A Fast Algorithm for Computing
a Nearly Equitable Edge Coloring
with Balanced Conditions

Akiyoshi SHIOURA[*]   Mutsunori YAGIURA[†]

January 27, 2009

Abstract

We discuss the nearly equitable edge coloring problem on a multigraph and propose an efficient algorithm for solving the problem, which has a better time complexity than the previous algorithms. The coloring computed by our algorithm satisfies additional balanced conditions on the number of edges used in each color class, where conditions are imposed on the balance among all edges in the multigraph as well as the balance among parallel edges between each vertex pair. None of the previous algorithms are guaranteed to satisfy these balanced conditions simultaneously. To achieve these improvements, we propose a new recoloring procedure, which is based on a set of edge-disjoint alternating walks, while the existing algorithms are based on an Eulerian circuit or a single alternating walk. This new recoloring procedure makes it possible to reduce the time complexity of the algorithm.

1 Introduction

1.1 Problem Definition and Main Results

We discuss the nearly equitable edge coloring problem on a multigraph. Let $G = (V, E)$ be a multigraph; a multigraph is an undirected graph which may have parallel edges and/or loops. Throughout this paper, we denote by n and m the numbers of vertices and edges in G, respectively. Let $\mathcal{C} = \{1, 2, \ldots, k\}$ be a set of k colors. An edge coloring of a multigraph G is an assignment of k colors to edges in E, which is represented by a function $\pi : E \to \mathcal{C}$.

Let $\pi : E \to \mathcal{C}$ be an edge coloring. For each vertex $v \in V$ and a color $i \in \mathcal{C}$, we denote by $d_\pi(v, i)$ the number of edges in E incident to v with color i. We say that an edge coloring π of a multigraph G is *nearly equitable* if it satisfies the condition

$$(\text{NEC}) \qquad |d_\pi(v, i) - d_\pi(v, j)| \leq 2 \qquad (\forall v \in V, \forall i, j \in \mathcal{C}).$$

The main aim of this paper is to propose a new algorithm for computing a nearly equitable edge coloring of a given multigraph. The time complexity of the proposed algorithm is better than the previous algorithms.

[*]Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan, `shioura@dais.is.tohoku.ac.jp`.

[†]Graduate School of Information Sciences, Nagoya University, Nagoya 464-8603, Japan, `yagiura@nagoya-u.jp`.

Table 1: Comparison of algorithms for the nearly equitable edge coloring problem. The mark "$\sqrt{}$" means that the output of the algorithm satisfies the conditions (B1) and/or (B2).

authors	time complexity	(B1)	(B2)	technique used
Hilton & de Werra (1982) [5]	$\mathrm{O}(km^2)$		$\sqrt{}$	single alternating walk
Nakano et al. (1995) [9]	$\mathrm{O}(m^2/k + mn)$			single alternating walk
Xie et al. (2004) [16]	$\mathrm{O}(m^2/k)$	$\sqrt{}$		Eulerian circuit
Xie et al. (2008) [17]	$\mathrm{O}(mn\log(m/(nk)+1))$	$\sqrt{}$		Eulerian circuit
Ours	$\mathrm{O}(\min\{mn, m^2/k\})$	$\sqrt{}$	$\sqrt{}$	edge-disjoint alternating walks

In addition to the condition (NEC), we consider the following two "balanced" conditions on the number of edges used in each color class:

$$(\text{B1}) \quad ||E_\pi^i| - |E_\pi^j|| \leq 1 \qquad (\forall i, j \in \mathcal{C}),$$
$$(\text{B2}) \quad ||E_\pi^i(u,v)| - |E_\pi^j(u,v)|| \leq 1 \qquad (\forall i, j \in \mathcal{C}, \forall u, v \in V),$$

where π is an edge coloring and

$$
\begin{aligned}
E_\pi^i &= \{e \in E \mid \pi(e) = i\} \quad (i \in \mathcal{C}), \\
E_\pi^i(u,v) &= \{e \in E \mid \pi(e) = i, \ e \text{ connects } u \text{ and } v\} \quad (i \in \mathcal{C}, \ u, v \in V).
\end{aligned}
$$

The first condition (B1) imposes that the number of all edges in each color class is almost the same, while the second condition (B2) imposes that each color class uses almost the same number of parallel edges between each pair of vertices. Note that the conditions (B1) and (B2) are equivalent to the following (B1$'$) and (B2$'$), respectively:

$$(\text{B1}') \quad |E_\pi^i| \in \{\lfloor m/k \rfloor, \lceil m/k \rceil\} \qquad (\forall i \in \mathcal{C}),$$
$$(\text{B2}') \quad |E_\pi^i(u,v)| \in \{\lfloor m(u,v)/k \rfloor, \lceil m(u,v)/k \rceil\} \qquad (\forall i \in \mathcal{C}, \forall u, v \in V),$$

where $m(u,v)$ $(u, v \in V)$ denotes the number of parallel edges connecting u and v. We show that the nearly equitable edge coloring computed by our algorithm satisfies both of the balanced conditions. Our main result is summarized as follows:

Theorem 1.1. *Our algorithm computes a nearly equitable edge coloring of a multigraph satisfying the conditions* (B1) *and* (B2) *in* $\mathrm{O}(\min\{mn, m^2/k\})$ *time.*

Table 1 shows a summary of the previous algorithms for the nearly equitable edge coloring problem. The time complexity of our algorithm is better than the previous best bound $\mathrm{O}(mn\log(m/(nk)+1))$ by Xie et al. [17].[1] Moreover, our algorithm is the first to compute a nearly equitable edge coloring satisfying both of the conditions (B1) and (B2). The algorithms in [16, 17] outputs a nearly equitable edge coloring satisfying (B1), and the output of the algorithm in [5] satisfies (B2), but none of the previous algorithms is guaranteed to obtain a coloring satisfying both of (B1) and (B2) (see Table 1).

---

[1]It is pointed out in Xie et al. [17] that $mn\log(m/(nk)+1) = \Theta(m^2/k)$ holds for any m, n and k satisfying $0 < m/nk \leq 1$. From this fact it is not difficult to show that the algorithm in [17] is never asymptotically slower than that of [16], and that our new algorithm is never asymptotically slower than that of [17].

To compute a nearly equitable edge coloring, our algorithm iteratively modifies an edge coloring. For this, we propose a new recoloring procedure, which is based on a set of edge-disjoint alternating walks, while the previous algorithms are based on an Eulerian circuit [16, 17] or a single alternating walk [5, 9] (see Table 1). This recoloring procedure makes it possible to reduce the time complexity of the algorithm while keeping the conditions (B1) and (B2) of an edge coloring.

In the following discussion, we assume $k \leq m$ without loss of generality, since otherwise the problem is trivial.

1.2 Previous and Related Work

An edge coloring π of a multigraph G is said to be *equitable* if it satisfies the condition

$$\text{(EC)} \qquad |d_\pi(v, i) - d_\pi(v, j)| \leq 1 \qquad (\forall i, j \in \mathcal{C}, \forall v \in V),$$

which is stronger than the condition (NEC). Although every bipartite multigraph has an equitable edge coloring, non-bipartite multigraphs may not have an equitable edge coloring (see, e.g., [6, 12, 13]). A typical example is an odd cycle, which has no equitable edge coloring with $k = 2$. Several sufficient conditions for multigraphs to have an equitable edge coloring are shown in [5, 6, 11, 14]. Note that the problem of determining the existence of an equitable edge coloring is NP-complete (see [17]).

The balanced conditions (B1) and (B2) have often been discussed in the literature of (nearly) equitable edge coloring [2, 5, 6, 13, 15, 16, 17]. The first condition (B1) is referred to as "equalized condition" in [2] and "balanced condition" in [10, 16, 17], and the second condition (B2) is referred to as "edge-balanced condition" in [5].

Recently, a weighted version of the equitable edge coloring problem is discussed in [1, 3], and the following conjecture for bipartite multigraphs is raised in [1]:

given a multigraph $G = (V, E)$, a set of colors $\mathcal{C} = \{1, 2, \ldots, k\}$, and weights w_i ($i \in \mathcal{C}$) with $0 < w_i < 1$ and $\sum_{i \in \mathcal{C}} w_i = 1$, there exists an edge coloring such that

$$\lfloor w_i d(v) \rfloor \leq d_\pi(v, i) \leq \lceil w_i d(v) \rceil \qquad (\forall i \in \mathcal{C}, \ \forall v \in V). \tag{1.1}$$

Note that the condition (1.1) coincides with the condition (EC) if $w_i = 1/k$ for all $i \in \mathcal{C}$. The conjecture holds for some special cases, but does not hold in general, especially when G is not bipartite. The following relaxed statement where both of the upper and lower bounds are relaxed by two is proven for bipartite multigraphs in [1] and for general multigraphs in [3].

Theorem 1.2 ([1, 3]). *Given a multigraph $G = (V, E)$, a set of colors $\mathcal{C} = \{1, 2, \ldots, k\}$, and weights w_i ($i \in \mathcal{C}$) with $0 < w_i < 1$ and $\sum_{i \in \mathcal{C}} w_i = 1$, there exists an edge coloring such that*

$$\lfloor w_i d(v) \rfloor - 2 \leq d_\pi(v, i) \leq \lceil w_i d(v) \rceil + 2 \qquad (\forall i \in \mathcal{C}, \ \forall v \in V).$$

1.3 Overview of Our Algorithm

Our algorithm starts with an initial edge coloring satisfying (B1) and (B2), and repeatedly improves the edge coloring, without violating (B1) and (B2), so that it satisfies the condition (NEC) in the end. As in many previous papers in the area of edge coloring, our algorithm

3

improves an edge coloring by *switching edge colors of alternating walks* (see, e.g., [4, 7]); the difference from the previous approach is that our algorithm uses a set of edge-disjoint alternating walks, not a single alternating walk, in each iteration. If a set of edge-disjoint alternating walks is chosen in a naive way, we can only show that the algorithm terminates in $O(m)$ iterations. To reduce the number of iterations, a set of edge-disjoint alternating walks is chosen in a deliberate way, which leads to the bound $O(\min\{kn, m\})$ on the number of iterations. We show that each iteration can be done in $O(m/k)$ time, and therefore the time complexity of the proposed algorithm is $O((m/k) \times \min\{kn, m\}) = O(\min\{mn, m^2/k\})$.

2   Switch of Edge Colors

The proposed algorithm modifies an edge coloring by using an operation called *switch*. For every distinct colors $\alpha, \beta \in \mathcal{C}$, we denote by $G_\pi(\alpha, \beta)$ the subgraph of G given by $G_\pi(\alpha, \beta) = (V, E_\pi^\alpha \cup E_\pi^\beta)$. Given an edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$, *switching* edge colors of S means to interchange the colors α and β of edges in S; more formally, switching edge colors of S is to modify the current edge coloring $\pi : E \to \mathcal{C}$ to the new edge coloring $\pi' : E \to \mathcal{C}$ given by

$$\pi'(e) = \begin{cases} \beta & (e \in S, \ \pi(e) = \alpha), \\ \alpha & (e \in S, \ \pi(e) = \beta), \\ \pi(e) & (e \in E \setminus S). \end{cases}$$

To switch edge colors, the algorithm uses an edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$ satisfying the following condition:

$$\left\{ \begin{array}{l} \text{if } d_\pi(v, \alpha) \geq d_\pi(v, \beta), \text{ then } 0 \leq d_\pi^S(v, \alpha) - d_\pi^S(v, \beta) \leq d_\pi(v, \alpha) - d_\pi(v, \beta), \\ \text{if } d_\pi(v, \alpha) \leq d_\pi(v, \beta), \text{ then } 0 \geq d_\pi^S(v, \alpha) - d_\pi^S(v, \beta) \geq d_\pi(v, \alpha) - d_\pi(v, \beta), \end{array} \right\} \tag{2.1}$$

where for each $v \in V$ and $i \in \{\alpha, \beta\}$, we denote by $d_\pi^S(v, i)$ the number of edges in S incident to v with color i. We say that S is *eligible* in the multigraph $G_\pi(\alpha, \beta)$ if it satisfies the condition (2.1) for all $v \in V$. Eligible edge sets are useful in getting a better edge coloring, as shown below.

Proposition 2.1. *Let $a, b, c \in \mathbb{R}$ be any real numbers such that $a \geq b$ and $0 \leq c \leq a - b$. Then, we have $\min\{a, b\} \leq \min\{a - c, b + c\} \leq \max\{a - c, b + c\} \leq \max\{a, b\}$.*

Lemma 2.2. *Let $\pi : E \to \mathcal{C}$ be an edge coloring and $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$ an eligible edge set. Then, the new edge coloring $\pi' : E \to \mathcal{C}$ obtained by switching edge colors of S satisfies*

$$\min\{d_\pi(v, \alpha), d_\pi(v, \beta)\} \leq \min\{d_{\pi'}(v, \alpha), d_{\pi'}(v, \beta)\}$$
$$\leq \max\{d_{\pi'}(v, \alpha), d_{\pi'}(v, \beta)\} \leq \max\{d_\pi(v, \alpha), d_\pi(v, \beta)\} \qquad (\forall v \in V).$$

Proof. The claim follows from (2.1), Proposition 2.1, and the following equations for each $v \in V$:

$$d_{\pi'}(v, \alpha) = d_\pi(v, \alpha) - \{d_\pi^S(v, \alpha) - d_\pi^S(v, \beta)\} \qquad d_{\pi'}(v, \beta) = d_\pi(v, \beta) + \{d_\pi^S(v, \alpha) - d_\pi^S(v, \beta)\}.$$

$\square$

To keep the balanced conditions (B1) and (B2), we consider the following two conditions for an edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$:

4

(S1)   if $|E_\pi^\alpha| = |E_\pi^\beta| + 1$, then $|S \cap E_\pi^\alpha| - |S \cap E_\pi^\beta| = 0$ or $+1$,
           if $|E_\pi^\alpha| = |E_\pi^\beta|$, then $|S \cap E_\pi^\alpha| - |S \cap E_\pi^\beta| = 0$,
           if $|E_\pi^\alpha| = |E_\pi^\beta| - 1$, then $|S \cap E_\pi^\alpha| - |S \cap E_\pi^\beta| = 0$ or -1,
(S2)   for every $u, v \in V$,
           if $|E_\pi^\alpha(u,v)| = |E_\pi^\beta(u,v)| + 1$, then $|S \cap E_\pi^\alpha(u,v)| - |S \cap E_\pi^\beta(u,v)| = 0$ or $+1$,
           if $|E_\pi^\alpha(u,v)| = |E_\pi^\beta(u,v)|$, then $|S \cap E_\pi^\alpha(u,v)| - |S \cap E_\pi^\beta(u,v)| = 0$,
           if $|E_\pi^\alpha(u,v)| = |E_\pi^\beta(u,v)| - 1$, then $|S \cap E_\pi^\alpha(u,v)| - |S \cap E_\pi^\beta(u,v)| = 0$ or -1.

Lemma 2.3. *Let $\pi : E \to \mathcal{C}$ be an edge coloring, and $\pi' : E \to \mathcal{C}$ be the new edge coloring obtained by switching edge colors of an edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$.*
(i) If π and S satisfy (B1) and (S1), respectively, then π' satisfies (B1).
(ii) If π and S satisfy (B2) and (S2), respectively, then π' satisfies (B2).

The following is one of the key properties used in our algorithm. The proof will be given in Section 5.

Lemma 2.4. *Let $\pi : E \to \mathcal{C}$ be an edge coloring. Suppose that there exist two distinct colors $\alpha, \beta \in \mathcal{C}$ and a vertex $u \in V$ such that $d_\pi(u, \alpha) - d_\pi(u, \beta) \geq 3$ holds. For any integer $r \in \mathbb{Z}$ such that $1 \leq r \leq d_\pi(u, \alpha) - d_\pi(u, \beta) - 2$, we can compute an eligible edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$ satisfying the conditions (S1), (S2), and $d_\pi^S(u, \alpha) - d_\pi^S(u, \beta) \in \{r, r + 1\}$ in $\mathrm{O}(|E_\pi^\alpha \cup E_\pi^\beta|)$ time.*

3   Proposed Algorithm

We explain our algorithm for computing a nearly equitable edge coloring satisfying the conditions (B1) and (B2).

Our algorithm starts with an initial edge coloring satisfying (B1) and (B2), which can be easily computed in $\mathrm{O}(m)$ time by using the following property.

Proposition 3.1. *Let $\{e_1, e_2, \ldots, e_m\}$ be an ordered list of the edges in E such that the parallel edges connecting the same pair of vertices are ordered consecutively, and color each edge e_t $(t = 1, 2, \ldots, m)$ by the color $(t \mod k) + 1$. Then, the resulting edge coloring satisfies the conditions (B1) and (B2).*

Proof. The condition (B1) is easy to see. The condition (B2) is satisfied since the parallel edges connecting the same pair of vertices are ordered consecutively. ☐

The algorithm always keeps the two conditions (B1) and (B2) satisfied, and iteratively improves the edge coloring so that the condition (NEC) is satisfied in the end.

To obtain an edge coloring π satisfying the condition (NEC), our algorithm processes each vertex $u \in V$ one by one. If the vertex u violates the condition

$$|d_\pi(u, i) - d_\pi(u, j)| \leq 2 \qquad (\forall i, j \in \mathcal{C}), \tag{3.1}$$

then the algorithm repeatedly updates the edge coloring π by switching edge colors of an eligible edge set S until the condition (3.1) is satisfied. By Lemma 2.2, once the vertex u satisfies the condition (3.1), the edge coloring always satisfies (3.1) in the following iterations.

Suppose that the vertex u violates the condition (3.1). Our algorithm implicitly maintains the following sets of colors:

$$
\begin{aligned}
\mathcal{C}_\pi^0(u) &= \{i \in \mathcal{C} \mid \lceil d(u)/k \rceil - 1 \le d_\pi(u,i) \le \lfloor d(u)/k \rfloor + 1\}, & (3.2)\\
\mathcal{C}_\pi^+(u) &= \{i \in \mathcal{C} \mid d_\pi(u,i) \ge \lfloor d(u)/k \rfloor + 2\}, & (3.3)\\
\mathcal{C}_\pi^-(u) &= \{i \in \mathcal{C} \mid d_\pi(u,i) \le \lceil d(u)/k \rceil - 2\}. & (3.4)
\end{aligned}
$$

Note that $\{\mathcal{C}_\pi^0(u), \mathcal{C}_\pi^+(u), \mathcal{C}_\pi^-(u)\}$ is a partition of $\mathcal{C}$. Whenever both of $\mathcal{C}_\pi^+(u)$ and $\mathcal{C}_\pi^-(u)$ are nonempty, the algorithm chooses two distinct colors α, β with $\alpha \in \mathcal{C}_\pi^+(u)$ and $\beta \in \mathcal{C}_\pi^-(u)$, which is done by choosing α and β satisfying $d_\pi(u,\alpha) = \max_{i \in \mathcal{C}} d_\pi(u,i)$ and $d_\pi(u,\beta) = \min_{i \in \mathcal{C}} d_\pi(u,i)$. Then, the algorithm updates the edge coloring π so that at least one of α and β is contained in $\mathcal{C}_\pi^0(u)$. This can be done efficiently by Lemma 2.4 with the value r given by

$$
r = \min\{d_\pi(u,\alpha) - (\lfloor d(u)/k \rfloor + 1), (\lceil d(u)/k \rceil - 1) - d_\pi(u,\beta)\}. \quad (3.5)
$$

Repeating these steps, we obtain either $\mathcal{C}_\pi^+(u) = \emptyset$ or $\mathcal{C}_\pi^-(u) = \emptyset$ (or both). Suppose that $\mathcal{C}_\pi^-(u) = \emptyset$ holds. Note that in this case, the right-hand side of (3.5) is zero. Then, the algorithm iteratively updates the edge coloring π so that the value $\sum\{d_\pi(u,i) - \lceil d(u)/k \rceil \mid i \in \mathcal{C}_\pi^+(u)\}$ decreases at least by one while keeping the condition $\mathcal{C}_\pi^-(u) = \emptyset$. This is done by choosing two colors α and β with the same rule as above, and then using Lemma 2.4 with $r = 1$. In this way, the algorithm computes an edge coloring π satisfying (3.1).

Our algorithm is described as follows.

Algorithm FASTBALANCING($G, \mathcal{C}$)
Input: a multigraph $G = (V, E)$ and a set of colors $\mathcal{C} = \{1, 2, \ldots, k\}$.
Output: a nearly equitable edge coloring $\pi : E \to \mathcal{C}$ of G satisfying (B1) and (B2).
1. Compute an initial edge coloring π satisfying the conditions (B1) and (B2).
2. **for** each $u \in V$ **do**
3.   Compute the value $d_\pi(u,i)$ for all $i \in \mathcal{C}$.
4.   **while** $\exists i, j \in \mathcal{C}$ such that $|d_\pi(u,i) - d_\pi(u,j)| \ge 3$ **do**
5.   Compute colors $\alpha, \beta \in \mathcal{C}$ such that $d_\pi(u,\alpha) = \max_{i \in \mathcal{C}} d_\pi(u,i)$, $d_\pi(u,\beta) = \min_{i \in \mathcal{C}} d_\pi(u,i)$.
6.   Compute an eligible edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$ satisfying (S1), (S2), and
         $d_\pi^S(u,\alpha) - d_\pi^S(u,\beta) \in \{r, r+1\}$, where r is given by
         $r = \max\{1, \min\{d_\pi(u,\alpha) - (\lfloor d(u)/k \rfloor + 1), (\lceil d(u)/k \rceil - 1) - d_\pi(u,\beta)\}\}$.
7.   Modify the edge coloring π by switching edge colors of S.
8. Output π and stop.

We note that an eligible edge set S in Line 6 can always be obtained by Lemma 2.4. It is easy to see that the condition (NEC) is satisfied when the algorithm terminates. Since the edge set S chosen in Line 6 satisfies the conditions (S1) and (S2), the edge coloring π always satisfies (B1) and (B2) by Lemma 2.3. Hence, the output of the algorithm is a nearly equitable edge coloring satisfying (B1) and (B2).

4   Analysis of Time Complexity

We analyze the time complexity of the algorithm FASTBALANCING. First of all, we analyze the number of iterations of Lines 5–7 for a fixed vertex $u \in V$, where we use a convex function

$\varphi_z : \mathbb{R} \to \mathbb{R}$ defined by

$$\varphi_z(x) = \max\{\lfloor z \rfloor - x, 0, x - \lceil z \rceil\} \qquad (x \in \mathbb{R}),$$

where $z \in \mathbb{R}$ is a real number.

Proposition 4.1. *Let $z \in \mathbb{R}$ be any real number, and $a, b, c \in \mathbb{Z}$ any integers such that $a > z > b$ and $1 \leq c \leq a - b - 1$. Then, we have $\varphi_z(a - c) + \varphi_z(b + c) \leq \varphi_z(a) + \varphi_z(b) - 1$.*

Proof. The proof is given in Appendix. $\qquad\square$

For an edge coloring $\pi : E \to \mathcal{C}$ and a vertex $u \in V$, we define

$$\Phi(\pi, u) = \sum_{i \in \mathcal{C}} \varphi_{d(u)/k}(d_\pi(u, i)).$$

The value $\Phi(\pi, u)$ is a nonnegative integer for every edge coloring π, and $\Phi(\pi, u) = 0$ holds if and only if $\lfloor d(u)/k \rfloor \leq d_\pi(u, i) \leq \lceil d(u)/k \rceil$ for all $i \in \mathcal{C}$. Thus, the value $\Phi(\pi, u)$ represents the degree of unbalance in the edge coloring π at the vertex u.

Lemma 4.2. *Let π be an edge coloring, $u \in V$ be a vertex, and $\alpha, \beta \in \mathcal{C}$ be distinct colors such that*

$$d_\pi(u, \alpha) = \max_{i \in \mathcal{C}} d_\pi(u, i), \quad d_\pi(u, \beta) = \min_{i \in \mathcal{C}} d_\pi(u, i), \quad d_\pi(u, \alpha) - d_\pi(u, \beta) \geq 3.$$

Suppose that π' is an edge coloring obtained by switching edge colors of an eligible edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$ with

$$1 \leq d_\pi^S(u, \alpha) - d_\pi^S(u, \beta) \leq d_\pi(u, \alpha) - d_\pi(u, \beta) - 1. \qquad (4.1)$$

Then, we have $\Phi(\pi', u) \leq \Phi(\pi, u) - 1$.

Proof. It suffices to show that

$$\varphi_{d(u)/k}(d_{\pi'}(u, \alpha)) + \varphi_{d(u)/k}(d_{\pi'}(u, \beta)) \leq \varphi_{d(u)/k}(d_\pi(u, \alpha)) + \varphi_{d(u)/k}(d_\pi(u, \beta)) - 1.$$

Since $(1/k) \sum_{i \in \mathcal{C}} d_\pi(u, i) = d(u)/k$ and $d_\pi(u, \alpha) = \max_{i \in \mathcal{C}} d_\pi(u, i) > \min_{i \in \mathcal{C}} d_\pi(u, i) = d_\pi(u, \beta)$, we have $d_\pi(u, \alpha) > d(u)/k > d_\pi(u, \beta)$. In addition, $d_{\pi'}(u, \alpha) = d_\pi(u, \alpha) - c$ and $d_{\pi'}(u, \beta) = d_\pi(u, \beta) + c$ hold with $c = d_\pi^S(u, \alpha) - d_\pi^S(u, \beta)$. Hence, Proposition 4.1 implies that

$$
\begin{aligned}
\varphi_{d(u)/k}(d_{\pi'}(u, \alpha)) + \varphi_{d(u)/k}(d_{\pi'}(u, \beta)) &= \varphi_{d(u)/k}(d_\pi(u, \alpha) - c) + \varphi_{d(u)/k}(d_\pi(u, \beta) + c) \\
&\leq \varphi_{d(u)/k}(d_\pi(u, \alpha)) + \varphi_{d(u)/k}(d_\pi(u, \beta)) - 1.
\end{aligned}
$$

$\qquad\square$

Lemma 4.3. *For a fixed vertex $u \in V$, the number of iterations in the while loop in the algorithm* FastBalancing *is* $O(d(u))$.

Proof. The eligible set S computed in Line 6 satisfies the condition (4.1). Hence, the claim follows from Lemma 4.2 and the fact that $\Phi(\pi, u) = O(d(u))$. $\qquad\square$

Lemma 4.4. *For a fixed vertex $u \in V$, the number of iterations in the while loop in the algorithm* FastBalancing *is* $O(k)$.

Proof. In each iteration of the while loop, we consider the sets $\mathcal{C}_\pi^0(u), \mathcal{C}_\pi^+(u), \mathcal{C}_\pi^-(u)$ defined by (3.2), (3.3), and (3.4), respectively. Suppose that the colors α and β chosen in Line 5 satisfy $\alpha \in \mathcal{C}_\pi^+(u)$, $\beta \in \mathcal{C}_\pi^-(u)$. Recall that α and β are such that $d_\pi(u,\alpha) = \max_{i \in \mathcal{C}} d_\pi(u,i)$ and $d_\pi(u,\beta) = \min_{i \in \mathcal{C}} d_\pi(u,i)$. Let S be an edge set chosen in Line 6. Since the value r in Line 6 satisfies

$$r = \min\{d_\pi(u,\alpha) - (\lfloor d(u)/k \rfloor + 1), (\lceil d(u)/k \rceil - 1) - d_\pi(u,\beta)\} \geq 1,$$

at least one of α and β is contained in $\mathcal{C}_\pi^0(u)$ after switching edge colors of S. This fact implies that in at most k iterations, we have either $\mathcal{C}_\pi^+(u) = \emptyset$ or $\mathcal{C}_\pi^-(u) = \emptyset$.

Assume, without loss of generality, that $\mathcal{C}_\pi^-(u) = \emptyset$. Then, we have

$$\frac{d(u)}{k} - d_\pi(u,i) \leq \left\lceil \frac{d(u)}{k} \right\rceil - d_\pi(u,i) \leq 1 \qquad (\forall i \in \mathcal{C}). \tag{4.2}$$

Since $\sum_{i \in \mathcal{C}} \{d_\pi(u,i) - d(u)/k\} = d(u) - d(u) = 0$, it holds that

$$\sum \left\{ d_\pi(u,i) - \frac{d(u)}{k} \;\Big|\; i \in \mathcal{C}, \; d_\pi(u,i) > \frac{d(u)}{k} \right\} = \sum \left\{ \frac{d(u)}{k} - d_\pi(u,i) \;\Big|\; i \in \mathcal{C}, \; d_\pi(u,i) \leq \frac{d(u)}{k} \right\}.$$

Hence, we have

$$
\begin{aligned}
\Phi(\pi,u) \;\; &\leq \;\; \sum_{i \in \mathcal{C}} \max \left\{ d_\pi(u,i) - \frac{d(u)}{k}, \frac{d(u)}{k} - d_\pi(u,i) \right\} \\
&= \;\; 2 \sum \left\{ \frac{d(u)}{k} - d_\pi(u,i) \;\Big|\; i \in \mathcal{C}, \; d_\pi(u,i) \leq \frac{d(u)}{k} \right\} \;\; \leq \;\; 2k,
\end{aligned}
$$

where the last inequality is by (4.2). This fact, together with Lemma 4.2, implies that the while loop terminates in at most $2k$ iterations. This concludes the proof. $\square$

By Lemmas 4.3 and 4.4, the number of iterations of Lines 5–7 for a fixed vertex $u \in V$ is $\mathrm{O}(\min\{k, d(u)\})$. We can compute an eligible edge set S satisfying the desired conditions in $\mathrm{O}(|E_\pi^\alpha \cup E_\pi^\beta|) = \mathrm{O}(m/k)$ time by Lemma 2.4. Switching edge colors in Line 7 requires $\mathrm{O}(|S|) = \mathrm{O}(m/k)$ time. Maintenance of values $d_\pi(u,i)$ and Line 5 can be done in $\mathrm{O}(m)$ time in total by using a data structure shown in [17, Section 3]. Hence, the algorithm FASTBALANCING computes a nearly equitable edge coloring of a multigraph satisfying the conditions (B1) and (B2) in $\mathrm{O}((m/k) \times \sum_{u \in V} \min\{k, d(u)\}) = \mathrm{O}(\min\{mn, m^2/k\})$ time. This concludes the proof of Theorem 1.1.

5 Computing Eligible Edge Sets

In this section we give a proof of Lemma 2.4, which states that an eligible edge set $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$ satisfying the conditions (S1), (S2), and an additional condition on the number $d_\pi^S(v,\alpha) - d_\pi^S(v,\beta)$ can be found in $\mathrm{O}(|E_\pi^\alpha \cup E_\pi^\beta|)$ time. To prove this, we consider a decomposition of the edge set $E_\pi^\alpha \cup E_\pi^\beta$ by using eligible alternating walks to be defined below.

A *walk* is a sequence of vertices and edges of the form $u_0 e_1 u_1 e_2 u_2 \ldots e_{t-1} u_{t-1} e_t u_t$, where $u_0, u_1, \ldots, u_t$ are vertices and $e_1, e_2, \ldots, e_t$ are distinct edges such that e_j connects the vertices u_{j-1} and u_j for $j = 1, 2, \ldots, t$. It should be mentioned that a walk may visit the same vertex more than once; in particular, it is possible that the first and last vertices u_0 and u_t are the same. A walk is said to be *eligible* if the set of all edges in the walk is eligible. In the following discussion, we may regard a walk as the set of edges $\{e_1, e_2, \ldots, e_t\}$ to simplify the description.

Let $\pi : E \to \mathcal{C}$ be an edge coloring, and $\alpha, \beta \in \mathcal{C}$ distinct colors. We call a walk P in the multigraph $G_\pi(\alpha, \beta)$ an *alternating walk* if any two consecutive edges in P have different colors. Alternating walks in $G_\pi(\alpha, \beta)$ can be categorized into the following three types. An *$\alpha\beta$-even alternating walk* is an alternating walk P such that $|P \cap E_\pi^\alpha| = |P \cap E_\pi^\beta|$. An *$\alpha$-odd alternating walk* (resp., a *β-odd alternating walk*) is an alternating walk P such that $|P \cap E_\pi^\alpha| = |P \cap E_\pi^\beta| + 1$ (resp., $|P \cap E_\pi^\beta| = |P \cap E_\pi^\alpha| + 1$). In the following, we mainly consider eligible alternating walks in $G_\pi(\alpha, \beta)$.

Lemma 5.1 ([7, 8, 9]). *Let $u_0 \in V$ be a vertex such that $d_\pi(u_0, \alpha) \neq d_\pi(u_0, \beta)$. Then, there exists an eligible alternating walk $P = u_0 e_1 u_1 e_2 u_2 \ldots e_{t-1} u_{t-1} e_t u_t$ starting from u_0.*

A partition $\{P_1, P_2, \ldots, P_s, R\}$ ($s \geq 0$) of the edge set $E_\pi^\alpha \cup E_\pi^\beta$ of the multigraph $G_\pi(\alpha, \beta)$ is called an *alternating walk decomposition* if P_h ($h = 1, 2, \ldots, s$) are eligible alternating walks satisfying the following condition:

$$\sum_{h=1}^s \{d_\pi^{P_h}(v, \alpha) - d_\pi^{P_h}(v, \beta)\} = d_\pi(v, \alpha) - d_\pi(v, \beta) \qquad (\forall v \in V). \tag{5.1}$$

Note that an alternating walk decomposition is not uniquely determined. An alternating walk decomposition always exists, and can be obtained by the following algorithm.

Step 0: Set $s := 0$ and $E' := E_\pi^\alpha \cup E_\pi^\beta$.
Step 1: If $d_\pi^{E'}(v, \alpha) = d_\pi^{E'}(v, \beta)$ ($\forall v \in V$), then output $\{P_1, P_2, \ldots, P_s, E'\}$ and stop.
Step 2: Let $v \in V$ be a vertex with $d_\pi^{E'}(v, \alpha) \neq d_\pi^{E'}(v, \beta)$.
Step 3: Find an eligible alternating walk P_{s+1} in the multigraph (V, E') starting from v.
Step 4: Set $E' := E' \setminus P_{s+1}$ and $s := s + 1$. Go to Step 1.

It is not difficult to implement this algorithm so that it runs in $O(|E_\pi^\alpha \cup E_\pi^\beta|)$ time.

We now prove Lemma 2.4. Suppose that there exist two distinct colors $\alpha, \beta \in \mathcal{C}$ and a vertex $u \in V$ such that $d_\pi(u, \alpha) - d_\pi(u, \beta) \geq 3$. Let $\{P_1, P_2, \ldots, P_s, R\}$ be an alternating walk decomposition of $E_\pi^\alpha \cup E_\pi^\beta$. In the following, we show that there exists a subset $\mathcal{P} \subseteq \{P_1, P_2, \ldots, P_s\}$ of alternating walks such that the set $S = \bigcup_{P \in \mathcal{P}} P$ satisfies the conditions (S1), (S2), and

$$d_\pi^S(u, \alpha) - d_\pi^S(u, \beta) \in \{r, r+1\}, \tag{5.2}$$

where r is an integer with $1 \leq r \leq d_\pi(u, \alpha) - d_\pi(u, \beta) - 2$. We note that for any $\mathcal{P} \subseteq \{P_1, P_2, \ldots, P_s\}$, the set $S = \bigcup_{P \in \mathcal{P}} P$ is eligible since $\{P_1, P_2, \ldots, P_s, R\}$ is an alternating walk decomposition. The proof given below is constructive, and it immediately yields an algorithm for computing an eligible edge set satisfying the desired conditions in $O(|E_\pi^\alpha \cup E_\pi^\beta|)$ time.

We first consider the condition (5.2). We assume that $P_1, \ldots, P_{s'}$ ($s' \geq 0$) are the alternating walks such that both of the end vertices are u, and $P_{s'+1}, \ldots, P_{s''}$ ($s'' \geq s'$) are the alternating walks such that only one of the end vertices is u. We start with $\mathcal{P} = \emptyset$, and add the walks $P_1, P_2, \ldots, P_{\min\{s', \lceil r/2 \rceil\}}$ to the set $\mathcal{P}$. If $s' \geq \lceil r/2 \rceil$, then the edge set $S = \bigcup_{P \in \mathcal{P}} P$ satisfies

$$d_\pi^S(u, \alpha) - d_\pi^S(u, \beta) = 2\lceil r/2 \rceil \in \{r, r+1\};$$

i.e., (5.2) holds. Otherwise (i.e., $s' < \lceil r/2 \rceil$), we further add the walks $P_{s'+1}, P_{s'+2}, \ldots, P_{s'+(r-2s')}$ to $\mathcal{P}$. Then, $S = \bigcup_{P \in \mathcal{P}} P$ satisfies (5.2). We note that $s' + (r - 2s') \leq s''$ holds since

$$2s' + (s'' - s') = d_\pi(u, \alpha) - d_\pi(u, \beta) > r.$$

9

We then consider the property (S1). We note that none of walks in the current set $\mathcal{P}$ is a β-odd alternating walk since every eligible alternating walk starting from the vertex u is either an $\alpha\beta$-even alternating walk or an α-odd alternating walk. Let t_α be the number of α-odd alternating walks in $\mathcal{P}$, and define t_β by

$$
t_\beta = \begin{cases} \max\{0, t_\alpha - 1\} & \text{if } |E_\pi^\alpha| = |E_\pi^\beta| + 1, \\ t_\alpha & \text{if } |E_\pi^\alpha| = |E_\pi^\beta| - 1 \text{ or } |E_\pi^\alpha| = |E_\pi^\beta|. \end{cases}
$$

We see from the following simple observation that the number of β-odd alternating walks in $\{P_1, P_2, \ldots, P_s\}$ is at least t_β.

Lemma 5.2. *Let $\{P_1, P_2, \ldots, P_s, R\}$ be an alternating walk decomposition of $E_\pi^\alpha \cup E_\pi^\beta$, and let s_α (resp., s_β) be the number of α-odd (resp., β-odd) alternating walks in $\{P_1, P_2, \ldots, P_s\}$. Then, we have $s_\alpha - s_\beta = |E_\pi^\alpha| - |E_\pi^\beta|$.*

We choose t_β β-odd alternating walks in the decomposition arbitrarily and add them to $\mathcal{P}$. Note that u cannot be an end vertex of a β-odd alternating walk, and hence the addition of β-odd alternating walks does not affect the condition (5.2). Therefore, the edge set $S = \bigcup_{P \in \mathcal{P}} P$ satisfies both of (5.2) and (S1).

Finally, we consider the condition (S2). We use a similar technique as in [5, 7, 8]. Let $G_\pi^*(\alpha, \beta)$ be a subgraph of $G_\pi(\alpha, \beta)$ defined as follows. From the multigraph $G_\pi(\alpha, \beta)$, delete successively all pairs of edges of color α and β respectively connecting the same two vertices as far as such a pair of edges exists, and let $G_\pi^*(\alpha, \beta) = (V, E^*)$ be the resulting multigraph. Obviously, for each pair of vertices v, v' there exists at most one edge connecting v and v'; an edge (v, v') with color α (resp., β) is in E^* if and only if $|E_\pi^\alpha(v, v')| = |E_\pi^\beta(v, v')| + 1$ (resp., $|E_\pi^\beta(v, v')| = |E_\pi^\alpha(v, v')| + 1$). Hence, any subset S of E^* satisfies the condition (S2). This means that if we consider an edge set of the graph $G_\pi^*(\alpha, \beta)$ instead of the original graph $G_\pi(\alpha, \beta)$, the condition (S2) is automatically satisfied. This modification does not affect (S1) since $G_\pi^*(\alpha, \beta)$ is obtained by removing the same number of edges from E_π^α and from E_π^β. Moreover, we have

$$
d_\pi^{E^*}(v, \alpha) - d_\pi^{E^*}(v, \beta) = d_\pi(v, \alpha) - d_\pi(v, \beta) \qquad (\forall v \in V).
$$

This implies that the conditions concerning the balance around each vertex such as eligibility condition (2.1) and the conditions (5.1) and (5.2) are not affected by the replacement of $G_\pi(\alpha, \beta)$ with $G_\pi^*(\alpha, \beta)$. In summary, this replacement of the multigraph does not affect the properties shown in the previous discussion. This concludes the proof of Lemma 2.4.

6   Conclusion

In this paper, we proposed a new algorithm for the nearly equitable edge coloring problem. Our algorithm FASTBALANCING computes a nearly equitable edge coloring of a multigraph satisfying the conditions (B1) and (B2) in $\text{O}(\min\{mn, m^2/k\})$ time. The time complexity of our algorithm is better than those of the previous algorithms. Moreover, our algorithm is the first to compute a nearly equitable edge coloring satisfying both of the conditions (B1) and (B2).

References

[1] J. Correa and M. X. Goemans, Improved bounds on nonblocking 3-stage Clos networks, *SIAM Journal on Computing* **37** (2007) 870–894.

[2] J. K. Dugdale and A. J. W. Hilton, Amalgamated factorizations of complete graphs, *Combinatorics, Probability, and Computing* **3** (1994) 215–231.

[3] U. Feige and M. Singh, Edge coloring and decompositions of weighted graphs, *Proceedings of ESA 2008*, Lecture Notes in Computer Science **5193**, 405–416.

[4] M. K. Goldberg, Edge-coloring of multigraphs: recoloring techniques, *Journal of Graph Theory* **8** (1984) 122–136.

[5] A.J.W. Hilton and D. de Werra, Sufficient conditions for balanced and for equitable edge-colouring of graphs, O. R. Working paper 82/3, Département de Mathématiques, École Polytechnique Fédérate de Lausanne, Switzerland, 1982.

[6] A.J.W. Hilton and D. de Werra, A sufficient condition for equitable edge-colourings of simple graphs, *Discrete Mathematics* **128** (1994) 179–201.

[7] S. Nakano and T. Nishizeki, Scheduling file transfers under port and channel constraints, *International Journal of Foundations of Computer Science* **4** (1993) 101–115..

[8] S. Nakano, T. Nishizeki, and N. Saito, On the *fg*-coloring of graphs, *Combinatorica* **10** (1990) 67–80.

[9] S. Nakano, Y. Suzuki, and T. Nishizeki, An algorithm for the nearly equitable edge-coloring of graphs (in Japanese), *IEICE Transactions on Information and Systems* **J78-D-I** (1995) 437–444.

[10] T. Ono, T. Hirata, An improved algorithm for the net assignment problem, *IEICE Transactions on Fundamentals* **E84-A** (2001) 1161–1165.

[11] H. Song, J. Wu, and G. Liu, The equitable edge-coloring of series-parallel graphs, *Proceedings of ICCS 2007*, Part III, Lecture Notes in Computer Science **4489**, 457–460.

[12] D. de Werra, Equitable colorations of graphs, *Revue française d'Informatique et de Recherche Operationelle* **R-3** (1971) 3–8.

[13] D. de Werra, Some results in chromatic scheduling, *Zeitschrift für Operations Research* **18** (1974) 167–175.

[14] D. de Werra, An extension of bipartite multigraphs, *Discrete Mathematics* **14** (1976) 133–138.

[15] D. de Werra, On the use of alternating chains and hypergraphs in edge coloring, *J. Graph Theory* **3** (1979) 175–182.

[16] X. Xie, T. Ono, S. Nakano, and T. Hirata, An improved algorithm for the nearly equitable edge-coloring problem, *IEICE Transactions on Fundamentals* **E87-A** (2004) 1029–1033.

[17] X. Xie, M. Yagiura, T. Ono, T. Hirata, and U. Zwick, An efficient algorithm for the nearly equitable edge coloring problem, *Journal of Graph Algorithms and Applications* **12** (2008) 383–399.

Appendix: Proof of Proposition 4.1

Define $a', b' \in \mathbb{Z}$ by

$$(a', b') = \begin{cases} (a - c, b + c) & \text{if } a - c \geq b + c, \\ (b + c, a - c) & \text{if } a - c < b + c. \end{cases}$$

Then, it suffices to show that $\{\varphi_z(a) + \varphi_z(b)\} - \{\varphi_z(a') + \varphi_z(b')\} \geq 1$. Note that that $a > a' \geq b' > b$ and $a - a' = b' - b \geq 1$. If $a' \geq z \geq b'$, then we have

$$\{\varphi_z(a) + \varphi_z(b)\} - \{\varphi_z(a') + \varphi_z(b')\}$$
$$= \{(a - \lceil z \rceil) + (\lfloor z \rfloor - b)\} - \{(a' - \lceil z \rceil) + (\lfloor z \rfloor - b')\} = (a - b) - (a' - b') \geq 2.$$

If $a > z > a'$, then we have

$$\{\varphi_z(a) + \varphi_z(b)\} - \{\varphi_z(a') + \varphi_z(b')\}$$
$$= \{(a - \lceil z \rceil) + (\lfloor z \rfloor - b)\} - \{(\lfloor z \rfloor - a') + (\lfloor z \rfloor - b')\} = 2a - \lceil z \rceil - \lfloor z \rfloor \geq 1,$$

where the inequality is by $a > z \geq \lfloor z \rfloor$. The case with $b' > z > b$ can be shown in the same way as the case with $a > z > a'$.