多重グラフの均等辺彩色問題に対するアルゴリズム

謝　旭珍、小野　孝男、平田　富夫
〒 464-8603 名古屋市千種区不老町
名古屋大学大学院情報科学研究科
電話: +81-52-789-3440
FAX: +81-52-789-3089
Email:
sharryx@hirata.nuee.nagoya-u.ac.jp, takao@hirata.nuee.nagoya-u.ac.jp, hirata@nuee.nagoya-u.ac.jp
中野　眞一
〒 376-8515 群馬県桐生市天神町
群馬大学工学部情報工学科
電話: +81-277-30-1812
Email: nakano@msc.cs.gunma-u.ac.jp

概要

与えられたグラフの各点の隣接辺をできるだけ均等に彩色する問題はグラフの均等辺彩色問題と呼ばれる。この問題は、$O(kn^2)$ 時間で解けることが証明されている、ここで、n は辺の数で、k は与えられた色の数である。その後、実行時間は $O(n^2/k+n|V|)$ に改良された。本論文では、新しいアルゴリズムを提案し、このアルゴリズムは多重グラフの均等辺彩色問題を $O(n^2/k)$ 時間で解くことを示す。

キーワード

均等彩色、歩道

An Improved Algorithm for the Nearly Equitable Edge-coloring Problem

XuZhen Xie, Takao Ono, Tomio Hirata
Graduate School of Information Science
Nagoya University
Furo-cho, Chikusa-ku
Nagoya 464–8603, Japan
Telephone: +81-52-789-3440
Fax: +81-52-789-3089
Email:
sharryx@hirata.nuee.nagoya-u.ac.jp, takao@hirata.nuee.nagoya-u.ac.jp, hirata@nuee.nagoya-u.ac.jp
Shin-ichi Nakano
Department of Computer Science
Faculty of Engineering, Gunma University
Tenjin-cho, Kiryu-Shi
Gunma 376–8515, Japan
Telephone: +81-277-30-1812
Email: nakano@msc.cs.gunma-u.ac.jp

Abstract

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.

Keywords

nearly equitable edge coloring, Euler circuit

I. 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 Section II, we give some definitions. In Section III, we introduce the results of Hilton and Werra. Sections IV and V are for the new algorithm and the results we obtained. Finally, concluding remarks are in Section VI.

II. 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 $\mathcal{C} = \{C_1, C_2, ..., C_k\}$, 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 $C_i, C_j \in \mathcal{C}$, $|d(v, C_i) - d(v, C_j)| \leq 2$ [1].

III. $O(kn^2)$-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 $C_i, C_j \in \mathcal{C}$ 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 $C_i, C_j \in \mathcal{C}$, $|d(v, C_i) - d(v, C_j)| \leq 2$ when the algorithm stops. The running time of the algorithm is proved to be $O(kn^2)$.

Define

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

then

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

$Cost$ 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][2][3][4][5]. We also use it for our new algorithm in the following.

IV. A New Algorithm for Nearly Equitable Edge-coloring

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 $(G, \mathcal{C})$

Input: a multigraph $G = (V, E)$ with $|E| = n$ and a color set $\mathcal{C}$ with $|\mathcal{C}| = k$

Output: a nearly equitable edge-coloring for G

Assign $C_1, C_2, ..., C_k$ to n edges repeatedly, so that $\lceil n/k \rceil$ or $\lfloor n/k \rfloor$ edges have the same color.

while there exists $v \in V$ and different colors $C_i, C_j \in \mathcal{C}$ such that $|d(v, C_i) - d(v, C_j)| \geq 3$ **do**

  for the vertex v, find $\alpha, \beta \in \mathcal{C}$ with

$$\begin{aligned} d(v, \alpha) &= \max\{d(v, C_i) : C_i \in \mathcal{C}\}, \\ d(v, \beta) &= \min\{d(v, C_i) : C_i \in \mathcal{C}\}. \end{aligned}$$

  Recolor$(G_{\alpha\beta}, \alpha, \beta, v)$.

Recolor$(G_{\alpha\beta}, \alpha, \beta, v)$.

Input: a multigraph $G_{\alpha\beta}$ with all edges colored with colors α and β and a selected vertex v

Output: a nearly equitable edge-coloring for $G_{\alpha\beta}$

Let $x \leftarrow \alpha$ and $y \leftarrow \beta$.

for each connected component H in G **do**

  Recolor-Component(H, x, y, v).

  if H has odd number of edges **then**

    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

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 β and α (β 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 $r \in H$ with $|d(r, x) - d(r, y)| \geq 2$ **then**

      Let r be the start vertex u.

  Traverse an Euler circuit starting at the vertex u and assign colors α and β (α comes first) to the edges alternately along the way.

V. Analysis of the Algorithm

A. 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.

B. *Our Results*

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

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

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 obtain the same results if we could find a suitable m such as they did. To do that, we first prove the following lemma.

Lemma 5 Suppose that four numbers x, y, x', y' satisfy that $x + y = x' + y'$ and $|x - y| \geq |x' - y'|$, then for any real a, the inequality $|x - a| + |y - a| \geq |x' - a| + |y' - a|$ holds.

Proof. For the four numbers x, y, x', y' with $x + y = x' + y'$ and $|x - y| \geq |x' - y'|$, we note that $x \leq x', y' \leq y$ or $y \leq x', y' \leq x$. Suppose by symmetry that $x \geq y$ and $x' \geq y'$, then $y \leq y' \leq x' \leq x$ and $x - y \geq x' - y'$.

Let $f_a = |x - a| + |y - a|$, $f'_a = |x' - a| + |y' - a|$ and $\Delta_a = f'_a - f_a$, now we prove that Δ_a never increase for any real a by dividing the range of a into five subranges.

1. $a \leq y$
$f_a = x - a + y - a = (x + y) - 2a$ and $f'_a = x' - a + y' - a = (x' + y') - 2a$, thus $\Delta_a = 0$.

2. $y < a < y'$
$f_a = x - a + a - y = x - y$ and $f'_a = (x' + y') - 2a$, thus $\Delta_a = 2(y - a) < 0$.

3. $y' \leq a \leq x'$
$f_a = x - y$ and $f'_a = x' - a + a - y' = x' - y'$, thus $\Delta_a = (x' - y') - (x - y) \leq 0$.

4. $x' < a < x$
$f_a = x - y$ and $f'_a = a - x' + a - y' = 2a - (x' + y')$, thus $\Delta_a = 2(a - x) < 0$.

5. $x \leq a$
$f_a = a - x + a - y = 2a - (x + y)$ and $f'_a = 2a - (x' + y')$, thus $\Delta_a = 0$. □

For a vertex $s \in V$, let $\bar{d}(s) = \lfloor d(s)/k \rfloor$, and we define the cost as the following:

$$\Phi(s) = \sum_{C_i \in \mathcal{C}} \{|d(s, C_i) - (\bar{d}(s) + 1/2)| - 1/2\},$$

it is obvious that $\Phi(s)$ is an integer.

$\Delta\Phi(s)$ denotes the difference of $\Phi(s)$ before and after the invocation of $\textsc{Recolor}(G_{\alpha\beta}, \alpha, \beta, v)$. $d(s, \alpha)$ and $d'(s, \alpha)$ denote the numbers of α-edges before and after the invocation of Recolor, respectively. Let v denote the special vertex selected by Algorithm $(G, \mathcal{C})$, we are going to consider how $\Delta\Phi(s)$ changes and to prove the following lemmas.

Lemma 6 For any $s \in V$ and $s \neq v$, $\Phi(s)$ does not increase by the call of Recolor.

Proof. For any $s \in V$ and $s \neq v$, it satisfies $d(s, \alpha) + d(s, \beta) = d'(s, \alpha) + d'(s, \beta)$.
If s is a start vertex when the connected component H has odd number of edges and no odd-degree vertices, it satisfies $|d'(s, \alpha) - d'(s, \beta)| = 2 \leq |d(s, \alpha) - d(s, \beta)|$.
Otherwise, $|d'(s, \alpha) - d'(s, \beta)| = 0 \leq |d(s, \alpha) - d(s, \beta)|$ holds for any even-degree vertex and $|d'(s, \alpha) - d'(s, \beta)| \leq 1 \leq |d(s, \alpha) - d(s, \beta)|$ holds for any odd-degree vertex.
Thus

$$\begin{aligned}
\Delta\Phi(s) &= \sum_{C_i \in \mathcal{C}} \{|d'(s, C_i) - (\bar{d}(s) + 1/2)| - 1/2\} \\
&\quad - \sum_{C_i \in \mathcal{C}} \{|d(s, C_i) - (\bar{d}(s) + 1/2)| - 1/2\} \\
&= \{|d'(s, \alpha) - (\bar{d}(s) + 1/2)| + |d'(s, \beta) - (\bar{d}(s) + 1/2)|\} \\
&\quad - \{|d'(s, \alpha) - (\bar{d}(s) + 1/2)| + |d'(s, \beta) - (\bar{d}(s) + 1/2)|\} \\
&\leq 0
\end{aligned}$$

by Lemma 5. □

Lemma 7 For the special vertex v, $\Phi(v)$ must decrease by the call of Recolor.

Proof. For the special vertex v selected by Algorithm $(G, \mathcal{C})$ with $|d(v, C_i) - d(v, C_j)| \geq 3$,

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

we note that $d(v, \alpha) \geq \bar{d}(v) + 1$ and $d(v, \beta) \leq \bar{d}(v)$. Thus $d(v, \beta) < \bar{d}(v) + 1/2 < d(v, \alpha)$, and we only need to consider sugranges 2, 3 and 4 same as the proof for Lemma 5. For subrange 3, it satisfies $|d'(v, \alpha) - d'(v, \beta)| \leq 2 < |d(v, \alpha) - d(v, \beta)|$, so $\Delta\Phi(v) < 0$. □

Lemma 8 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 $d(v, \alpha) - d(v, \beta) \geq 3$. Let $\Phi = \sum_{s \in V} \Phi(s)$ be the cost of the coloring, then Φ must decrease by at least 1 if we invoke $\textsc{Recolor}(G_{\alpha\beta}, \alpha, \beta, v)$.

Proof. For any vertex $s \neq v$, $\Phi(s)$ does not increase by Lemma 6. For the special vertex v, $\Phi(v)$ must decrease by Lemma 7 and the amount of decrease must be at least 1 because $\Phi(v)$ is an integer. □

C. Result for Running Time

The cost Φ of the coloring is bounded as follows:

$$\Phi = \sum_{s \in V} \Phi(s)$$

$$
\begin{aligned}
&= \sum_{s\in V}\sum_{C_i\in\mathcal{C}}\{|d(s,C_i)-(\bar{d}(s)+1/2)|-1/2\} \\
&\le \sum_{s\in V}\sum_{C_i\in\mathcal{C}}\{d(s,C_i)+\bar{d}(s)+1/2-1/2\} \\
&= \sum_{s\in V}\sum_{C_i\in\mathcal{C}} d(s,C_i)+\sum_{s\in V}\sum_{C_i\in\mathcal{C}}\bar{d}(s) \\
&\le 2d(s)=4|E|=4n.
\end{aligned}
$$

We have proved Lemma 8, 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 $O(n \times n/k) = O(n^2/k)$. □

VI. 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*, 1982. Dépt. of Math., École Polytechnique Fédérate de Lausanne, Switzerland.
[2] S. Nakano, Y. Suzuki, and T. Nishizeki. An algorithm for the nearly equitable edge-coloring of graphs. *IEICE Trans.*, J78-D-I(5):437–444, 1995.
[3] T. Ono and T. Hirata. An improved algorithm for the net assignment problem. *IEICE Trans.*, E84-A(5):1161–1165, 2001.
[4] D. S. Hochbaum, T. Nishizeki, and D. B. Shmoys. A better than 'best possible' algorithm to edge color multigraphs. *Journal of Algorithms*, 7(1):79–104, 1986.
[5] T. Nishizeki and K. Kashiwagi. On the 1.1 edge-coloring of multigraphs. *SIAM J. Disc. Math.*, 3(3):391–410, 1990.