PAPER Special Section on Discrete Mathematics and Its Applications

An Improved Algorithm for the Nearly Equitable Edge-Coloring Problem

Xuzhen XIE[†a)], **Takao ONO**[†], *Nonmembers*, **Shin-ichi NAKANO**[††], *and* **Tomio HIRATA**[†], *Members*

SUMMARY A nearly equitable edge-coloring of a multigraph is a coloring such that edges incident to each vertex are colored equitably in number. This problem was solved in $O(kn^2)$ time, where n and k are the numbers of the edges and the colors, respectively. The running time was improved to be $O(n^2/k + n|V|)$ later. We present a more efficient algorithm for this problem that runs in $O(n^2/k)$ time.

key words: *nearly equitable edge coloring, Euler circuit*

1. Introduction

A nearly equitable edge-coloring of a multigraph $G = (V, E)$ is a coloring such that edges incident to each vertex are colored equitably in number with given colors. Hilton and Werra [1] solved this problem in $O(kn^2)$ time in 1982, where n and k are the numbers of the edges and the colors, respectively. Later in 1995, Nakano, Suzuki and Nishizeki [2] presented a new algorithm that runs in $O(n^2/k + n|V|)$ time. Using the idea of "balanced constraint" [2], Ono and Hirata [3] transformed a restricted case of the Net Assignment Problem into a Balanced m-edge Coloring Problem, where m is the bound of the edges colored with any given color for each vertex. They presented an algorithm for the Balanced m-edge Coloring Problem of $O(n^2/k)$ time. Here, using Ono and Hirata's technique, we present a more efficient algorithm for the Nearly Equitable Edge-coloring Problem. The new algoirthm also runs in $O(n^2/k)$ time and satisfies "balanced constraint."

The rest of the paper is structured as follows. In Sect. 2, we give some definitions. In Sect. 3, we introduce the results of Hilton and Werra. Sections 4 and 5 are for the new algorithm and the results we obtained. Finally, concluding remarks is in Sect. 6.

2. Preliminaries and Definitions

Let us introduce the graph-theoretic notation that will be used throughout this paper.

For a multigraph G, let V denote the vertices of G, E denote the edges of G, and n denote the number of the edges. We use $d_G(v)$ to denote the degree of a vertex v and C_i-edge an edge with color C_i. $d_G(v, C_i)$ stands for the number of C_i-edges incident to a vertex v in G, $e_G(C_i)$ stands for the number of C_i-edges in G and $G_{C_iC_j}$ is the subgraph of G induced by all the C_i-edges and C_j-edges in G. We omit the subscript G if it is clear from the context.

Given a multigraph $G = (V, E)$ and a k-color set $C = \{C_1, C_2, \cdots, C_k\}$, the Nearly Equitable Edge-coloring is an edge-coloring of G with the k colors such that for any vertex $v \in V$ and diferent colors $C_i, C_j \in C$, $|d(v, C_i) - d(v, C_j)| \leq 2$ [1].

3. $O(kn^2)$-Time for Nearly Equitable Edge-Coloring

The Nearly Equitable Edge-coloring Problem was solved by Hilton and Werra [1] in 1982. Using Euler circuit, they presented a simple algorithm of $O(kn^2)$ time with k colors. We have a brief introduction of their algorithm in the following.

Assign the given multigraph G with the given k colors. Whenever there exists $v \in V$ and different colors $C_i, C_j \in C$ such that $|d(v, C_i) - d(v, C_j)| > 2$, add a new vertex w adjacent to all odd-degree vertices in $G_{C_iC_j}$ to form a new graph $G'_{C_iC_j}$. For any connected component in $G'_{C_iC_j}$, traverse an Euler circuit and assign colors C_i and C_j alternately along the way, and delete the edges adjacent to the new vertex w.

The multigraph G is nearly equitably edge-colored, that is, for any $v \in V$ and different colors $C_i, C_j \in C$, $|d(v, C_i) - d(v, C_j)| \leq 2$ when the algorithm stops. The running time of the algorithm is proved to be $O(kn^2)$.

Define

$$Cost = \sum_{v \in V} \sum_{C_i \in C} \sum_{C_j \in C} |d(v, C_i) - d(v, C_j)|,$$

then

$$\begin{aligned} Cost &\leq \sum_{v \in V} \sum_{C_i \in C} \sum_{C_j \in C} d(v, C_i) \\ &= \sum_{C_j \in C} \{ \sum_{v \in V} \sum_{C_i \in C} d(v, C_i) \} \\ &= \sum_{C_j \in C} 2|E| = 2kn. \end{aligned}$$

$Cost$ decreases by at least 2 each time the Euler circuit is traversed. Each Euler circuit costs $O(|E|) = O(n)$ time, so after at most kn traverses of Euler circuits, $Cost$ must be 0 and the algorithm runs in $O(kn^2)$ time.

Euler circuit is normally used for edge-coloring [1]–[5]. We also use it for our new algorithm in the following.

Manuscript received August 22, 2003.
Manuscript revised November 14, 2003.
Final manuscript received January 22, 2004.

[†]The authors are with the Graduate School of Information Science, Nagoya University, Nagoya-shi, 464-8603 Japan.
[††]The author is with the Department of Computer Science, Faculty of Engineering, Gunma University, Kiryu-shi, 376-8515 Japan.
a) E-mail: sharryx@hirata.nuee.nagoya-u.ac.jp

4. A New Algorithm for Nearly Equitable Edge-Coloring

4.1 Algorithm (G, C)

Ono and Hirata [3] presented an $O(n^2/k)$-time algorithm for the Balanced m-edge Coloring Problem. Here, we use the same technique for the Nearly Equitable Edge-coloring Problem.

```
Algorithm(G, C)
    Input: a multigraph G = (V, E) with |E| = n and a
    color set C with |C| = k
    Output: a nearly equitable edge-coloring for G
    Assign C_1, C_2, ..., C_k to n edges repeatedly, so that
    each color class has ⌈n/k⌉ or ⌊n/k⌋ edges.
    while there exists v ∈ V and different colors
    C_i, C_j ∈ C such that |d(v, C_i) − d(v, C_j)| ≥ 3 do
        for the vertex v, find α, β ∈ C with

            d(v, α) = max{d(v, C_i) : C_i ∈ C},
            d(v, β) = min{d(v, C_i) : C_i ∈ C}.

        Recolor(G_αβ, α, β, v).
```

To maintain the inequality $|e_G(\alpha) - e_G(\beta)| \leq 1$, we recolor all connected components of $G_{\alpha\beta}$. For this purpose, we swap α and β each time the odd connected component of $G_{\alpha\beta}$ is recolored.

```
Recolor(G_αβ, α, β, v)
    Input: a multigraph G_αβ with all edges colored
    with colors α and β and a selected vertex v
    Output: a nearly equitable edge-coloring for G_αβ
    Let x ← α and y ← β.
    for each connected component H in G_αβ do
        Recolor-Component(H, x, y, v).
        if H has odd number of edges then
            ▷ (in this case e_H(x) = e_H(y) + 1)
            Swap x and y.
```

```
Recolor-Component(H, x, y, v)
    Input: a connected component H with all edges
    colored with colors x and y and a selected vertex v
    Output: a nearly equitable edge-coloring for H,
    satisfying e_H(y) ≤ e_H(x) ≤ e_H(y) + 1
    if H has odd-degree vertices then
        Add a new vertex w adjacent to all the odd-
        degree vertices to form a new graph H'.
        Traverse an Euler circuit starting at the vertex
        w and assign colors y and x (y comes first) to
        the edges alternately along the way.
    else
        if v is a vertex of H then
            Let v be the start vertex u.
        else
            if there exists a vertex r ∈ H with |d(r, x) −
            d(r, y)| ≥ 2 then
                Let r be the start vertex u.
            else
                Let u be an arbitrary vertex.
        Traverse an Euler circuit starting at the vertex u
        and assign colors x and y (x comes first) to the
        edges alternately along the way.
```

5. Analysis of the Algorithm

5.1 Results of Ono and Hirata

Using the same proof as in [3], we can obtain the following results.

Lemma 1 [3] The coloring after an invocation of Recolor-Component(H, α, β, v) for a connected graph $H = (V_H, E_H)$ satisfies the following conditions:

a. If all vertices in H are even-degree and $|E_H|$ is even, then $d(s, \alpha) = d(s, \beta)$ for any vertex s.

b. If all vertices in H are even-degree and $|E_H|$ is odd, then $d(u, \alpha) = d(u, \beta) + 2$ for the start vertex u and $d(s, \alpha) = d(s, \beta)$ for any vertex s other then u.

c. If there are odd-degree vertices in H, then $|d(s, \alpha) - d(s, \beta)| = 1$ for any odd-degree vertex s and $d(s, \alpha) = d(s, \beta)$ for any even-degree vertex s.

Lemma 2 [3] The coloring after an invocation of Recolor-Component(H, α, β, v) satisfies the strict balance condition, that is,

$$e_H(\beta) \leq e_H(\alpha) \leq e_H(\beta) + 1.$$

Corollary 1 [3] Lemma 2 holds for Recolor $(G_{\alpha\beta}, \alpha, \beta, v)$ instead of Recolor-Component(H, α, β, v).

Corollary 2 [3] At any time of the algorithm, the coloring satisfies "balanced constraint:"
$|e_H(\alpha) - e_H(\beta)| \leq 1$ for any colors α and β.

Lemma 3 [3] The running time of Recolor $(G_{\alpha\beta}, \alpha, \beta, v)$ is $O(e(\alpha) + e(\beta))$.

Lemma 4 [3] $e(\alpha) = O(n/k)$ for any color α at any time of the algorithm.

5.2 Our Results

Ono and Hirata [3] defined the excess $\Phi(s)$ for an vertex $s \in V$ as follows:

$$\Phi(s) = \sum_{C_i \in C : d(s, C_i) > m} (d(s, C_i) - m),$$

where m is the number of the connections between an FPGA and a crossbar. They proved that the excess Φ, the summation of $\Phi(s)$ over all vertices s, must decrease by constant when running their algorithm.

It seems that we can use the same proof to obtain the same results if we could find a suitable m such as they did. It is possible to find some m which makes Φ decrease, but it is hard to insure a constant decrease. Thus we make the following definations, which also leads to a much more complex proof in Sect. 5.3.

Omitting the subscript G, let $\bar{d}(s) = \lfloor d(s)/k \rfloor$ for a vertex $s \in V$, we define the color sets as follows:

$$C_1(s) = \{C_i \in C | d(s, C_i) \leq \bar{d}(s) - 1\},$$
$$C_2(s) = \{C_i \in C | d(s, C_i) = \bar{d}(s)\},$$
$$C_3(s) = \{C_i \in C | d(s, C_i) \geq \bar{d}(s) + 1\}.$$

Define the cost of s as

$$\Phi(s) = \sum_{C_i \in C : d(s,C_i) \geq \bar{d}(s)+1} \{d(s, C_i) - (\bar{d}(s) + 1)\} + \sum_{C_i \in C : d(s,C_i) \leq \bar{d}(s)-1} \{(\bar{d}(s) - 1) - d(s, C_i)\}.$$

The cost Φ of the coloring is the summation of $\Phi(s)$ over all vertices s, and it is bounded as follows:

$$\begin{aligned}\Phi &= \sum_{s \in V} \Phi(s) \\ &< \sum_{s \in V} \{ \sum_{C_i \in C : d(s,C_i) \geq \bar{d}(s)+1} d(s, C_i) \\ &\quad + \sum_{C_i \in C : d(s,C_i) \leq \bar{d}(s)-1} \bar{d}(s)\} \\ &< \sum_{s \in V} 2d(s) = 4|E| = 4n.\end{aligned}$$

Let $\Delta\Phi(s)$ denote the difference of $\Phi(s)$ before and after the invocation of RECOLOR$(G_{\alpha\beta}, \alpha, \beta, v)$. In the following, $d(s, \alpha)$ and $d'(s, \alpha)$ denote the numbers of α-edges before and after the invocation of RECOLOR, respectively.

For the special vertex v selected by Algorithm (G, C),

$$d(v, \alpha) = \max\{d(v, C_i) : C_i \in C\},$$
$$d(v, \beta) = \min\{d(v, C_i) : C_i \in C\},$$

we note that $d(v, \alpha) \geq \bar{d}(v) + 1$ and $d(v, \beta) \leq \bar{d}(v)$.

Lemma 5 Assume that there exists a vertex v and colors α, β such that $d(v, \alpha) \geq \bar{d}(v) + 1$, $d(v, \beta) \leq \bar{d}(v)$ and $d(v, \alpha) - d(v, \beta) \geq 3$. Φ must decrease if we invoke RECOLOR$(G_{\alpha\beta}, \alpha, \beta, v)$.

In the following proof, $d_{\alpha\beta}(s)$ denotes the degree of a vertex s in the subgraph $G_{\alpha\beta}$, that is, $d_{\alpha\beta}(s) = d(s, \alpha) + d(s, \beta)$.

5.3 Proof of Lemma 5

Let v be the special vertex selected by Algorithm (G, C), we rewrite Lemma 1 [3] in the following way in 3 cases.

Lemma 1' [3] The coloring after an invocation of RECOLOR-COMPONENT(H, α, β, v) for a connected graph $H = (V_H, E_H)$ satisfies the following conditions:

a. For any even-degree vertex s not belonging to case c, $d'(s, \alpha) = d'(s, \beta)$.

b. For any odd-degree vertex s, $|d'(s, \alpha) - d'(s, \beta)| = 1$.

c. For the start vertex s when the connected component H has odd number of edges and no odd-degree vertices, $d'(s, \alpha) = d'(s, \beta) + 2$.

We first discuss how $\Delta\Phi(s)$ changes for any vertex $s \neq v$ for the three cases. For such vertices in case c, they are selected by RECOLOR-COMPONENT(H, x, y, v) with $|d(s, \alpha) - d(s, \beta)| \geq 2$ before the invocation. If $|d(s, \alpha) - d(s, \beta)| = 2$, it is easy to see that $\Delta\Phi(s) = 0$ for $\Phi(s)$ does not change after the invocation. So for case c, we only need to discuss on the vertices where $s \neq v$ and $|d(s, \alpha) - d(s, \beta)| \geq 4$. After that, we show how $\Delta\Phi(v)$ changes for the special vertex v. In the end, we conclude the response of Φ of the coloring over all vertices.

In the following proof, we suppose by symmetry that $d(s, \alpha) \geq d(s, \beta)$ before the invocation and $d'(s, \alpha) \geq d'(s, \beta)$ after the invocation. During the discussions for any vertex $s \neq v$, we ommit (s) for all the notations corresponding with s, such as $C_1(s), d_{\alpha\beta}(s), \bar{d}(s), \cdots$, etc.

1. if $\alpha \in C_1$ and $\beta \in C_1$

a. $d_{\alpha\beta} \leq 2\bar{d} - 2$ and $d'(s, \alpha) = d'(s, \beta) \leq \bar{d} - 1$, thus

$$\begin{aligned}\Delta\Phi &= \{(\bar{d} - 1) - d'(s, \alpha) + (\bar{d} - 1) - d'(s, \beta)\} \\ &\quad -\{(\bar{d} - 1) - d(s, \alpha) + (\bar{d} - 1) - d(s, \beta)\} \\ &= 0.\end{aligned}$$

b. $d_{\alpha\beta} \leq 2\bar{d} - 3$ for s is odd-degree and $d'(s, \beta) < d'(s, \alpha) \leq \bar{d} - 1$, thus $\Delta\Phi = 0$ same as case a.

c. $d(s, \alpha) \leq \bar{d} - 1$ and $d(s, \beta) \leq \bar{d} - 5$ for $d(s, \alpha) - d(s, \beta) \geq 4$, then $d_{\alpha\beta} \leq 2\bar{d} - 6$ and $d'(s, \beta) < d'(s, \alpha) \leq \bar{d} - 2$, thus $\Delta\Phi = 0$ same as case a.

2. if $\alpha \in C_2$ and $\beta \in C_1$

a. $d_{\alpha\beta} \leq 2\bar{d} - 2$ for s is even-degree and $d'(s, \alpha) = d'(s, \beta) \leq \bar{d} - 1$, thus

$$\begin{aligned}\Delta\Phi &= \{(\bar{d} - 1) - d'(s, \alpha) + (\bar{d} - 1) - d'(s, \beta)\} \\ &\quad -\{(\bar{d} - 1) - d(s, \beta)\} \\ &= \bar{d} - 1 - d(s, \alpha) = -1 < 0.\end{aligned}$$

b. $d_{\alpha\beta} \leq 2\bar{d} - 1$.

If $d_{\alpha\beta} = 2\bar{d} - 1$, then $d'(s, \alpha) = d(s, \alpha) = \bar{d}$ and $d'(s, \beta) = d(s, \beta) = \bar{d} - 1$, thus $\Delta\Phi = 0$ for Φ does not change after the invocation.

If $d_{\alpha\beta} \leq 2\bar{d} - 3$, then $d'(s, \beta) < d'(s, \alpha) \leq \bar{d} - 1$, thus we obtain that $\Delta\Phi = \bar{d} - 1 - d(s, \alpha) = -1 < 0$ same as case a.

c. $d_{\alpha\beta} \leq 2\bar{d} - 4$ for $d(s, \alpha) - d(s, \beta) \geq 4$, thus $d'(s, \beta) < d'(s, \alpha) \leq \bar{d} - 1$ and $\Delta\Phi = \bar{d} - 1 - d(s, \alpha) = -1 < 0$ same as case a.

3. if $\alpha \in C_3$ and $\beta \in C_1$

a. If $d_{\alpha\beta} \leq 2\bar{d} - 2$, then $d'(s, \alpha) = d'(s, \beta) \leq \bar{d} - 1$, thus

$$\begin{aligned}\Delta\Phi &= \{(\bar{d} - 1) - d'(s, \alpha) + (\bar{d} - 1) - d'(s, \beta)\} \\ &\quad -\{d(s, \alpha) - (\bar{d} + 1) + (\bar{d} - 1) - d(s, \beta)\}\end{aligned}$$

$$= 2(\bar{d} - d(s,\alpha)) < 0.$$

If $d_{\alpha\beta} = 2\bar{d}$, then $d'(s,\alpha) = d'(s,\beta) = \bar{d}$, thus

$$\Delta\Phi = 0 - \{d(s,\alpha) - (\bar{d}+1) + (\bar{d}-1) - d(s,\beta)\}$$
$$= d(s,\beta) + 2 - d(s,\alpha) \leq 0.$$

If $d_{\alpha\beta} \geq 2\bar{d}+2$, then $d'(s,\alpha) = d'(s,\beta) \geq \bar{d}+1$, thus

$$\Delta\Phi = \{d'(s,\alpha) - (\bar{d}+1) + d'(s,\beta) - (\bar{d}+1)\}$$
$$-\{d(s,\alpha) - (\bar{d}+1) + (\bar{d}-1) - d(s,\beta)\}$$
$$= 2(d(s,\beta) - \bar{d}) < 0.$$

b. If $d_{\alpha\beta} \leq 2\bar{d}-3$, then $d'(s,\beta) < d'(s,\alpha) \leq \bar{d}-1$, thus $\Delta\Phi = 2(\bar{d} - d(s,\alpha)) < 0$.

If $2\bar{d}-1 \leq d_{\alpha\beta} \leq 2\bar{d}+1$, then $\bar{d} \leq d'(s,\alpha) \leq \bar{d}+1$ and $\bar{d}-1 \leq d'(s,\beta) \leq \bar{d}$, thus $\Delta\Phi = d(s,\beta) + 2 - d(s,\alpha) \leq 0$.

If $d_{\alpha\beta} \geq 2\bar{d}+3$, then $d'(s,\alpha) > d'(s,\beta) \geq \bar{d}+1$, thus $\Delta\Phi = 2(d(s,\beta) - \bar{d}) < 0$.

c. If $d_{\alpha\beta} \leq 2\bar{d}-4$, then $d'(s,\beta) < d'(s,\alpha) \leq \bar{d}-1$ and $\Delta\Phi = 2(\bar{d} - d(s,\alpha)) < 0$.

If $d_{\alpha\beta} = 2\bar{d}-2$, then $d'(s,\alpha) = \bar{d}$ and $d'(s,\beta) = \bar{d}-2$, thus

$$\Delta\Phi = \{(\bar{d}-1) - d'(s,\beta)\}$$
$$-\{d(s,\alpha) - (\bar{d}+1) + (\bar{d}-1) - d(s,\beta)\}$$
$$= d(s,\beta) + 3 - d(s,\alpha).$$

If $d_{\alpha\beta} = 2\bar{d}$, then $d'(s,\alpha) = \bar{d}+1$ and $d'(s,\beta) = \bar{d}-1$, thus $\Delta\Phi = d(s,\beta) + 2 - d(s,\alpha)$.

If $d_{\alpha\beta} = 2\bar{d}+2$, then $d'(s,\alpha) = \bar{d}+2$ and $d'(s,\beta) = \bar{d}$, thus we also get that $\Delta\Phi = d(s,\beta) + 3 - d(s,\alpha)$.

For $d(s,\alpha) - d(s,\beta) \geq 4$, $\Delta\Phi < 0$ for the forward three cases of $d_{\alpha\beta} = 2\bar{d} \mp 2$ and $d_{\alpha\beta} = 2\bar{d}$.

If $d_{\alpha\beta} \geq 2\bar{d}+4$, then $d'(s,\alpha) > d'(s,\beta) \geq \bar{d}+1$, thus $\Delta\Phi = 2(d(s,\beta) - \bar{d}) < 0$.

4. if $\alpha \in C_2$ and $\beta \in C_2$

a. $d'(s,\alpha) = d'(s,\beta) = \bar{d}$, thus

$$\Delta\Phi = 0 - 0 = 0.$$

For $d_{\alpha\beta} = 2\bar{d}$, no such case for b and c.

5. if $\alpha \in C_3$ and $\beta \in C_2$

a. $d_{\alpha\beta} \geq 2\bar{d}+2$ for s is even-degree and $d'(s,\alpha) = d'(s,\beta) \geq \bar{d}+1$, thus

$$\Delta\Phi = \{d'(s,\alpha) - (\bar{d}+1) + d'(s,\beta) - (\bar{d}+1)\}$$
$$-\{d(s,\alpha) - (\bar{d}+1)\}$$
$$= d(s,\beta) - (\bar{d}+1) = -1 < 0.$$

b. $d_{\alpha\beta} \geq 2\bar{d}+1$.

If $d_{\alpha\beta} = 2\bar{d}+1$, then $d'(s,\alpha) = d(s,\alpha) = \bar{d}+1$ and $d'(s,\beta) = d(s,\beta) = \bar{d}$, thus $\Delta\Phi = 0$ for Φ does not change after the invocation.

If $d_{\alpha\beta} \geq 2\bar{d}+3$, then $d'(s,\alpha) > d'(s,\beta) \geq \bar{d}+1$, thus $\Delta\Phi = d(s,\beta) - (\bar{d}+1) = -1 < 0$.

c. $d_{\alpha\beta} \geq 2\bar{d}+4$ for $d(s,\alpha) - d(s,\beta) \geq 4$, thus $d'(s,\alpha) > d'(s,\beta) \geq \bar{d}+1$ and $\Delta\Phi = d(s,\beta) - (\bar{d}+1) = -1 < 0$.

6. if $\alpha \in C_3$ and $\beta \in C_3$

a. $d_{\alpha\beta} \geq 2\bar{d}+2$ and $d'(s,\alpha) = d'(s,\beta) \geq \bar{d}+1$, thus

$$\Delta\Phi = \{d'(s,\alpha) - (\bar{d}+1) + d'(s,\beta) - (\bar{d}+1)\}$$
$$-\{d(s,\alpha) - (\bar{d}+1) + d(s,\beta) - (\bar{d}+1)\}$$
$$= 0.$$

b. $d_{\alpha\beta} \geq 2\bar{d}+3$ for s is odd-degree and $d'(s,\alpha) > d'(s,\beta) \geq \bar{d}+1$, thus $\Delta\Phi = 0$ same as case a.

c. $d(s,\beta) \geq \bar{d}+1$ and $d(s,\alpha) \geq \bar{d}+5$ for $d(s,\alpha) - d(s,\beta) \geq 4$, then $d_{\alpha\beta} \geq 2\bar{d}+6$ and $d'(s,\alpha) > d'(s,\beta) \geq \bar{d}+2$, thus $\Delta\Phi = 0$ same as case a.

We proved that for any vertex $s \neq v$, $\Delta\Phi(s)$ never increase. For the special vertex v, it is possible for cases 3 and 5 for $d(v,\alpha) \geq \bar{d}(v)+1$ and $d(v,\beta) \leq \bar{d}(v)$.

3. if $\alpha \in C_3(v)$ and $\beta \in C_1(v)$

b. $\Delta\Phi(v) < 0$ for $d(v,\alpha) - d(v,\beta) \geq 3$.

a and c. $d(v,\alpha) - d(v,\beta) \geq 4$ for $d_{\alpha\beta}(v)$ is even, thus $\Delta\Phi(v) < 0$.

5. if $\alpha \in C_3(v)$ and $\beta \in C_2(v)$

b. $d(v,\beta) = \bar{d}(v)$ and $d(v,\alpha) - d(v,\beta) \geq 3$, then $d_{\alpha\beta}(v) \geq 2\bar{d}(v)+3$. Thus $\Delta\Phi(v) < 0$ same as any vertex $s \neq v$ we have shown.

a and c. $\Delta\Phi(v) < 0$ same as any vertex $s \neq v$ we have shown.

5.4 Result for Running Time

We have proved Lemma 5 in all cases, that is, in the course of the algorithm, the value of Φ must decrease by at least 1 when we invoke Recolor. Thus after at most $4n$ invocations of Recolor, Φ must be 0. Now we obtain the main theorem.

Theorem 1 Algorithm (G,C) solves the Nearly Equitable Edge-coloring Problem in $O(n^2/k)$ time for any multigraph G, where n and k are the numbers of the edges and the colors, respectively.

Proof: From lemmas 3 and 4, Recolor$(G_{\alpha\beta},\alpha,\beta,v)$ takes $O(n/k)$ time, thus we conclude that the running time of our algorithm is $O(n \times n/k) = O(n^2/k)$. □

6. Concluding Remarks

We use the same technique of Ono and Hirata [3] and present a new algorithm that nearly equitably colors any multigraph G with n edges using k colors. It runs in $O(n^2/k)$ time, which slightly improves the result of $O(n^2/k + n|V|)$ time [2].

Acknowledgment

The first author thanks the supportment by the COE program, Intelligent Media Integration in Nagoya University.

References

[1] A.J.W. Hilton and D. de Werra, "Sufficient conditions for balanced and for equitable edge-coloring of graphs," O.R. Working paper 82/3, Dépt. of Math., École Polytechnique Fédérate de Lausanne, Switzerland, 1982.
[2] S. Nakano, Y. Suzuki, and T. Nishizeki, "An algorithm for the nearly equitable edge-coloring of graphs," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J78-D-I, no.5, pp.437–444, May 1995.
[3] T. Ono and T. Hirata, "An improved algorithm for the net assignment problem," IEICE Trans. Fundamentals, vol.E84-A, no.5, pp.1161–1165, May 2001.
[4] D.S. Hochbaum, T. Nishizeki, and D.B. Shmoys, "A better than 'best possible' algorithm to edge color multigraphs," J. Algorithms, vol.7, no.1, pp.79–104, 1986.
[5] T. Nishizeki and K. Kashiwagi, "On the 1.1 edge-coloring of multigraphs," SIAM J. Disc. Math., vol.3, no.3, pp.391–410, 1990.

Xuzhen Xie received B.E. from East China University of Science and Technology in 1995 and M.E. from Nagoya University in 2003. She is currently a doctor student in the Graduate School of Information Science, Nagoya University. Her research interests include approximation algorithms.

Takao Ono received B.E., M.E. and Ph.D. from Nagoya University in 1993, 1995 and 1999, respectively. He is currently a Research Associate in the Graduate School of Information Science, Nagoya University. His research interests include approximation algorithms.

Shin-ichi Nakano received B.E., M.E. and Ph.D. from Tohoku University in 1985, 1987 and 1992, respectively. He is currently an Associate Professor in the Department of Computer Science, Faculty of Engineering, Gunma University. His research interests include graph algorithms and approximation algorithms.

Tomio Hirata received B.S., M.S. and Ph.D. in Computer Science, all from Tohoku University in 1976, 1978, and 1981, respectively. He is currently a Professor in the Graduate School of Information Science, Nagoya University. His research interests include graph algorithms and approximation algorithms.