

## 多重グラフにおける均等辺彩色を求める高速アルゴリズム

塩浦 昭義 (東北大学) 柳浦 睦憲 (名古屋大学)

本稿では、多重グラフにおける均等辺彩色問題について議論し、この問題に対する効率的なアルゴリズムを提案する。提案するアルゴリズムの計算複雑度は、既存のアルゴリズムより改善されたものとなっている。提案するアルゴリズムにより求められる彩色は、枝の数のバランスに関する付加的な2つの条件を満たす。ここでは、多重グラフの全ての枝に関するバランス、および各頂点間の並列枝に関するバランスを考慮している。これまで、これらのバランス条件を同時に満たすアルゴリズムは存在しなかった。この改善を実現するために、本稿では新たな彩色更新法を提案する。既存のアルゴリズムでは、彩色を更新するためにオイラー閉路や1本の交互歩道を用いていたが、提案する彩色更新法では、互いに辺素な複数の交互歩道を利用する。この新しい彩色更新法により、計算複雑度の削減が可能となる。

### A Fast Algorithm for Computing a Nearly Equitable Edge Coloring on a Multigraph

Akiyoshi Shioura (Tohoku University) Mutsunori Yagiura (Nagoya University)

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

**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  $\mathcal{C} = \{1, 2, \dots, k\}$  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  $\pi : E \rightarrow \mathcal{C}$ .

Let  $\pi : E \rightarrow \mathcal{C}$  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  $\pi$  of a multigraph  $G$  is *nearly equitable* if it satisfies the condition

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

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.

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

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

Table 1: Comparison of algorithms for the nearly equitable edge coloring problem. The mark “√” 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) [2]	$O(km^2)$		√	single alternating walk
Nakano et al. (1995) [6]	$O(m^2/k + mn)$			single alternating walk
Xie et al. (2004) [10]	$O(m^2/k)$	√		Eulerian circuit
Xie et al. (2008) [11]	$O(mn \log(m/(nk) + 1))$	√		Eulerian circuit
Ours	$O(\min\{mn, m^2/k\})$	√	√	edge-disjoint alternating walks

where  $\pi$  is an edge coloring and

$$E_\pi^i = \{e \in E \mid \pi(e) = i\} \quad (i \in \mathcal{C}),$$

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

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. 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.** *Our algorithm computes a nearly equitable edge coloring of a multigraph satisfying the conditions (B1) and (B2) in  $O(\min\{mn, m^2/k\})$  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  $O(mn \log(m/(nk) + 1))$  by Xie et al. [11]. Moreover, our algorithm is the first to compute a nearly equitable edge coloring satisfying both of the conditions (B1) and (B2). The algorithms in [10, 11] outputs a nearly equitable edge coloring satisfying (B1), and the output of the algorithm in [2] satisfies (B2), but none of the previous algorithms is guaranteed to obtain a coloring satisfying both of (B1) and (B2) (see Table 1).

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 [10, 11] or a single alternating walk [2, 6] (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.

**Previous and Related Work** An edge coloring  $\pi$  of a multigraph  $G$  is said to be *equitable* if it satisfies the condition  $|d_\pi(v, i) - d_\pi(v, j)| \leq 1$  ( $\forall i, j \in \mathcal{C}, \forall v \in V$ ), 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., [3, 9]). A typical example is an odd cycle, which has no equitable edge coloring with  $k = 2$ . Several sufficient conditions for graphs to have an equitable edge coloring are shown in [2, 3, 8]. Note that the problem of determining the existence of an equitable edge coloring is NP-complete (see [11]).

The balanced conditions (B1) and (B2) have often been discussed in the literature of (nearly) equitable edge coloring [1, 2, 3, 9, 10, 11]. The first condition (B1) is referred to as “equalized condition” in [1] and “balanced condition” in [7, 10, 11], and the second condition (B2) is referred to as “edge-balanced condition” in [2].

**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 improves an edge coloring by *switching edge colors of alternating walks* (see, e.g., [4]); 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  $\alpha$  and  $\beta$  of edges in  $S$ ; more formally, switching edge colors of  $S$  is to modify the current edge coloring  $\pi : E \rightarrow \mathcal{C}$  to the new edge coloring  $\pi' : E \rightarrow \mathcal{C}$  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:

$$\left\{ \begin{array}{l} \text{if } d_\pi(v, \alpha) \geq d_\pi(v, \beta), \text{ then } 0 \leq d_\pi^S(v, \alpha) - d_\pi^S(v, \beta) \leq d_\pi(v, \alpha) - d_\pi(v, \beta), \\ \text{if } d_\pi(v, \alpha) \leq d_\pi(v, \beta), \text{ then } 0 \geq d_\pi^S(v, \alpha) - d_\pi^S(v, \beta) \geq d_\pi(v, \alpha) - d_\pi(v, \beta), \end{array} \right\} \quad (1)$$

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 (1) for all  $v \in V$ . Eligible edge sets are useful in getting a better edge coloring, as shown below.

**Lemma 2.** *Let  $\pi : E \rightarrow \mathcal{C}$  be an edge coloring and  $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$  an eligible edge set. Then, the new edge coloring  $\pi' : E \rightarrow \mathcal{C}$  obtained by switching edge colors of  $S$  satisfies*

$$\begin{aligned} \min\{d_\pi(v, \alpha), d_\pi(v, \beta)\} &\leq \min\{d_{\pi'}(v, \alpha), d_{\pi'}(v, \beta)\} \\ &\leq \max\{d_{\pi'}(v, \alpha), d_{\pi'}(v, \beta)\} \leq \max\{d_\pi(v, \alpha), d_\pi(v, \beta)\} \quad (\forall v \in V). \end{aligned}$$

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

- (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  $|S \cap E_\pi^\alpha| - |S \cap E_\pi^\beta| = 0, +1$ , or  $-1$ ,  
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  $|S \cap E_\pi^\alpha(u, v)| - |S \cap E_\pi^\beta(u, v)| = 0, +1$ , or  $-1$ ,  
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 3.** *Let  $\pi : E \rightarrow \mathcal{C}$  be an edge coloring satisfying (B1) and (B2), and  $\pi' : E \rightarrow \mathcal{C}$  be the new edge coloring obtained by switching edge colors of an edge set  $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$  satisfying (S1) and (S2). Then,  $\pi'$  also satisfies (B1) and (B2).*

The following is one of the key properties used in our algorithm, which is proven in Section 5.

**Lemma 4.** *Let  $\pi : E \rightarrow \mathcal{C}$  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  $O(|E_\pi^\alpha \cup E_\pi^\beta|)$  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  $O(m)$  time by using the following property.

**Proposition 5.** *Let  $\{e_1, e_2, \dots, e_m\}$  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  $e_t$  ( $t = 1, 2, \dots, m$ ) by the color  $(t \bmod k) + 1$ . Then, the resulting edge coloring satisfies the conditions (B1) and (B2).*

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  $\pi$  satisfying the condition (NEC), our algorithm processes each vertex  $u \in V$  one by one. If the vertex  $u$  violates the condition

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

then the algorithm repeatedly updates the edge coloring  $\pi$  by switching edge colors of an eligible edge set  $S$  until the condition (2) is satisfied. By Lemma 2, once the vertex  $u$  satisfies the condition (2), the edge coloring always satisfies (2) in the following iterations.

Suppose that the vertex  $u$  violates the condition (2). Our algorithm implicitly maintains the following sets of colors:

$$\mathcal{C}_\pi^0(u) = \{i \in \mathcal{C} \mid \lceil d(u)/k \rceil - 1 \leq d_\pi(u, i) \leq \lfloor d(u)/k \rfloor + 1\}, \quad (3)$$

$$\mathcal{C}_\pi^+(u) = \{i \in \mathcal{C} \mid d_\pi(u, i) \geq \lfloor d(u)/k \rfloor + 2\}, \quad (4)$$

$$\mathcal{C}_\pi^-(u) = \{i \in \mathcal{C} \mid d_\pi(u, i) \leq \lceil d(u)/k \rceil - 2\}. \quad (5)$$

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  $\alpha, \beta$  with  $\alpha \in \mathcal{C}_\pi^+(u)$  and  $\beta \in \mathcal{C}_\pi^-(u)$ , which is done by choosing  $\alpha$  and  $\beta$  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  $\pi$  so that at least one of  $\alpha$  and  $\beta$  is contained in  $\mathcal{C}_\pi^0(u)$ . This can be done efficiently by Lemma 4 with the value  $r$  given by

$$r = \min\{d_\pi(u, \alpha) - (\lfloor d(u)/k \rfloor + 1), (\lceil d(u)/k \rceil - 1) - d_\pi(u, \beta)\}. \quad (6)$$

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 (6) is zero. Then, the algorithm iteratively updates the edge coloring  $\pi$  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  $\alpha$  and  $\beta$  with the same rule as above, and then using Lemma 4 with  $r = 1$ . In this way, the algorithm computes an edge coloring  $\pi$  satisfying (2).

Our algorithm is described as follows.

**Algorithm** FASTBALANCING( $G, \mathcal{C}$ )

**Input:** a multigraph  $G = (V, E)$  and a set of colors  $\mathcal{C} = \{1, 2, \dots, k\}$ .

**Output:** a nearly equitable edge coloring  $\pi : E \rightarrow \mathcal{C}$  of  $G$  satisfying (B1) and (B2).

1. Compute an initial edge coloring  $\pi$  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  $|d_\pi(u, i) - d_\pi(u, j)| \geq 3$  **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  $\pi$  by switching edge colors of  $S$ .
8.     Output  $\pi$  and stop.

We note that an eligible edge set  $S$  in Line 6 can always be obtained by Lemma 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  $\pi$  always satisfies (B1) and (B2) by Lemma 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  $\varphi_z : \mathbb{R} \rightarrow \mathbb{R}$  defined by  $\varphi_z(x) = \max\{\lfloor z \rfloor - x, 0, x - \lceil z \rceil\}$  ( $x \in \mathbb{R}$ ), where  $z \in \mathbb{R}$  is a real number.

**Proposition 6.** *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  $\varphi_z(a - c) + \varphi_z(b + c) \leq \varphi_z(a) + \varphi_z(b) - 1$ .*

For an edge coloring  $\pi : E \rightarrow \mathcal{C}$  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  $\pi$ , 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  $\pi$  at the vertex  $u$ . Using Proposition 6, we can show that the value  $\Phi(\pi, u)$  is monotonically decreasing in each iteration of the algorithm.

**Lemma 7.** *Let  $\pi$  be an edge coloring,  $u \in V$  be a vertex, and  $\alpha, \beta \in \mathcal{C}$  be distinct colors such that  $d_\pi(u, \alpha) = \max_{i \in \mathcal{C}} d_\pi(u, i)$ ,  $d_\pi(u, \beta) = \min_{i \in \mathcal{C}} d_\pi(u, i)$ , and  $d_\pi(u, \alpha) - d_\pi(u, \beta) \geq 3$ . Suppose that  $\pi'$  is an edge coloring obtained by switching edge colors of an eligible edge set  $S \subseteq E_\pi^\alpha \cup E_\pi^\beta$  with*

$$1 \leq d_\pi^S(u, \alpha) - d_\pi^S(u, \beta) \leq d_\pi(u, \alpha) - d_\pi(u, \beta) - 1. \quad (7)$$

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

Lemma 7 implies that the number of iterations in the while loop in the algorithm FASTBALANCING is  $O(d(u))$  for a fixed vertex  $u \in V$  since  $\Phi(\pi, u) = O(d(u))$ . In fact, we can obtain the following better bound.

**Lemma 8.** *For a fixed vertex  $u \in V$ , the number of iterations in the while loop in the algorithm FASTBALANCING is  $O(\min\{k, d(u)\})$ .*

*Outline of 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), (4), and (5), respectively. Suppose that the colors  $\alpha$  and  $\beta$  chosen in Line 5 satisfy  $\alpha \in \mathcal{C}_\pi^+(u)$ ,  $\beta \in \mathcal{C}_\pi^-(u)$ . Recall that  $\alpha$  and  $\beta$  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  $\alpha$  and  $\beta$  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  $\Phi(\pi, u) \leq 2k$ , and therefore Lemma 7 implies that the while loop terminates in at most  $2k$  iterations. This concludes the proof.  $\square$

By Lemma 8, the number of iterations of Lines 5–7 for a fixed vertex  $u \in V$  is  $O(\min\{k, d(u)\})$ . We can compute an eligible edge set  $S$  satisfying the desired conditions in  $O(|E_\pi^\alpha \cup E_\pi^\beta|) = O(m/k)$  time by Lemma 4. Switching edge colors in Line 7 requires  $O(|S|) = O(m/k)$  time. Maintenance of values  $d_\pi(u, i)$  and Line 5 can be done in  $O(m)$  time in total by using a data structure shown in [11, Section 3]. Hence, the algorithm FASTBALANCING computes a nearly equitable edge coloring of a multigraph satisfying the conditions (B1) and (B2) in  $O((m/k) \times \sum_{u \in V} \min\{k, d(u)\}) = O(\min\{mn, m^2/k\})$  time. This concludes the proof of Theorem 1.

## 5 Computing Eligible Edge Sets

In this section we give a proof of Lemma 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  $O(|E_\pi^\alpha \cup E_\pi^\beta|)$  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  $u_0 e_1 u_1 e_2 u_2 \dots e_{t-1} u_{t-1} e_t u_t$ , where  $u_0, u_1, \dots, u_t$  are vertices and  $e_1, e_2, \dots, e_t$  are distinct edges such that  $e_j$  connects the vertices  $u_{j-1}$  and  $u_j$  for  $j = 1, 2, \dots, t$ . 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  $\{e_1, e_2, \dots, e_t\}$  to simplify the description.

Let  $\pi : E \rightarrow \mathcal{C}$  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  $\beta$ -*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 9** ([4, 5, 6]). *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  $P = u_0 e_1 u_1 e_2 u_2 \dots e_{t-1} u_{t-1} e_t u_t$  starting from  $u_0$ .*

A partition  $\{P_1, P_2, \dots, P_s, R\}$  ( $s \geq 0$ ) 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  $P_h$  ( $h = 1, 2, \dots, s$ ) are eligible alternating walks satisfying the following condition:

$$\sum_{h=1}^s \{d_\pi^{P_h}(v, \alpha) - d_\pi^{P_h}(v, \beta)\} = d_\pi(v, \alpha) - d_\pi(v, \beta) \quad (\forall v \in V). \quad (8)$$

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  $\{P_1, P_2, \dots, P_s, E'\}$  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 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  $\{P_1, P_2, \dots, P_s, R\}$  be an alternating walk decomposition of  $E_\pi^\alpha \cup E_\pi^\beta$ . In the following, we show that there exists a subset  $\mathcal{P} \subseteq \{P_1, P_2, \dots, P_s\}$  of alternating walks such that the set  $S = \bigcup_{P \in \mathcal{P}} P$  satisfies the conditions (S1), (S2), and

$$d_\pi^S(u, \alpha) - d_\pi^S(u, \beta) \in \{r, r + 1\}, \quad (9)$$

where  $r$  is an integer with  $1 \leq r \leq d_\pi(u, \alpha) - d_\pi(u, \beta) - 2$ . We note that for any  $\mathcal{P} \subseteq \{P_1, P_2, \dots, P_s\}$ , the set  $S = \bigcup_{P \in \mathcal{P}} P$  is eligible since  $\{P_1, P_2, \dots, P_s, R\}$  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 (9). We assume that  $P_1, \dots, P_{s'}$  ( $s' \geq 0$ ) are the alternating walks such that both of the end vertices are  $u$ , and  $P_{s'+1}, \dots, P_{s''}$  ( $s'' \geq s'$ ) are the alternating walks such that only one of the end vertices is  $u$ . We start with  $\mathcal{P} = \emptyset$ , and add the walks  $P_1, P_2, \dots, P_{\min\{s', \lceil r/2 \rceil\}}$  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., (9) holds. Otherwise (i.e.,  $s' < \lceil r/2 \rceil$ ), we further add the walks  $P_{s'+1}, P_{s'+2}, \dots, P_{s'+(r-2s')}$  to  $\mathcal{P}$ . Then,  $S = \bigcup_{P \in \mathcal{P}} P$  satisfies (9). We note that  $s' + (r - 2s') \leq s''$  holds since

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

We then consider the property (S1). We note that none of walks in the current set  $\mathcal{P}$  is a  $\beta$ -odd alternating walk since every eligible alternating walk starting from the vertex  $u$  is either an  $\alpha\beta$ -even alternating walk or an  $\alpha$ -odd alternating walk. Let  $t_\alpha$  be the number of  $\alpha$ -odd alternating walks in  $\mathcal{P}$ , and define  $t_\beta$  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  $\beta$ -odd alternating walks in  $\{P_1, P_2, \dots, P_s\}$  is at least  $t_\beta$ .

**Lemma 10.** *Let  $\{P_1, P_2, \dots, P_s, R\}$  be an alternating walk decomposition of  $E_\pi^\alpha \cup E_\pi^\beta$ , and let  $s_\alpha$  (resp.,  $s_\beta$ ) be the number of  $\alpha$ -odd (resp.,  $\beta$ -odd) alternating walks in  $\{P_1, P_2, \dots, P_s\}$ . Then, we have  $s_\alpha - s_\beta = |E_\pi^\alpha| - |E_\pi^\beta|$ .*

We choose  $t_\beta$   $\beta$ -odd alternating walks in the decomposition arbitrarily and add them to  $\mathcal{P}$ . Note that  $u$  cannot be an end vertex of a  $\beta$ -odd alternating walk, and hence the addition of  $\beta$ -odd alternating walks does not affect the condition (9). Therefore, the edge set  $S = \bigcup_{P \in \mathcal{P}} P$  satisfies both of (9) and (S1).

Finally, we consider the condition (S2). We use a similar technique as in [2, 4, 5]. 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  $\alpha$  and  $\beta$  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  $\alpha$  (resp.,  $\beta$ ) 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  $E_\pi^\alpha$  and from  $E_\pi^\beta$ . Moreover, we have

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

This implies that the conditions concerning the balance around each vertex such as eligibility condition (1) and the conditions (8) and (9) 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 4.

## References

- [1] J. K. Dugdale and A. J. W. Hilton, Amalgamated factorizations of complete graphs, *Combinatorics, Probability, and Computing* **3** (1994) 215–231.
- [2] 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 Fédérative de Lausanne, Switzerland, 1982.
- [3] A.J.W. Hilton and D. de Werra, A sufficient condition for equitable edge-colourings of simple graphs, *Discrete Mathematics* **128** (1994) 179–201.
- [4] S. Nakano and T. Nishizeki, Scheduling file transfers under port and channel constraints, *International J. Foundations of Computer Science* **4** (1993) 101–115..
- [5] S. Nakano, T. Nishizeki, and N. Saito, On the  $fg$ -coloring of graphs, *Combinatorica* **10** (1990) 67–80.
- [6] S. Nakano, Y. Suzuki, and T. Nishizeki, An algorithm for the nearly equitable edge-coloring of graphs (in Japanese), *IEICE Trans. Information and Systems* **J78-D-I** (1995) 437–444.
- [7] T. Ono, T. Hirata, An improved algorithm for the net assignment problem, *IEICE Trans. Fundamentals* **E84-A** (2001) 1161–1165.
- [8] H. Song, J. Wu, and G. Liu, The equitable edge-coloring of series-parallel graphs, *Proceedings of the International Conference on Computational Science (ICCS 2007)*, Part III, Beijing, China, May 27–30, 2007, Lecture Notes in Computer Science **4489**, 457–460.
- [9] D. de Werra, Some results in chromatic scheduling, *Zeitschrift für Operations Research* **18** (1974) 167–175.
- [10] X. Xie, T. Ono, S. Nakano, and T. Hirata, An improved algorithm for the nearly equitable edge-coloring problem, *IEICE Trans. Fundamentals* **E87-A** (2004) 1029–1033.
- [11] X. Xie, M. Yagiura, T. Ono, T. Hirata, and U. Zwick, An efficient algorithm for the nearly equitable edge coloring problem, *J. Graph Algorithms and Applications*, **12** (2008) 383–399.