. 1 (a), but this 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 illustrated in Fig. 1 (b). However, every simple graph G has an optimal edge-coloring using $\Delta(G)$ or $\Delta(G) + 1$ colors [6], [8], and every bipartite (multi)graph G and hence every tree has an optimal edge-coloring using $\Delta(G)$ (= $\chi'(G)$) colors [1], [6]. The edge-chromatic sum problem, introduced by Giaro and Kubale [5], is merely the cost edge-coloring problem for the special case where $\omega(c_i) = i$ for each integer $i \ge 1$.

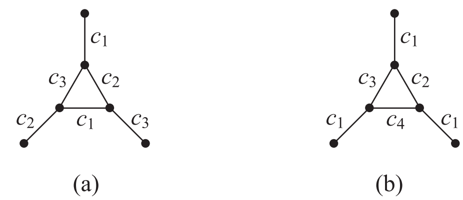

Fig. 1 (a) Edge-coloring using $\chi'(G) = 3$ colors, and (b) optimal edge-coloring using $\chi'(G) + 1 = 4$ colors, where $\omega(c_1) = 1$ and $\omega(c_2) = \omega(c_3) = \omega(c_4) = 5$.

The cost edge-coloring problem has a natural application for scheduling [10]. 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 [2]. 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. Suppose that a task executed in the ith time slot takes the cost $\omega(c_i)$. Then the goal is to find a schedule that minimizes the total cost, and hence this corresponds to the cost edge-coloring problem.

The cost edge-coloring problem is APX-hard even for bipartite graphs [3], and hence there is no polynomial-time approximation scheme (PTAS) for the problem unless P = NP. On the other hand, Zhou and Nishizeki gave an algorithm to solve the cost edge-coloring problem for trees T in time $O(n\Delta^{1.5}\log(nN_\omega))$, where n is the number of vertices in T, Δ is the maximum degree of T, and N_ω is the maximum absolute cost $|\omega(c)|$ of colors c in C [10]. The algorithm is based on a dynamic programming approach, and computes

Manuscript received March 29, 2010.
†The authors are with the Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8579 Japan.
††The author is with School of Science and Technology, Kwansei Gakuin University, Sanda-shi, 669–1337 Japan.
a) E-mail: takehiro@ecei.tohoku.ac.jp
b) E-mail: sakamoto@nishizeki.ecei.tohoku.ac.jp
c) E-mail: zhou@ecei.tohoku.ac.jp
d) E-mail: nishi@kwansei.ac.jp
DOI: 10.1587/transinf.E94.D.190

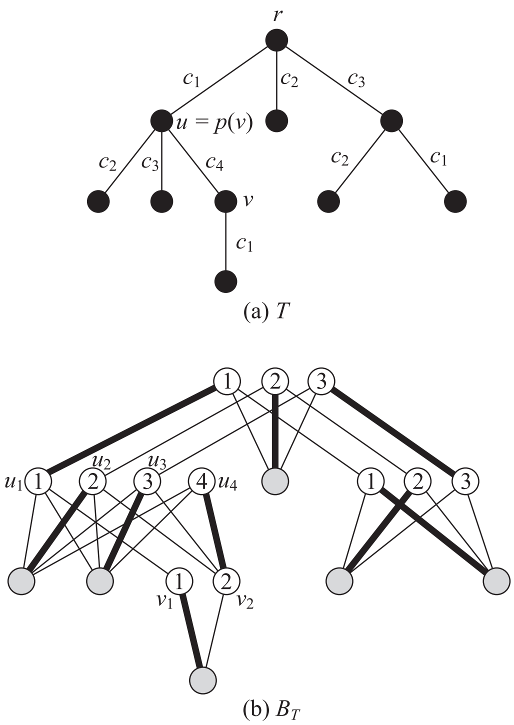

Fig. 2 (a) Optimal compact edge-coloring of a tree T, and (b) perfect matching of B_T, whose edges are drawn by thick lines.

a DP table for each vertex of a given tree T from the leaves to the root of T. For computing the DP tables, the algorithm needs to construct $O(n)$ bipartite graphs in total and solves the minimum weight perfect matching problem for each of them.

In this paper, we first show that the cost edge-coloring problem for a tree T can be simply reduced in polynomial time to the problem of finding a minimum weight perfect matching in an edge-weighted bipartite graph B_T constructed from T, as illustrated in Fig. 2. The reduction takes time $O(n\Delta)$, and yields an efficient simple algorithm to find an optimal edge-coloring of T in time $O(n^{1.5}\Delta\log(nN_\omega))$. Our algorithm constructs a single bipartite graph B_T, and solves only once the minimum weight perfect matching problem for B_T. Thus, our algorithm is much simpler than the known algorithm [10], and can be easily implemented. We then show that the algorithm for trees can be extended for multitrees, which will be defined in Sect. 5.

The rest of the paper is organized as follows. In Sect. 2 we first define some basic terms which will be used throughout the paper. We then give the reduction in Sect. 3. In Sect. 4 we prove a lemma used by the reduction. In Sect. 5 we show that the algorithm for trees can be extended for multitrees. Finally, in Sect. 6 we give a conclusion.

2. Preliminaries

In this section, we define some basic terms.

Let $T = (V, E)$ be a tree with a set V of vertices and a set E of edges. We sometimes denote by $V(T)$ and $E(T)$ the vertex set and the edge set of T, respectively. We choose an arbitrary vertex r of T as the *root*, and regard T as a rooted tree. We denote by n the number of vertices in T, that is, $n = |V|$. One may assume that $n \geq 2$. The *degree* $d(v)$ of a vertex v is the number of edges in E incident to v. We denote the maximum degree of T by $\Delta(T)$ or simply by Δ. We denote by $\mathrm{ch}(v)$ the number of edges joining a vertex v and its children in T. Then, $\mathrm{ch}(r) = d(r)$, and $\mathrm{ch}(v) = d(v)-1$ for every vertex $v \in V \setminus \{r\}$. We denote by $p(v)$ the parent of a vertex $v \in V \setminus \{r\}$ in T.

Although T has an optimal edge-coloring using $\Delta(T)$ colors [1], [6], we assume for the sake of convenience that $|C| = \Delta(T) + 1$, and we write $C = \{c_1, c_2, \cdots, c_{\Delta+1}\}$. An *edge-coloring* $f : E \to C$ of a tree $T = (V, E)$ is to color all the edges of T by colors in C so that any two adjacent edges are colored with different colors. Let $\omega : C \to \mathbb{Z}$ be a *cost function*, where $\mathbb{Z}$ is the set of all integers. One may assume without loss of generality that ω is non-decreasing, that is, $\omega(c_i) \leq \omega(c_{i+1})$ for every index i, $1 \leq i \leq \Delta$. 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 *optimal* 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.

For an edge-coloring f of a tree T and a vertex v of T, we denote by $C(f, v)$ the set of all colors that are assigned to the 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 $c \notin C(f, v)$. We denote by $\mathrm{Miss}(f, v)$ the set of all colors missing at v, that is, $\mathrm{Miss}(f, v) = C \setminus C(f, v)$.

Interchanging colors in an "alternating path" is one of the standard techniques for ordinary edge-colorings [9], which we also use in the paper. Let f be an edge-coloring of a tree T, let c_α and c_β be any two colors in C, and let $T(c_\alpha, c_\beta)$ be the subgraph of T induced by all edges colored with c_α or c_β. Since T is a tree, each connected component of $T(c_\alpha, c_\beta)$ is a path, called a $c_\alpha c_\beta$*-alternating path*, whose edges are colored alternately with c_α and c_β. A vertex $v \in V$ is an end of a $c_\alpha c_\beta$-alternating path if and only if exactly one of the two colors c_α and c_β is missing at v. We denote by $P(v; c_\alpha, c_\beta)$ a $c_\alpha c_\beta$-alternating path starting with v. Interchanging colors c_α and c_β in $P(v; c_\alpha, c_\beta)$, one can obtain another edge-coloring f' of T.

For a graph $G = (V, E)$, a subset M of E is called a *matching* of G if no two edges in M share a common vertex. A matching M of G is *perfect* if every vertex of G is an end of an edge in M. Thus, $|M| = \frac{1}{2}|V|$ for every perfect matching M of G. Let $w : E \to \mathbb{Z}$ be a weight function which assigns an integer weight $w(e) \in \mathbb{Z}$ to each edge e in G. Then, the *weight* $w(M)$ *of a matching* M of G is defined as follows:

$$w(M) = \sum_{e \in M} w(e).$$

The *minimum weight perfect matching problem* is to find a perfect matching M of a given graph G such that $w(M)$ is minimum among all perfect matchings in G. The problem can be solved for a bipartite graph $G = (V, E)$ in time $O(\sqrt{|V|}|E| \log(|V|N_w))$, where N_w is the maximum absolute weight $|w(e)|$ of edges e in E [4].

3. Reduction

Our main result is the following.

Theorem 1: The cost edge-coloring problem for a tree T can be reduced in time $O(n\Delta)$ to the minimum weight perfect matching problem for a single bipartite graph B_T constructed from T.

Before presenting the reduction, we introduce a "compact" edge-coloring of a tree. Let $T = (V, E)$ be a tree with root r. An edge-coloring f of T is *compact* if the following two conditions (i) and (ii) hold:

(i) for the root r of T, $C(f, r) = \{c_1, c_2, \cdots, c_{\mathrm{ch}(r)}\}$; and
(ii) for each vertex $v \in V \setminus \{r\}$, $C(f, v) = \{c_1, c_2, \cdots, c_{\mathrm{ch}(v)}, c_k\}$ for some index k such that
  (a) $k \geq \mathrm{ch}(v) + 1$; and
  (b) if $k \geq d(v) + 1$, then $k \leq d(u)$ and c_k is assigned to the edge joining v and the parent $u = p(v)$.

For example, the edge-coloring in Fig. 2 (a) is compact. Clearly, a compact edge-coloring uses colors $c_1, c_2, \cdots, c_\Delta$ and does not use color $c_{\Delta+1}$. We then have the following lemma, whose proof will be given in Sect. 4.

Lemma 1: Every tree T has an optimal edge-coloring which is compact.

We now give the reduction from the cost edge-coloring problem for a tree T to the minimum weight perfect matching problem for a bipartite graph B_T.

The bipartite graph $B_T = (V_B, E_B)$ can be constructed from a tree $T = (V, E)$, as follows. (See Figs. 2 and 3.)

(i) For each vertex $v \in V$, add $d(v)$ vertices $v_1, v_2, \cdots, v_{d(v)}$ to V_B.
(ii) For each edge $(u, v) \in E$ with $u = p(v)$, add $d(u)$ edges to E_B, as follows: for each index i, $1 \leq i \leq d(u)$, join vertices u_i and v_j by an edge whose weight is $w((u_i, v_j)) = \omega(c_i)$, where

$$j = \begin{cases} i & \text{if } i \leq d(v); \\ d(v) & \text{otherwise.} \end{cases}$$

Clearly, $|V_B| = \sum_{v \in V} d(v) = 2(n-1)$ and $|E_B| = \sum_{(u,v) \in E} d(u) = O(n\Delta)$. Therefore, the bipartite graph B_T can be constructed from T in time $O(n\Delta)$. Clearly, the maximum absolute weight $N_w = \max\{|\omega(c_1)|, |\omega(c_\Delta)|\}$ of edges in B_T is not greater than the maximum absolute cost $N_\omega = \max\{|\omega(c_1)|, |\omega(c_{\Delta+1})|\}$ of colors in C.

For each edge (u, v) in T, we denote by $B_T(u, v)$ the subgraph of B_T induced by vertices $u_1, u_2, \cdots, u_{d(u)}$ and $v_1, v_2, \cdots, v_{d(v)}$. $B_T(u, v)$ corresponds to edge (u, v) of T. (See Fig. 3.) We then have the following lemma.

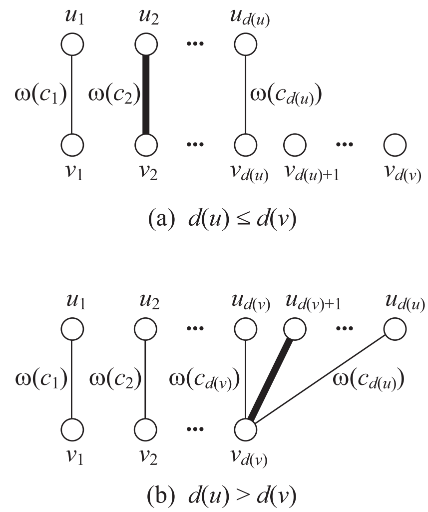

Fig. 3 Subgraph $B_T(u, v)$ of B_T corresponding to an edge (u, v) of T.

Lemma 2: For every tree T, the following (a) and (b) hold:

(a) every perfect matching M of B_T contains exactly one of the edges in $B_T(u, v)$ for every edge (u, v) of T, as illustrated in Fig. 3 where edges in M are drawn by thick lines; and
(b) every perfect matching M of B_T induces a compact edge-coloring f of T. Conversely, every compact edge-coloring f of T induces a perfect matching M of B_T. Furthermore, $\omega(f) = w(M)$.

Proof. (a) Let M be a perfect matching of B_T. We prove from the leaves to the root that M contains exactly one of the edges of $B_T(u, v)$. One may assume that $u = p(v)$.

If v is a leaf of T, then $B_T(u, v)$ is a star with center v_1 and only the edges of $B_T(u, v)$ are incident to v_1 in B_T. Therefore, the perfect matching M of B_T contains exactly one edge of $B_T(u, v)$, say (u_k, v_1) for some index k, $1 \leq k \leq d(u)$.

One may thus assume that v is an internal vertex of T, and that M contains exactly one of the edges of $B_T(v, w)$ for each child w of v in T. Since v has a parent u in T, we have $v \neq r$ and hence $\mathrm{ch}(v) = d(v) - 1$. Therefore, M contains exactly $d(v) - 1$ edges in the bipartite subgraphs corresponding to the edges of T joining v and its $d(v) - 1$ children. Hence, exactly one of the vertices $v_1, v_2, \cdots, v_{d(v)}$, say v_j, is not an end of these $d(v) - 1$ edges in M. Since M is a perfect matching of B_T, M contains exactly one edge (u_k, v_j) of $B_T(u, v)$ for some index k, $1 \leq k \leq d(u)$.

(b) Every perfect matching M of B_T induces an edge-coloring f of T, in which each edge (u, v) of T is colored with c_k for the index k above; the edge of $B_T(u, v)$ contained in M has an end u_k, $1 \leq k \leq d(u)$. One can easily observe that the edge-coloring f is compact.

Conversely, every compact edge-coloring f of T induces a perfect matching M of B_T; if $u = p(v)$ and $f((u, v)) = c_i$, $1 \le i \le d(u)$, then M contains an edge joining u_i and v_j where

$$j = \begin{cases} i & \text{if } i \le d(v); \\ d(v) & \text{otherwise.} \end{cases}$$

Obviously, $\omega(f) = w(M)$. □

By Lemma 1 every tree T has an optimal edge-coloring f which is compact, and hence by Lemma 2 (b) B_T has a perfect matching M such that $w(M) = \omega(f)$. Remember that $|V_B| = O(n)$, $|E_B| = O(n\Delta)$, and the maximum absolute weight N_w of edges in B_T is not greater than the maximum absolute cost N_ω of colors in C. Since a minimum weight perfect matching of B_T can be found in time $O(\sqrt{|V_B|}|E_B| \log(|V_B|N_w))$ [4], we can find an optimal edge-coloring of T in time $O(n^{1.5}\Delta \log(nN_\omega))$.

4. Proof of Lemma 1

In this section, we give a proof of Lemma 1.

Let $T = (V, E)$ be a tree with root r. For a vertex w of T, we denote by T_w the subtree of T which is rooted at w and is induced by w and all descendants of w in T. (See Fig. 4 (a).) Clearly, $T = T_r$.

Let w be an arbitrary vertex of T. Since $\chi'(T_w) \le \Delta(T)$ and $|C| = \Delta(T) + 1$, for each color $c_i \in C$, T_w has an edge-coloring f in which c_i is not used and hence $c_i \in \mathrm{Miss}(f, w)$. Let

$$\omega(T_w, i) = \min\{\omega(f) \mid f \text{ is an edge-coloring of } T_w \text{ such that } c_i \in \mathrm{Miss}(f, w)\}.$$

For a color $c_i \in C$, an edge-coloring f of T_w is defined to be (w, i)-*compact* if the following two conditions (i) and (ii) hold:

(i) $c_i \in \mathrm{Miss}(f, w)$; and
(ii) if $i \ge \mathrm{ch}(w) + 1$ then $C(f, w) = \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\}$, and otherwise $C(f, w) \cup \{c_i\} = \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)+1}\}$.

We then have the following lemma.

Lemma 3: For each color $c_i \in C$, T_w has a (w, i)-compact edge-coloring f such that $\omega(f) = \omega(T_w, i)$.

Proof. We give a proof only for the case where $i \ge \mathrm{ch}(w) + 1$. (The proof for the other case is similar.) The definition of $\omega(T_w, i)$ implies that T_w has an edge-coloring f such that $c_i \in \mathrm{Miss}(f, w)$ and $\omega(f) = \omega(T_w, i)$. In particular, let f be an edge-coloring of T_w such that $|C(f, w) \cap \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\}|$ is maximum among all these edge-colorings. Suppose for a contradiction that f is not (w, i)-compact. Then, $C(f, w) \ne \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\}$. Since $|C(f, w)| = \mathrm{ch}(w)$, there exist two colors c_α and c_β such that

$$c_\alpha \in \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\} \setminus C(f, w)$$

and

$$c_\beta \in C(f, w) \setminus \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\}.$$

Since $\alpha \le \mathrm{ch}(w) < \beta$, we have $\omega(c_\alpha) \le \omega(c_\beta)$. Since $i \ge \mathrm{ch}(w) + 1$, we have $c_\alpha \ne c_i$. Since $c_i \in \mathrm{Miss}(f, w)$ and $c_\beta \in C(f, w)$, we have $c_\beta \ne c_i$. Since $c_\alpha \in \mathrm{Miss}(f, w)$ and $c_\beta \in C(f, w)$, there is a $c_\alpha c_\beta$-alternating path $P(w; c_\alpha, c_\beta)$ starting from w. We obtain another edge-coloring f' of T_w by interchanging colors c_α and c_β in $P(w; c_\alpha, c_\beta)$. Since $\omega(c_\alpha) \le \omega(c_\beta)$, $\omega(f') \le \omega(f)$. Since $c_i \ne c_\alpha, c_\beta$ and $c_i \in \mathrm{Miss}(f, w)$, we have $c_i \in \mathrm{Miss}(f', w)$ and hence $\omega(T_w, i) \le \omega(f')$. Therefore, $\omega(T_w, i) \le \omega(f') \le \omega(f) = \omega(T_w, i)$ and hence $\omega(f') = \omega(T_w, i)$. Since $c_\alpha \in C(f', w)$ and $\alpha \le \mathrm{ch}(w) < \beta$, we have

$$C(f', w) \cap \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\} = \left(C(f, w) \cap \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\}\right) \cup \{c_\alpha\}$$

and hence

$$|C(f', w) \cap \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\}| > |C(f, w) \cap \{c_1, c_2, \cdots, c_{\mathrm{ch}(w)}\}|,$$

a contradiction. □

A (w, i)-compact edge-coloring f of T_w is defined to be (T_w, i)-*compact* if the following condition (iii) holds:

(iii) for each vertex $v \in V(T_w) \setminus \{w\}$, $C(f, v) = \{c_1, c_2, \cdots, c_{\mathrm{ch}(v)}, c_k\}$ for some index k such that
  (a) $k \ge \mathrm{ch}(v) + 1$; and
  (b) if $k \ge d(v) + 1$, then $k \le d(u)$ and c_k is assigned to the edge joining v and the parent $u = p(v)$.

Clearly, an edge-coloring f of T with root r is compact if f is $(T_r, \mathrm{ch}(r) + 1)$-compact. One can show that the cost $\omega(f)$

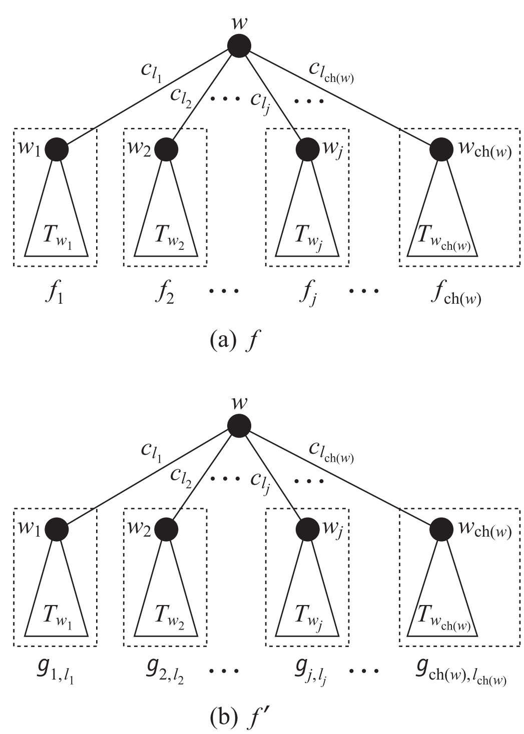

Fig. 4 (a) A (w, i)-compact edge-coloring f of T_w, and (b) a (T_w, i)-compact edge-coloring f' of T_w.

of an optimal edge-coloring f of T is equal to $\omega(T_r, \mathrm{ch}(r)+1)$ [10, Lemma 4]. Therefore, as a proof of Lemma 1, it suffices to prove the following lemma.

Lemma 4: For each vertex w of T and each color $c_i \in C$, T_w has a (T_w, i)-compact edge-coloring f such that $\omega(f) = \omega(T_w, i)$.

Proof. We prove the lemma by induction on the number of vertices in T_w.

For the base case, let w be a leaf of T. Then, T_w is a tree of a single vertex w, and hence the lemma trivially holds.

Let c_i be a color in C, and let w be an internal vertex of T. Let $w_1, w_2, \cdots, w_{\mathrm{ch}(w)}$ be the children of w, as illustrated in Fig. 4 (a). Suppose as the induction hypothesis that the lemma holds for each color $c_l \in C$ and each subtree T_{w_j}, $1 \le j \le \mathrm{ch}(w)$. Then, for each color $c_l \in C$, T_{w_j} has a (T_{w_j}, l)-compact edge-coloring $g_{j,l}$ such that $\omega(g_{j,l}) = \omega(T_{w_j}, l)$.

By Lemma 3, T_w has a (w, i)-compact edge-coloring f such that $\omega(f) = \omega(T_w, i)$. If f is (T_w, i)-compact, then we have done. So we may assume that f is not (T_w, i)-compact. For each subtree T_{w_j}, $1 \le j \le \mathrm{ch}(w)$, let $f_j = f|T_{w_j}$ be the restriction of f to T_{w_j}, that is, $f_j(e) = f(e)$ for each edge e of T_{w_j}. Let c_{l_j} be the color assigned to the edge (w, w_j), $1 \le j \le \mathrm{ch}(w)$, by f, as illustrated in Fig. 4 (a). Then one can easily observe that $c_{l_j} \in \mathrm{Miss}(f_j, w_j)$ and $\omega(f_j) = \omega(T_{w_j}, l_j) = \omega(g_{j,l_j})$ for each j, $1 \le j \le \mathrm{ch}(w)$. We now construct another edge-coloring f' of T_w, as follows (see Fig. 4 (b)):

$$f'(e) = \begin{cases} g_{j,l_j}(e) & \text{if } e \in E(T_{w_j}) \text{ for some } j,\ 1 \le j \le \mathrm{ch}(w); \\ f(e) & \text{otherwise.} \end{cases}$$

Clearly, f' is (T_w, i)-compact and $\omega(f') = \omega(f) = \omega(T_w, i)$. □

5. Multitrees

Replace each edge in a tree by multiple edges, as illustrated in Fig. 5 (a). The resulting multigraph is called a *multitree*. In this section, we show that our reduction for trees can be extended for multitrees.

Theorem 2: The cost edge-coloring problem for multitrees $T = (V, E)$ can be reduced in time $O(|E|\Delta)$ to the minimum weight perfect matching problem for a bipartite graph B_T, and can be solved in time $O(|E|^{1.5}\Delta \log(|E|N_\omega))$.

Let $T = (V, E)$ be a multitree with root r. Since T is a bipartite multigraph, T has an optimal edge-coloring using Δ colors [1]. For a vertex $v \in V \setminus \{r\}$, we denote by $m(v)$ the number of multiple edges joining v and $p(v)$. Thus $m(v) = d(v) - \mathrm{ch}(v)$. Similarly as for trees, an edge-coloring f of a multitree T is defined to be *compact* if the following two conditions (i) and (ii) hold:

(i) for the root r of T, $C(f, r) = \{c_1, c_2, \cdots, c_{\mathrm{ch}(r)}\}$; and
(ii) for each vertex $v \in V \setminus \{r\}$, $C(f, v) = \{c_1, c_2, \cdots, c_{\mathrm{ch}(v)}, c_{k_1}, c_{k_2}, \cdots, c_{k_{m(v)}}\}$ for some indices k_j, $1 \le j \le m(v)$, such that
 (a) $k_j \ge \mathrm{ch}(v) + 1$; and
 (b) if $k_j \ge d(v) + 1$, then $k_j \le d(u)$ and c_{k_j} is assigned to an edge joining v and the parent $u = p(v)$.

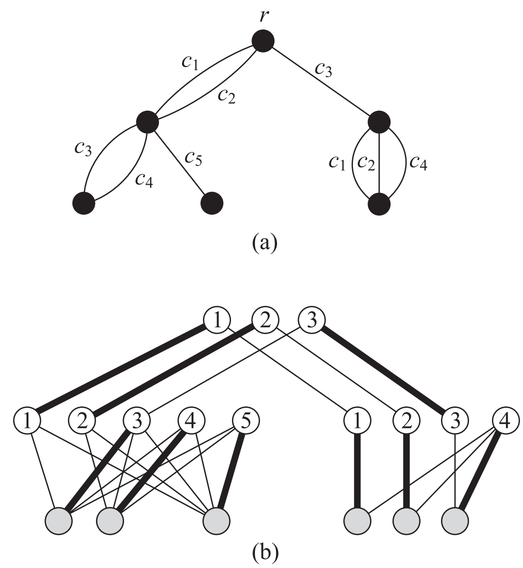

Fig. 5 (a) Optimal compact edge-coloring of a multitree T, and (b) its corresponding perfect matching in B_T whose edges are drawn by thick lines.

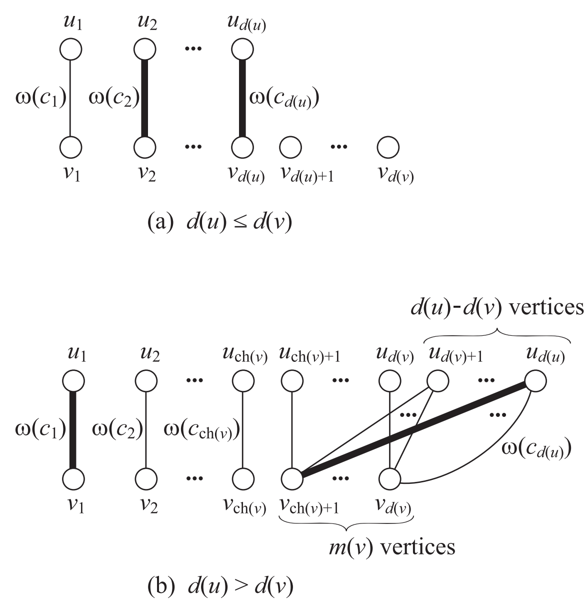

Fig. 6 Subgraph $B_T(u, v)$ of B_T corresponding to multiple edges joining v and $u = p(v)$ in T.

Figure 5 (a) depicts a compact edge-coloring of a multitree. Clearly, a compact edge-coloring uses colors $c_1, c_2, \cdots, c_\Delta$ and does not use color $c_{\Delta+1}$. Similarly as Lemma 1, one can prove that every multitree has an optimal edge-coloring which is compact.

The bipartite graph $B_T = (V_B, E_B)$ for a multitree $T = (V, E)$ can be constructed as follows. (See Figs. 5 and 6.)

(i) For each vertex $v \in V$, add $d(v)$ vertices $v_1, v_2, \cdots, v_{d(v)}$ to V_B.
(ii) For each set of $m(v)$ multiple edges joining vertices v

and $u = p(v)$, add edges to E_B, as follows: for each index i, $1 \le i \le d(u)$, join vertices u_i and v_j by an edge whose weight is $w((u_i, v_j)) = \omega(c_i)$, where

$$j = \begin{cases} i & \text{if } i \le d(v); \\ \mathrm{ch}(v)+1, \mathrm{ch}(v)+2, \cdots, d(v) & \text{otherwise.} \end{cases}$$

Clearly, $|V_B| = 2|E|$. If $d(u) \le d(v)$, then $|E(B_T(u,v))| = d(u)$. If $d(u) > d(v)$, then $|E(B_T(u,v))| = d(v) + (d(u) - d(v))m(v)$. In either case, $|E(B_T(u,v))| \le d(u)m(v)$ because $m(v) \ge 1$. Therefore, $|E_B| \le \sum d(u)m(v) = O(\Delta|E|)$, where the summention is taken over all pairs (u,v) such that $u = p(v)$.

Similarly as in Lemma 2, one can prove that every perfect matching M of B_T contains exactly $m(v)$ edges in $B_T(u,v)$; every compact edge-coloring f of a multitree T induces a perfect matching M of B_T, and *vice versa*; and $\omega(f) = w(M)$. Thus, our reduction for trees can be extended for multitrees, and hence Theorem 2 holds.

6. Conclusions

In this paper, we show that the cost edge-coloring problem for a tree T can be reduced in time $O(n\Delta)$ to the minimum weight perfect matching problem for the bipartite graph B_T. This reduction immediately yields an algorithm which actually finds an optimal edge-coloring of T in time $O(n^{1.5}\Delta \log(nN_\omega))$. We then show that the algorithm for trees can be extended for multitrees.

References

[1] J. Cardinal, V. Ravelomanana, and M. Valencia-Pabon, "Chromatic edge strength of some multigraphs," Electronic Notes in Discrete Mathematics, vol.30, pp.39–44, 2008.
[2] E.G. Coffman, M.R. Garey, D.S. Johnson, and A.S. LaPaugh, "Scheduling file transfers," SIAM J. Comput., vol.14, pp.744–780, 1985.
[3] D. Marx, "Complexity results for minimum sum edge coloring," Discrete Appl. Math., vol.157, pp.1034–1045, 2009.
[4] H.N. Gabow and R.E. Tarjan, "Faster scaling algorithms for network problems," SIAM J. Comput., vol.18, pp.1013–1036, 1989.
[5] K. Giaro and M. Kubale, "Edge-chromatic sum of trees and bounded cyclicity graphs," Inf. Process. Lett., vol.75, pp.65–69, 2000.
[6] H. Hajiabolhassan, M. L. Mehrabadi, and R. Tusserkani, "Minimal coloring and strength of graphs," Discrete Mathematics, vol.215, pp.265–270, 2000.
[7] H. Krawczyk and M. Kubale, "An approximation algorithm for diagnostic test scheduling in multicomputer systems," IEEE Trans. Comput., vol.34, no.9, pp.869–872, 1985.
[8] J. Mitchem, P. Morriss, and E. Schmeichel, "On the cost chromatic number of outerplanar, planar, and line graphs," Discussiones Mathematicae Graph Theory, vol.17, pp.229–241, 1997.
[9] D. B. West, Introduction to Graph Theory, 2nd ed., Prentice Hall, New Jersey, 2000.
[10] X. Zhou and T. Nishizeki, "Algorithm for the cost edge-coloring of trees," J. Combinatorial Optimization, vol.8, pp.97–108, 2004.

Takehiro Ito skipped the second semester of the senior grade of an undergraduate course at Tohoku University, Japan, and received the M.S. and Ph.D. degrees from Tohoku University in 2003 and 2006, respectively. He is currently Assistant Professor of Graduate School of Information Sciences, Tohoku University. His research interests include graph algorithms and computational complexity.

Naoki Sakamoto received the B.E. and M.S. degrees from Tohoku University, Japan, in 2007 and 2009, respectively. His research interests include graph coloring algorithms.

Xiao Zhou received the B.E. degree from the East China Normal University, Shanghai, China, in 1986, and the M.E. and Ph.D. degrees from Tohoku University, Japan, in 1992 and 1995, respectively. He is currently Professor of Graduate School of Information Sciences, Tohoku University. His areas of research interest are combinatorial algorithms, graph theory and computational complexity.

Takao Nishizeki received the B.E., M.E., and Dr. Eng. degrees from Tohoku University, Japan, in 1969, 1971, and 1974, respectively, all in electrical communication engineering. He joined Tohoku University in 1974, and has been a Professor of Graduate School of Information Sciences until March 2010. He is currently an emeritus professor of Tohoku University and a professor of Kwansei Gakuin University, School of Science and Technology. His areas of research interest are combinatorial algorithms, graph theory, and cryptology. He is a Fellow of ACM, IEEE and the Information Processing Society of Japan.