Densities, Matchings, and Fractional Edge-Colorings

Xujin Chen[a*] Wenan Zang[b†] Qiulan Zhao[c‡]

[a] Institute of Applied Mathematics, Chinese Academy of Sciences
Beijing 100190, China

[b] Department of Mathematics, The University of Hong Kong
Hong Kong, China

[c] Department of Mathematics, Nanjing University
Nanjing 210093, China

Abstract

Given a multigraph $G = (V, E)$ with a positive rational weight $w(e)$ on each edge e, the *weighted density problem* (WDP) is to find a subset U of V, with $|U| \geq 3$ and odd, that maximizes $\frac{2w(U)}{|U|-1}$, where $w(U)$ is the total weight of all edges with both ends in U, and the *weighted fractional edge-coloring problem* can be formulated as the linear program

$$\begin{array}{ll} \text{Minimize} & \mathbf{1}^T \boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} = \boldsymbol{w} \\ & \quad \boldsymbol{x} \geq \mathbf{0}, \end{array}$$

where A is the edge-matching incidence matrix of G. These two problems are closely related to the celebrated Goldberg-Seymour conjecture on edge-colorings of multigraphs, and are interesting in their own right. Even when $w(e) = 1$ for all edges e, determining whether WDP can be solved in polynomial time was posed by Jensen and Toft [9] and by Stiebitz *et al.* [22] as an open problem. In this paper we present strongly polynomial-time algorithms for solving them exactly, and develop a novel matching removal technique for multigraph edge-coloring.

MSC 2000 subject classification. Primary: 90C27, 68Q25.
OR/MS subject classification. Primary: Programming/graphs.
Key words. Multigraph, density, matching, fractional edge-coloring, algorithm.

*Supported in part by NSF of China under Grant 11531014.
†Supported in part by the Research Grants Council of Hong Kong.
‡Corresponding author. E-mail: qiulanzhao@163.com.

1 Introduction

Multigraphs considered in this paper may have parallel edges but contain no loops. Given a multigraph $G = (V, E)$, the *edge-coloring problem* (ECP) is to color the edges of G with the minimum number of colors so that no two adjacent edges have the same color, where two edges are called *adjacent* if they are incident with a common vertex. The optimal value of ECP, denoted by $\chi'(G)$, is called the *chromatic index* of G. Holyer [12] proved that ECP is NP-hard, even when restricted to a simple cubic graph, so there is no efficient algorithm for solving it exactly unless $NP = P$. Let $\Delta(G)$ be the maximum degree of G, and let the *density* of G be defined as

$$\Gamma(G) = \max\Big\{\frac{2|E(U)|}{|U|-1} : \ U \subseteq V, \ |U| \geq 3 \ \text{ and } \ \text{odd}\Big\},$$

where $E(U)$ is the set of all edges of G with both ends in U. Clearly, $\chi'(G) \geq \max\{\Delta(G), \, \Gamma(G)\}$; this lower bound, as shown by Seymour [21] using Edmonds' matching polytope theorem [3], is precisely the *fractional chromatic index* of G, which is the optimal value of the *fractional edge-coloring problem* (FECP):

$$\begin{array}{ll} \text{Minimize} & \mathbf{1}^T \boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} = \mathbf{1} \\ & \ \ \boldsymbol{x} \geq \mathbf{0}, \end{array}$$

where A is the edge-matching incidence matrix of G. In the 1970s, Goldberg [6] and Seymour [21] independently made the following celebrated conjecture.

Conjecture 1.1. *Every multigraph G satisfies $\chi'(G) \leq \max\{\Delta(G) + 1, \, \lceil \Gamma(G) \rceil\}$.*

Its validity would imply that, first, for any multigraph its chromatic index differs from its fractional chromatic index by at most one, so FECP enjoys a fascinating rounding property; second, ECP can be approximated within one of the optimum, and hence is one of the "easiest" NP-hard problems; third, an analogue to Vizing's theorem [24] on edge-coloring simple graphs, a fundamental result in graph theory, holds for multigraphs.

Over the past four decades, Conjecture 1.1 has been a subject of extensive research in the fields of operations research, computer science, and graph theory, and has inspired a significant body of work, with contributions from many researchers; see Stiebitz *et al.* [22] for a comprehensive account. Given its intimate connection with Conjecture 1.1, we study FECP in this paper. For convenience, we shall actually investigate its weighted version (WFECP):

$$\begin{array}{ll} \text{Minimize} & \mathbf{1}^T \boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} = \boldsymbol{w} \\ & \ \ \boldsymbol{x} \geq \mathbf{0}, \end{array}$$

where $\boldsymbol{w} = (w(e) : e \in E)$ and $w(e)$ is a positive rational weight (not necessarily integral) associated with each edge e of G. Let $\chi^*_w(G)$ denote the optimal value of WFECP. For each $F \subseteq E$, let $w(F) = \sum_{e \in F} w(e)$. For each $U \subseteq V$, let $w(U) = w(E(U))$. For each $v \in V$, let $\partial(v) \subseteq E$ denote the star centered at v and let $d_{G,w}(v) = w(\partial(v))$. Set $\Delta_w(G) = \max_{v \in V} d_{G,w}(v)$ and

$$\Gamma_w(G) = \max\Big\{\frac{2w(U)}{|U|-1} : \ U \subseteq V, \ |U| \geq 3 \ \text{ and } \ \text{odd}\Big\}.$$

We call $d_{G,w}(v)$, $\Delta_w(G)$, and $\Gamma_w(G)$ the *weighted degree* of v, the *maximum weighted degree* of G, and the *weighted density* of G with respect to $\boldsymbol{w}$, respectively. Throughout this paper, we set $\Omega(G) = \{v \in V : d_{G,w}(v) = \Delta_w(G)\}$, and use $n(G)$, $m(G)$, and $\ell(G)$ to denote the number of vertices, the number of edges, and the number of adjacent vertex pairs in G, respectively. Observe that the equality $m(G) = \ell(G)$ holds only when G is a simple graph. For each $U \subseteq V$, let $\overline{U} = V \backslash U$. For any disjoint vertex subsets T and U of V, let $[T, U]$ be the set of all edges between T and U in G, and let $w[T, U] = \sum_{e \in [T,U]} w(e)$. As usual, let $\mathbb{Q}$ be the set of rationals and let $\mathbb{Q}_+$ be the set of nonnegative rationals. For any set Λ of numbers and any finite set K, we use Λ^K to denote the set of vectors $\boldsymbol{x} = (x(k) : \ k \in K)$ whose coordinates are members of Λ.

It is routine to check that the aforementioned Seymour's theorem [21] holds in the weighted case as well.

Theorem 1.1 (Seymour [21])**.** *Every multigraph G satisfies $\chi_w^*(G) = \max\{\Delta_w(G), \Gamma_w(G)\}$ for all $\boldsymbol{w} \in \mathbb{Q}_+^{E(G)}$.*

Nemhauser and Park [17] observed that FECP can be solved in polynomial time by an ellipsoid algorithm, because the separation problem of its LP dual is exactly the maximum-weight matching problem (see also Schrijver [20], Theorem 28.6 on page 477). In his thesis [14], Kennedy briefly sketched a combinatorial polynomial-time algorithm for FECP. However, it does not seem to work; see the appendix for details.

One objective of this paper is to design a strongly polynomial-time algorithm for WFECP.

Theorem 1.2. *Let G be a multigraph with a positive rational weight $w(e)$ on each edge e. Then the WFECP on G can be solved in time $O(mn + n^5\ell^2 \log(n^2/\ell))$, where $n = n(G)$, $m = m(G)$, and $\ell = \ell(G)$.*

Let us introduce some terminology and notions before proceeding. A subset U of V is called an *odd set* of G if $|U| \geq 3$ and is odd. An odd set U is called *optimal* if $\frac{2w(U)}{|U|-1} = \Gamma_w(G)$. For simplicity, we abbreviate optimal odd set as *OoS*. We reserve the symbol $\mathcal{O}(G)$ for the family of all OoS's of G throughout, and refer to the problem of finding an OoS of G as the *weighted density problem* (WDP). From Theorem 1.1 it can be seen that this problem plays a crucial role in the resolution of WFECP and even Conjecture 1.1. Clearly, it is interesting in its own right. We point out that when $\Gamma(G) \geq \Delta(G)$, the value of $\Gamma(G)$ can be determined in polynomial time by combining the Padberg-Rao separation algorithm for b-matching polyhedra [18] (see also [16, 19]) with binary search. As remarked by Jensen and Toft [9] and by Stiebitz *et al.* [22], it is not clear whether $\Gamma(G)$ can be found in polynomial time in any case. In this paper we demonstrate that actually WDP, a more general problem, admits a strongly polynomial-time algorithm.

Theorem 1.3. *Let G be a multigraph with a positive rational weight $w(e)$ on each edge e. Then an optimal odd set of G can be found in time $O(m + n^4\ell \log(n^2/\ell))$, where $n = n(G)$, $m = m(G)$, and $\ell = \ell(G)$.*

Recall that WFECP consists in finding matchings $M_1, M_2, \ldots, M_t$ of G and nonnegative numbers $x(M_1), x(M_2), \ldots, x(M_t)$, such that $\sum_{e \in M_i} x(M_i) = w(e)$ for each edge e and that

$\sum_{i=1}^{t} x(M_i)$ is as small as possible. To solve it, we shall focus our attention on some special types of matchings.

Consider a matching M of G. We call M *near-perfect* if it covers all but one vertex of G (so $n(G)$ is odd). We say that M *saturates* an odd set U of G if $|E(U) \cap M| = \frac{|U|-1}{2}$; that is, M restricts to a near-perfect matching on $G[U]$, the subgraph of G induced by U. Let S be a subset of V and let $\mathcal{T}$ be a family of odd sets of G. We also say that

- M is an S*-matching* if it covers all vertices in S;
- M is a $\mathcal{T}$*-matching* if it saturates all odd sets in $\mathcal{T}$; and
- M is an $\{S, \mathcal{T}\}$*-matching* if it is both an S-matching and a $\mathcal{T}$-matching.

Caprara and Rizzi [2] proved that if $\Delta(G) \geq \Gamma(G)$, then G contains a matching that covers all vertices of maximum degree. The weighted version of this statement is given below.

Theorem 1.4 (Caprara and Rizzi [2])**.** *Let G be a multigraph with a positive rational weight $w(e)$ on each edge e. If $\Delta_w(G) \geq \Gamma_w(G)$, then we can find an $\Omega(G)$-matching of G in time $O(n^{1/2}\ell)$, where $n = n(G)$ and $\ell = \ell(G)$.*

The following theorem guarantees the existence of some other types of matchings.

Theorem 1.5. *Let G be a multigraph with a positive rational weight $w(e)$ on each edge e. Then we can find a matching M of G in time $O(m + n^5\ell\log(n^2/\ell))$, where $n = n(G)$, $m = m(G)$, and $\ell = \ell(G)$, such that*

(i) M is an $\mathcal{O}(G)$-matching if $\Delta_w(G) < \Gamma_w(G)$; and

(ii) M is an $\{\Omega(G), \mathcal{O}(G)\}$-matching if $\Delta_w(G) = \Gamma_w(G)$.

Its constructive proof is perhaps of more interest than the assertion. So far the most powerful and sophisticated technique for multigraph edge-coloring was invented by Tashkinov [23] in 2000, which generalizes the earlier methods of Kempe chains, Vizing fans [24], and Kierstead paths [15]. The crux of this technique is to capture the density $\Gamma(G)$ required to prove Conjecture 1.1, by exploring a sufficiently large tree, the so-called *Tashkinov tree*. However, this target may become unreachable when $\chi'(G)$ gets close to $\Delta(G)$, even if we allow for an unlimited number of Kempe changes; such an example can be found in Asplund and McDonald [1]. Therefore it is desirable to have some new approaches to multigraph edge-coloring. As we shall see, Theorem 1.5 together with Theorem 1.4 leads to a novel matching removal technique for this purpose, which relies heavily on density analysis, and can obviously circumvent the difficulties encountered by the method of Tashkinov trees. We believe that our proof technique can be further developed to establish the following conjecture, which is an important endeavor towards a proof of Conjecture 1.1.

Conjecture 1.2. *Let G be a multigraph with $\Delta(G) < \Gamma(G)$. Then G contains a matching M such that $\Gamma(G - M) \leq \lceil \Gamma(G) \rceil - 1$. Furthermore, there is a combinatorial polynomial-time algorithm for finding such a matching.*

The remainder of this paper is organized as follows. In Section 2, we recall some fundamental results from matching theory, and exhibit some properties enjoyed by optimal odd sets, which will be used in our search for desired matchings. In Section 3, we present a strongly polynomial-time algorithm for WDP, by using Isbell and Marlow's method [13] for fractional programming, Padberg and Rao's algorithm [18] the minimum T-cut problem, and Goemans and Ramakrishnan's algorithm [4] for and the minimum s-t T-cut problem. In Section 4, we give a combinatorial polynomial-time algorithm for finding the matching described in Theorem 1.5 based on density analysis. In Section 5, we devise a strongly polynomial-time algorithm for WFECP using a matching removal technique.

2 Preliminaries

Let us make some preparations for the algorithms to be designed in subsequent sections.

Let $G = (V, E)$ be a graph. For each $U \subseteq V$, let $o(G\backslash U)$ denote the number of odd components of $G\backslash U$; and its *deficiency*, denoted by $def(U)$, is defined to be $o(G\backslash U) - |U|$. We call U a *Tutte set* of G if $def(U) > 0$. The *deficiency* of G, denoted by $def(G)$, is defined to be $\max_{U\subseteq V} def(U)$. The following two lemmas are well known; see, for instance, Lemmas 2.1 and 2.2 in West [25].

Lemma 2.1 (Parity Lemma)**.** *Let U be a vertex subset of a graph $G = (V, E)$. Then $o(G\backslash U) - |U| \equiv |V| \pmod 2$. In particular, if U is a Tutte set and V is even, then $o(G\backslash U) \geq |U| + 2$.*

Lemma 2.2. *Let U be a maximal vertex set of a graph $G = (V, E)$ with $def(U) = def(G)$. Then all components of $G\backslash U$ are odd.*

As stated before, we shall resolve WFECP by using a matching removal method. Our search for the desired matchings is based on Lemmas 2.3-2.8 below, where we assume that $G = (V, E)$ is a multigraph with a positive rational weight $w(e)$ on each edge e and with $\Delta_w(G) \leq \Gamma_w(G)$. In their proofs, δ stands for $\Delta_w(G)$ and γ stands for $\Gamma_w(G)$.

Lemma 2.3. *Let U be an OoS of G. Then the following statements hold:*

(i) $w[U, \overline{U}] \leq \Delta_w(G)$, with equality only when $\Delta_w(G) = \Gamma_w(G)$ and $U \subseteq \Omega(G)$;

(ii) if $\Delta_w(G) = \Gamma_w(G)$, then $G[U]$ has a vertex v such that $d_{G[U],w}(v) < \Delta_w(G)$; and

(iii) $\Gamma_w(H) \leq \Gamma_w(G)$, where H arises from G by contracting U.

Proof. (i) By definition, $\frac{2w(U)}{|U|-1} = \gamma$. So $2w(U) = \gamma(|U| - 1)$. Since $2w(U) + w[U, \overline{U}] = \sum_{v\in U} d_{G,w}(v) \leq \delta|U|$, we have $w[U, \overline{U}] \leq \delta|U| - 2w(U) = \delta|U| - \gamma(|U| - 1) \leq \delta|U| - \delta(|U| - 1) = \delta$. Therefore, $w[U, \overline{U}] = \delta$ iff all inequalities in this paragraph hold with equalities iff $\delta = \gamma$ and $d_{G,w}(v) = \delta$ for all $v \in U$.

(ii) Assume on the contrary that $d_{G[U],w}(v) = \delta$ for all $v \in U$. Then $2w(U)/(|U| - 1) = \delta|U|/(|U| - 1) > \delta = \gamma$, a contradiction.

(iii) Assume on the contrary that $\frac{2w(S)}{|S|-1} > \gamma$ for some set $S \subseteq V(H)$ with $|S| \geq 3$ and odd. Then S contains the vertex v arising from contracting U. Let T be the vertex subset of G obtained from S by replacing v with U. Then $2w(T) = 2w(S) + 2w(U) > \gamma(|S| - 1) + \gamma(|U| - 1) = \gamma(|S| + |U| - 2) = \gamma(|T| - 1)$, a contradiction. ∎

A graph H is called *factor-critical* if $H\backslash v$ has a perfect matching for each vertex v. Note that every factor-critical graph has an odd number of vertices.

Lemma 2.4. *Let U be an OoS of G. Then the following statements hold:*

(i) if $\Delta_w(G) < \Gamma_w(G)$, then $G[U]$ is factor-critical;

(ii) if $\Delta_w(G) = \Gamma_w(G)$, then $G[U]\backslash v$ has a perfect matching for any $v \in U$ with $d_{G[U],w}(v) < \Delta_w(G)$.

Proof. Let v be an arbitrary vertex in U if $\Delta_w(G) < \Gamma_w(G)$ and let v be as specified in (ii) if $\Delta_w(G) = \Gamma_w(G)$. Suppose on the contrary that $G[U]\backslash v$ has no perfect matching. Since $|U\backslash\{v\}|$ is even, by Lemmas 2.1 and 2.2 (with $G[U]\backslash v$ in place of G), $G[U]\backslash v$ contains a vertex subset S such that all components T_1, T_2,...,T_k of $G[U]\backslash(S \cup \{v\})$ are odd, with $k \geq |S|+2$. Let $X = S \cup \{v\}$, let Y_i be the vertex set of T_i for each i, and let $Y = \cup_{i=1}^{k} Y_i$. Then

(a) $k \geq |X| + 1$ and $w(U) = w(X) + w[X, Y] + w(Y)$. So

(b) $2w(U) = (2w(X) + w[X, Y]) + w[X, Y] + \sum_{i=1}^{k} 2w(Y_i)$.

Observe that

(c) $(2w(X) + w[X, Y]) + w[X, Y] \leq 2\delta|X|$, with equality only when $w(X) = 0$, $X \subseteq \Omega(G)$ and $w[X, Y] = \delta|X|$.

By (b) and (c), we have $\gamma(|U|-1) = 2w(U) \leq 2\delta|X| + \sum_{i=1}^{k} \gamma(|Y_i|-1) = 2\delta|X| + \gamma|Y| - k\gamma$. As $|U| = |X| + |Y|$ and $\delta \leq \gamma$, we obtain $\gamma(|U|-1) \leq 2\gamma|X| + \gamma|Y| - k\gamma = \gamma(2|X| + |Y| - k) \leq \gamma(|X| + |Y| - 1) = \gamma(|U|-1)$, where the third inequality follows from (a). So all inequalities in this paragraph hold with equalities, and hence $k = |X| + 1$, $\delta = \gamma$, $w(X) = 0$, $X \subseteq \Omega(G)$ and $w[X, Y] = \delta|X|$ by (a) and (c), which contradicts the hypothesis of (i) or (ii). ∎

Corollary 2.5. *Let U be an OoS of G. Then $G[U]$ contains a near-perfect matching.*

Proof. The statement follows instantly from Lemma 2.4 and Lemma 2.3(ii). ∎

Lemma 2.6. *Let U_1 and U_2 be two distinct OoS's of G with $U_1 \subset U_2$, and let H be obtained from G by contracting U_1 into a single vertex v. Then the following statements hold:*

(i) $\Delta_w(H) \leq \Delta_w(G)$ and $\Gamma_w(H) = \Gamma_w(G)$;

(ii) $(U_2\backslash U_1) \cup \{v\}$ is an OoS of H.

Proof. By Lemma 2.3(i), we have $d_{H,w}(v) = w[U_1, \overline{U_1}] \leq \Delta_w(G)$. So $\Delta_w(H) \leq \Delta_w(G)$.

Write $S = (U_2\backslash U_1) \cup \{v\}$. By Lemma 2.3(iii), $\Gamma_w(H) \leq \Gamma_w(G)$. To establish the equation in (i) and the statement in (ii), it suffices to show that $\frac{2w(S)}{|S|-1} = \Gamma_w(G)$; this equation holds because $2w(S) = 2w(U_2) - 2w(U_1) = \gamma(|U_2|-1) - \gamma(|U_1|-1) = \gamma(|U_2| - |U_1|) = \gamma(|S|-1)$. It follows that $\frac{2w(S)}{|S|-1} = \Gamma_w(H) = \Gamma_w(G)$. Therefore, both (i) and (ii) hold. ∎

Lemma 2.7. *Let U_1 and U_2 be two OoS's of G with $|U_1 \cap U_2|$ odd and with $U_1\backslash U_2 \neq \emptyset \neq U_2\backslash U_1$, and let $H_1 = G[U_1 \cap U_2]$ and $H_2 = G[U_1] \cup G[U_2]$. Then the following statements hold:*

(i) G has no edge between $U_1\backslash U_2$ and $U_2\backslash U_1$;

(ii) $U_1 \cup U_2$ is an OoS, and so is $U_1 \cap U_2$ if $|U_1 \cap U_2| \geq 3$; and

(iii) if a matching of G restricts to a near-perfect matching on both H_1 and H_2, then it also restricts to a near-perfect matching on $G[U_i]$ for $i = 1, 2$.

Proof. Let $\alpha = 0$ if $|U_1 \cap U_2| = 1$ and $\alpha = \frac{2w(U_1 \cap U_2)}{|U_1 \cap U_2| - 1}$ otherwise. Then $\alpha \leq \gamma$. Observe that $w(U_1 \cup U_2) = w(U_1) + w(U_2) - w(U_1 \cap U_2) + w[U_1 \backslash U_2, U_2 \backslash U_1]$. By definition, $\frac{2w(U_1 \cup U_2)}{|U_1 \cup U_2| - 1} \leq \gamma$. So $\gamma(|U_1 \cup U_2| - 1) \geq 2w(U_1 \cup U_2) \geq 2w(U_1) + 2w(U_2) - 2w(U_1 \cap U_2) = \gamma(|U_1| - 1) + \gamma(|U_2| - 1) - \alpha(|U_1 \cap U_2| - 1) \geq \gamma(|U_1| + |U_2| - |U_1 \cap U_2| - 1) = \gamma(|U_1 \cup U_2| - 1)$. Thus all the preceding inequalities hold with equalities, and hence both (i) and (ii) hold.

It is a routine matter to check that if a matching restricts to a near-perfect matching on both H_1 and H_2, then it also restricts to a near-perfect matching on $G[U_i]$ for $i = 1, 2$. So (iii) also holds. ∎

Lemma 2.8. *Let U_1 and U_2 be two OoS's of G with $|U_1 \cap U_2| > 0$ and even, and let $T_i = U_i \backslash U_{3-i}$ for $i = 1, 2$. Then the following statements hold:*

(i) $\Delta_w(G) = \Gamma_w(G)$;

(ii) T_i is an OoS if $|T_i| \geq 3$ for $i = 1, 2$;

(iii) $U_1 \cap U_2 \subseteq \Omega(G)$ and no vertex in $U_1 \cap U_2$ is adjacent to any vertex outside $U_1 \cup U_2$ in G; and

(iv) if a matching of G covers all vertices in $U_1 \cap U_2$ and restricts to a near-perfect matching on both $G[T_1]$ and $G[T_2]$, then it also restricts to a near-perfect matching on $G[U_i]$ for $i = 1, 2$.

Proof. For $i = 1, 2$, let $\alpha_i = 0$ if $|T_i| = 1$ and $\alpha_i = \frac{2w(T_i)}{|T_i| - 1}$ otherwise. Then $\alpha_i \leq \gamma$. Observe that $w(U_1) + w(U_2) = w(T_1) + w(T_2) + 2w(U_1 \cap U_2) + w[U_1 \cap U_2, T_1 \cup T_2]$. Since $2w(U_1 \cap U_2) + w[U_1 \cap U_2, T_1 \cup T_2] \leq \delta|U_1 \cap U_2|$, we have $2w(U_1) + 2w(U_2) \leq 2w(T_1) + 2w(T_2) + 2\delta|U_1 \cap U_2|$. Hence $\gamma(|U_1| - 1) + \gamma(|U_2| - 1) \leq \alpha_1(|T_1| - 1) + \alpha_2(|T_2| - 1) + 2\delta|U_1 \cap U_2| \leq \gamma(|T_1| - 1) + \gamma(|T_2| - 1) + 2\gamma|U_1 \cap U_2| = \gamma(|U_1| - 1) + \gamma(|U_2| - 1)$; so all the preceding inequalities hold with equalities. It follows that $\delta = \gamma$, $\alpha_i = \gamma$ if $|T_i| \geq 3$, $U_1 \cap U_2 \subseteq \Omega(G)$ and no vertex in $U_1 \cap U_2$ is adjacent to any vertex outside $U_1 \cup U_2$. Therefore, (i)-(iii) are justified.

In view of (iii), it is easy to see that (iv) holds. ∎

The lemma below will be used to estimate the computational complexities of the algorithms to be designed in subsequent sections.

Lemma 2.9. *Let $G = (V, E)$ be a multigraph with a nonnegative integral weight $c(e)$ on each edge e. Then the smallest difference between the two different possible values of $\frac{2c(U)}{|U|-1}$ is at least $\frac{2}{(n-1)(n-2)}$, where U is an odd set of G and $n = n(G)$.*

Proof. Since

$$\frac{2c(U)}{|U| - 1} \in \Big\{ \frac{2m'}{n' - 1} : 0 \leq m' \leq m, \, 3 \leq n' \leq n \Big\},$$

where $m = \sum_{e \in E} c(e)$, and both m' and n' are integers, the difference θ between two different possible values of $\frac{2c(U)}{|U|-1}$ is

$$\theta = \left| \frac{2m_1}{n_1 - 1} - \frac{2m_2}{n_2 - 1} \right| = \frac{2|m_1(n_2 - 1) - m_2(n_1 - 1)|}{(n_1 - 1)(n_2 - 1)}.$$

If $n_1 = n_2$, then $\theta = \frac{2|m_1 - m_2|}{n_1 - 1} \geq \frac{2}{n_1 - 1} \geq \frac{2}{n-1} \geq \frac{2}{(n-1)(n-2)}$. Otherwise, symmetry allows us to assume that $n_1 > n_2$. Thus $(n_1 - 1)(n_2 - 1) \leq (n_1 - 1)(n_1 - 2) \leq (n - 1)(n - 2)$. It follows that $\theta \geq \frac{2}{(n_1-1)(n_2-1)} \geq \frac{2}{(n-1)(n-2)}$, as desired. ∎

3 Densities

In this section we present a strongly polynomial-time algorithm for the weighted density problem (WDP), whose output is an optimal odd set in the input multigraph.

Given a multigraph $G = (V, E)$ with a weight $w(e)$ on each edge e, the *simplification* of G is the weighted simple graph $G^* = (V, E^*)$, such that two vertices are adjacent in G^* iff they are adjacent in G, and that the weight $w^*(e)$ on each edge e in G^* is $\sum_{f \in E_G(e)} w(f)$, where $E_G(e)$ stands for the set of all edges between u and v in G for each edge $e = uv$ of G^*. Clearly, G^* can be constructed in time $O(m)$, where $m = m(G)$. Note that for each $U \subseteq V$, we have $w^*(U) = w(U)$ and $w[U, \overline{U}] = w^*[U, \overline{U}]$.

Replacing G by G^* if necessary, we may assume that G is a simple graph throughout this section, unless otherwise stated. As a consequence, we need to add $O(m)$ to the computational complexity of an algorithm in most cases, when we address the original multigraph G.

Since WDP has a fractional objective function, we shall appeal to a classical method for fractional programming. Recall that a *fractional programming* problem is generally of the form

$$\alpha(\boldsymbol{x}^*) = \max_{\boldsymbol{x} \in S} \Big\{ \alpha(\boldsymbol{x}) = \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \Big\}, \tag{3.1}$$

where $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are real-valued functions on a subset S of $\mathbb{R}^n$, and $g(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in S$. Isbell and Marlow [13] observed that (3.1) is closely related to the following problem:

$$z(\boldsymbol{x}^*, \alpha) = \max_{\boldsymbol{x} \in S} \{ z(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) - \alpha g(\boldsymbol{x}) \}, \tag{3.2}$$

where α is a real constant, in the sense that $\boldsymbol{x}^*$ solves (3.1) iff $(\boldsymbol{x}^*, \alpha^*)$ solves (3.2) for $\alpha = \alpha^* = \alpha(\boldsymbol{x}^*)$ giving the value $z(\boldsymbol{x}^*, \alpha^*) = 0$. They also proposed an iterative method for the case when both f and g are linear, which generates a sequence of solutions to the latter problem until the above optimality criterion is satisfied. When restricted to WDP, S is the family of all odd sets of G, $f(U) = 2w(U)$, and $g(U) = |U| - 1$ for each odd set U. Thus Isbell and Marlow's method [13] goes as follows.

Algorithm 3.1 for WDP

Step 0. Let U_0 be an arbitrary odd set of G. Compute $\alpha_0 = \frac{2w(U_0)}{|U_0|-1}$ and set $k = 0$.

Step 1. Solve the problem

$$z(U_{k+1}, \alpha_k) = \max_{\substack{U \subseteq V \\ |U| \geq 3, \text{ odd}}} \{ z(U, \alpha_k) = 2w(U) - \alpha_k(|U| - 1) \}, \tag{3.3}$$

obtaining a solution U_{k+1}.

Step 2. If $z(U_{k+1}, \alpha_k) = 0$, stop: $U^* = U_{k+1}$ is an optimal solution. Else, set $\alpha_{k+1} = \frac{2w(U_{k+1})}{|U_{k+1}|-1}$ and $k = k + 1$, return to Step 1.

Clearly, the technical part of this algorithm is to solve (3.3), which can be reduced to a certain generalized minimum T-cut problem, as we shall prove.

Let $H = (V, E)$ be a *simple* graph with a rational weight $c(e)$ (possibly negative) on each edge e, and let $T \subseteq V$ with $|T|$ even. As defined before, for each $U \subseteq V$, $[U, \overline{U}]$ is the set of all edges of H with precisely one end in U; we call $[U, \overline{U}]$ a *cut* and call $c[U, \overline{U}] = \sum_{e \in [U, \overline{U}]} c(e)$ the *weight* of $[U, \overline{U}]$. This cut is referred to as a *T-cut* if $|T \cap U|$ is odd, and as a *minimum T-cut* if it is a T-cut with minimum weight. The *generalized minimum T-cut problem* is to find a minimum T-cut; this problem is so named because it is a generalization of the classical *minimum T-cut problem*, where $c(e) \geq 0$ for all edges e. Note that if $c(e) < 0$ for each edge e and $T = \{s, t\}$, then the generalized minimum T-cut problem is equivalent to finding a maximum s-t cut with respect to H and $-\boldsymbol{c}$. So this generalized version contains the maximum cut problem as a special case (simply exhaust all possible pairs s, t of vertices), and hence is NP-hard in general.

Padberg and Rao [18] proposed a strongly polynomial-time algorithm for the (classical) minimum T-cut problem, which runs in time $O(n^2 m \log(n^2/m))$, where $n = |V|$ and $m = |E|$. We define a few terms before describing their algorithm. Let s, t be two vertices of H and let $[U, \overline{U}]$ be a cut. We say that $[U, \overline{U}]$ is an s-t cut if $|\{s, t\} \cap U| = 1$. A *Gomory-Hu tree* for H and $\boldsymbol{c}$ is a tree $K = (V, F)$, such that for each edge $e = st$ of K, $[U_e, \overline{U}_e]$ is a minimum s-t cut of H, where U_e is any of the two components of $K - e$; such a cut is called a *fundamental cut* with respect to K. (Note that K is not required to be a subgraph of G.) Gomory and Hu [8] showed that for each H and $\boldsymbol{c}$, there indeed exists a Gomory-Hu tree, and that it can be found in $O(n\tau)$ time, if for any $s, t \in V$ a minimum s-t cut can be found in time τ (see also Schrijver [20], Corollary 15.15a). In view of the complexity of the Goldberg-Tarjan algorithm [5] for the maximum-flow problem, we obtain $\tau = O(nm \log(n^2/m))$. Padberg and Rao [18] proved that one of the fundamental cuts is a minimum T-cut of H (see also Schrijver [20], Theorem 29.6). Thus their algorithm proceeds by first constructing a Gomory-Hu tree for H and $\boldsymbol{c}$, and then finding the fundamental cut that is a minimum T-cut.

Recall that the correctness of the Gomory-Hu tree argument is based on the submodular inequality satisfied by the cut function, which is no longer valid in the presence of negative weights. So the Padberg-Rao algorithm does not work for the generalized minimum T-cut problem we consider. Fortunately, all edges with negative weights involved in our problem are incident with a certain vertex; in this case, we can reduce our problem to a restricted version of the minimum T-cut problem.

Let $\boldsymbol{c}$ be a nonnegative weight function on E, and let s, t be two distinct vertices in H. A T-cut $[U, \overline{U}]$ of H is called an *s-t T-cut* if U contains s but not t. The *minimum s-t T-cut problem* is to find an s-t T-cut with minimum weight. As pointed out by Grötschel, Lovász, and Schrijver [11] (see page 191), this problem can be solved in polynomial time by using their characterization and algorithm [10, 11] developed for minimizing submodular functions over families of sets. Goemans and Ramakrishnan [4] (see page 507) gave a detailed description of this algorithm: For each pair of vertices $\{a, b\}$ with $a \neq t$ and $b \neq s$ in H, find a minimum $\{s, a\}$-$\{t, b\}$ cut $[S_{ab}, \overline{S}_{ab}]$ with S_{ab} minimal, and then choose in the collection $\{S_{ab} : a, b \in V, a \neq t, b \neq s\}$ a set S, such that $|T \cap S|$ is odd and $c[S, \overline{S}]$ is minimum. Goemans and Ramakrishnan [4] proved that (see Theorem 2 on page 502) such a set S exists and $[S, \overline{S}]$ is a minimum s-t T-cut. From this description we see that the minimum s-t T-cut problem can be reduced to a sequence of $O(n^2)$ minimum s-t cut problems, and hence is solvable in time $O(n^3 m \log(n^2/m))$.

Lemma 3.1. *Let $H = (V, E)$ be a simple graph with a rational weight $c(e)$ (possibly negative) on each edge e, and let $T \subseteq V$ with $|T|$ even. Suppose all edges with negative weights are incident with a distinguished vertex s, if any. Then a minimum T-cut for H and $\boldsymbol{c}$ can be found in time*

$O(n^2 m \log(n^2/m))$ *if all weights are nonnegative and in time* $O(n^3 m \log(n^2/m))$ *otherwise.*

Proof. If $c(e) \geq 0$ for all edges e, then a minimum T-cut can be determined in time $O(nm \log(n^2/m))$ by using the Padberg-Rao algorithm [18]. So we assume that $c(e) < 0$ for some edges e. For convenience, we further assume that H is a complete graph, otherwise we can add edges to H and assign 0 as their weights. Let $Z = \{v \in V : c(sv) < 0\}$. Then $Z \neq \emptyset$. Let a be a sufficiently large integer such that $c(sv) + a \geq 0$ for all $v \in Z$. Let H' be obtained from H by adding a new vertex t and adding an edge between t and each vertex of H with weight 0; we still use $\boldsymbol{c}$ to denote this extension of $\boldsymbol{c}$ to H'.

Let $\boldsymbol{c}'$ be obtained from $\boldsymbol{c}$ by replacing $c(sv)$ and $c(vt)$ with $c(sv)+a$ and $c(vt)+a$, respectively, if $v \in Z$. Let $[U, \overline{U}]$ be a T-cut of H. Renaming U and $\overline{U}$ if necessary, we may assume that $s \in U$. Then $[U, \overline{U} \cup \{t\}]$ is an s-t T-cut in H', with $c'[U, \overline{U} \cup \{t\}] = c[U, \overline{U}] + a|Z|$. Conversely, let $[U, \overline{U} \cup \{t\}]$ be an s-t T-cut of H'. Then $[U, \overline{U}]$ is a T-cut in H, with $c[U, \overline{U}] = c'[U, \overline{U} \cup \{t\}] - a|Z|$. Since $a|Z|$ is a fixed constant, a T-cut $[U, \overline{U}]$ with $s \in U$ is minimum in H with respect to $\boldsymbol{c}$ iff $[U, \overline{U} \cup \{t\}]$ is a minimum s-t T-cut in H' with respect to $\boldsymbol{c}'$. As a minimum s-t T-cut in H' with respect to $\boldsymbol{c}'$ can be determined in time $O(n^3 m \log(n^2/m))$ by using Goemans and Ramakrishnan's algorithm [4], a minimum T-cut for H and $\boldsymbol{c}$ can be found in time $O(n^3 m \log(n^2/m))$. ∎

Now we are ready to establish the correctness of Algorithm 3.1 and estimate its computational complexity.

Lemma 3.2. *If* $\alpha_k \geq \Delta_w(G)$*, then the optimal solution* U_{k+1} *to problem (3.3) in Step 1 can be found in time* $O(n^2 \ell \log(n^2/\ell))$*; otherwise, it can be found in time* $O(n^3 \ell \log(n^2/\ell))$*, where* $n = n(G)$ *and* $\ell = \ell(G)$*.*

Proof. The objective function of problem (3.3) is $z(U, \alpha_k) = 2w(U) - \alpha_k(|U| - 1) = \sum_{v \in U} d_{G,w}(v) - w[U, \overline{U}] - \alpha_k(|U| - 1)$. So

(1) $z(U, \alpha_k) = \sum_{v \in U}(d_{G,w}(v) - \alpha_k) - w[U, \overline{U}] + \alpha_k$.

Since α_k is a fixed constant, from (1) we see that solving (3.3) is equivalent to solving:

$$\min_{\substack{U \subseteq V \\ |U| \geq 3, \text{ odd}}} \; w[U, \overline{U}] + \sum_{v \in U} (\alpha_k - d_{G,w}(v)) \tag{3.4}$$

Let us show that this problem is equivalent to a minimum T-cut problem. To justify this, let $G' = (V', E')$ be the weighted graph obtained from $G = (V, E)$ by adding a dummy vertex r and adding an edge between r and each vertex of G, such that

- for each edge $e \in E$, its weight in G' is $c(e) = w(e)$; and
- for each edge rv with $v \in V$, its weight in G' is $c(rv) = \alpha_k - d_{G,w}(v)$.

This construction is due to Padberg and Rao [18].

Let $T = V$ if $|V|$ is even and $T = V \cup \{r\}$ otherwise. Note that every T-cut of G' is of the form $[U, \overline{U} \cup \{r\}]$, where $U \subseteq V$ with $|U|$ odd and $\overline{U} = V \backslash U$. The capacity of such a T-cut is

(2) $c[U, \overline{U} \cup \{r\}] = w[U, \overline{U}] + \sum_{v \in U}(\alpha_k - d_{G,w}(v))$.

If $\alpha_k \geq \Delta_w(G)$, then $c(e) \geq 0$ for all edges e, we can find a minimum T-cut $[X, \overline{X} \cup \{r\}]$ for G' and $\boldsymbol{c}$ in time $O(n^2 \ell \log(n^2/\ell))$ by Padberg-Rao algorithm. Assume $\alpha_k < \Delta_w(G)$. Since all edges e with negative weights $c(e)$, if any, are incident with r, we can find a minimum T-cut

$[X, \overline{X} \cup \{r\}]$ for G' and $\boldsymbol{c}$ in time $O(n^3 \ell \log(n^2/\ell))$ by Lemma 3.1, where $X \subseteq V$ with $|X|$ odd and $\ell = \ell(G) = |E^*|$.

If $|X| \geq 3$, then $U_{k+1} = X$ is an optimal solution to (3.4) and hence to (3.3).

If $|X| = 1$, letting $X = \{x\}$, then $c[X, \overline{X} \cup \{r\}] = d_{G,w}(x) + (\alpha_k - d_{G,w}(x)) = \alpha_k$ by the definition of $\boldsymbol{c}$. Since for any $U \subseteq V$ with $|U|$ odd, $c[X, \overline{X} \cup \{r\}] \leq c[U, \overline{U} \cup \{r\}]$, it follows from (2) that

(3) $w[U, \overline{U}] + \sum_{v \in U} (\alpha_k - d_{G,w}(v)) \geq \alpha_k$.

Combining (1) and (3), we see that

(4) $z(U, \alpha_k) \leq 0$ for any $U \subseteq V$ with $|U|$ odd.

Since $z(U_k, \alpha_k) = 2w(U_k) - \alpha_k(|U_k| - 1) = 0$ by the definition of α_k and $|U_k| \geq 3$, from (4) we deduce that $U_{k+1} = U_k$ is an optimal solution to (3.3).

So the optimal solution U_{k+1} in Step 1 can be found in time $O(n^2 \ell \log(n^2/\ell))$ if $\alpha_k \geq \Delta_w(G)$, and in time $O(n^3 \ell \log(n^2/\ell))$ otherwise. ∎

Lemma 3.3. *Algorithm 3.1 terminates in n or fewer iterations, where $n = n(G)$.*

Proof. Recall that for each iteration $k \geq 0$, if $z(U_{k+1}, \alpha_k) = 0$, then the algorithm terminates with output $U^* = U_{k+1}$; otherwise, $z(U_{k+1}, \alpha_k) > 0$, so $2w(U_{k+1}) - \alpha_k(|U_{k+1}| - 1) > 0$. By the definition of α_{k+1}, we have

(1) $\alpha_{k+1} > \alpha_k$ for $k \geq 0$.

Consider an iteration $k \geq 1$ with $z(U_{k+1}, \alpha_k) > 0$. Clearly, $z(U_k, \alpha_{k-1}) > 0$ as well. Note that

$$\begin{aligned} z(U_{k+1}, \alpha_k) &= 2w(U_{k+1}) - \alpha_k(|U_{k+1}| - 1) \\ &= [2w(U_{k+1}) - \alpha_{k-1}(|U_{k+1}| - 1)] + [\alpha_{k-1}(|U_{k+1}| - 1) - \alpha_k(|U_{k+1}| - 1)] \\ &= [2w(U_{k+1}) - \alpha_{k-1}(|U_{k+1}| - 1)] - (\alpha_k - \alpha_{k-1})(|U_{k+1}| - 1). \end{aligned}$$

By the definitions of U_k and α_k, we obtain

$$2w(U_{k+1}) - \alpha_{k-1}(|U_{k+1}| - 1)] \leq 2w(U_k) - \alpha_{k-1}(|U_k| - 1) = (\alpha_k - \alpha_{k-1})(|U_k| - 1).$$

So

(2) $0 < z(U_{k+1}, \alpha_k) \leq (\alpha_k - \alpha_{k-1})(|U_k| - |U_{k+1}|)$.

It follows from (1) that

(3) $|U_k| > |U_{k+1}|$ for $k \geq 1$.

As $3 \leq |U_k| \leq n$ for all $k \geq 1$, we conclude from (3) that Algorithm 3.1 terminates within n iterations. ∎

Note that Theorem 1.3 follows instantly from Lemmas 3.2 and 3.3. If $\Gamma_w(G) \geq \Delta_w(G)$, then the running time in Theorem 1.3 can be improved to $O(m + n^3 \ell \log(n^2/\ell))$, as shown below.

Lemma 3.4. *Let G be a multigraph with a positive rational weight $w(e)$ on each edge e. Then the following two statements hold:*

(i) We can determine in time $O(n^2 \ell \log(n^2/\ell))$ whether $\Gamma_w(G) \geq \Delta_w(G)$.

(ii) If $\Gamma_w(G) \geq \Delta_w(G)$, then an optimal odd set of G can be found in time $O(m + n^3 \ell \log(n^2/\ell))$, where $n = n(G)$, $m = m(G)$, and $\ell = \ell(G)$.

Proof. (i) Let U be an optimal odd set. Since $2w(U) + w[U, \overline{U}] = \sum_{v \in U} d_{G,w}(v)$, we may rewrite $\Gamma_w(G) \geq \Delta_w(G)$ in an equivalent form as follows:

$$w[U, \overline{U}] + \sum_{v \in U} (\Delta_w(G) - d_{G,w}(v)) \leq \Delta_w(G). \tag{3.5}$$

Let $G' = (V', E')$ be the weighted graph obtained from $G = (V, E)$ by adding a dummy vertex r and adding an edge between r and each vertex of G, such that

- for each edge $e \in E$, its weight in G' is $c(e) = w(e)$; and
- for each edge rv with $v \in V$, its weight in G' is $c(rv) = \Delta_w(G) - d_{G,w}(v)$.

This construction is also due to Padberg and Rao [18].

Let $T = V$ if $|V|$ is even and $T = V \cup \{r\}$ otherwise. Note that every T-cut of G' is of the form $[U, \overline{U} \cup \{r\}]$, where $U \subseteq V$ with $|U|$ odd and $\overline{U} = V \backslash U$. The weight of such a T-cut is

$$c[U, \overline{U} \cup \{r\}] = w[U, \overline{U}] + \sum_{v \in U} (\Delta_w(G) - d_{G,w}(v)). \tag{3.6}$$

Using (3.5) and (3.6), we deduce that $\Gamma_w(G) \geq \Delta_w(G)$ iff G' has a T-cut $[U, \overline{U} \cup \{r\}]$ with weight $c[U, \overline{U} \cup \{r\}]$ at most $\Delta_w(G)$. Since we can find a minimum T-cut $[X, \overline{X} \cup \{r\}]$ for G' and $\boldsymbol{c}$ in time $O(m + n^2 \ell \log(n^2/\ell))$, statement (i) is true.

(ii) Let $[U_0, \overline{U}_0 \cup \{r\}]$ be a minimum T-cut for G' and $\boldsymbol{c}$ returned by the Padberg-Rao algorithm. Since $\Gamma_w(G) \geq \Delta_w(G)$, we have $c[U_0, \overline{U}_0 \cup \{r\}] \leq \Delta_w(G)$. It follows that $\frac{2w(U_0)}{|U_0|-1} \geq \Delta_w(G)$. So we can choose U_0 with $\alpha_0 = \frac{2w(U_0)}{|U_0|-1} \geq \Delta_w(G)$ in Step 0 of Algorithm 3.1. From the proof of Lemma 3.3, we see that the sequence of values $\{\alpha_k\}$ generated by Algorithm 3.1 is increasing. Thus $\alpha_k \geq \alpha_0 \geq \Delta_w(G)$ for each iteration k and hence, by Lemma 3.2, the optimal solution U_k to problem (3.3) in Step 1 can be found in time $O(n^2 \ell \log(n^2/\ell))$. Therefore statement (ii) follows directly from Lemma 3.3. ∎

4 Matchings

In this section we devise efficient algorithms for finding matchings as specified in Theorem 1.5. Replacing G by its simplification G^* if necessary, we again assume that G is a simple graph throughout this section.

To facilitate a better understanding of Caprara and Rizzi's theorem [2], we give a sketch of their proof and construction below.

Proof sketch of Theorem 1.4. Recall that G is assumed to be a simple graph. Let $X = \Omega(G)$. Caprara and Rizzi [2] observed that G does not contain an X-matching iff it contains a vertex subset S, such that $G \backslash S$ contains strictly more than $|S|$ odd components containing only vertices in X (see Lemma 4 in [2]). Let $T_1, T_2, \ldots, T_p$ be components of $G \backslash S$, with $p > |S|$, $|T_i|$ odd, and $T_i \subseteq X$ for $1 \leq i \leq p$. They then proved that

(1) $2w(T_i)/(|T_i| - 1) > \Delta_w(G)$ for some i,

for otherwise, $2w(T_i)/(|T_i| - 1) \leq \Delta_w(G)$ for all i. As all vertices in each T_i have degree $\Delta_w(G)$, we have $\Delta_w(G)|T_i| = 2w(T_i) + w[T_i, \overline{T}_i]$, so $w[T_i, \overline{T}_i] \geq \Delta_w(G)$. Hence $w[S, \overline{S}] =$

$\sum_{i=1}^{p} w[T_i, \overline{T}_i] \geq p\Delta_w(G) > |S|\Delta_w(G)$; this contradiction establishes (1). Thus $\Gamma_w(G) > \Delta_w(G)$, contradicting the hypothesis of this theorem.

To find the desired matching, Caprara and Rizzi [2] took two copies $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ of G, and constructed a graph $\tilde{G} = (\tilde{V}, \tilde{E})$, which arises from the disjoint union of G_1 and G_2 by adding an edge v_1v_2 for each $v \in V \backslash X$, where v_i is the vertex corresponding to v in G_i for $i = 1, 2$. Clearly, G contains an X-matching iff $\tilde{G}$ contains a perfect matching. Thus an X-matching can be found in G in time $O(\sqrt{n}\ell)$ (see Schrijver [20], page 423). ∎

Lemma 4.1. *Let G be a simple graph with a positive rational weight $w(e)$ on each edge e, and let M be a matching of G. Then we can find an OoS of G that is not saturated by M, if any, in time $O(n^4 \ell \log(n^2/\ell))$, where $n = n(G)$ and $\ell = \ell(G)$.*

Proof. For each edge e, write $w(e) = \frac{p(e)}{q(e)}$, where $p(e)$ and $q(e)$ are relatively prime integers. Let $\epsilon_w(G) = \frac{1}{\Pi_{e \in E} q(e)}$ and let

$$\Gamma_w^-(G) = \max \Big\{ \frac{2w(U)}{|U|-1} : \ U \text{ is an odd set outside } \mathcal{O}(G) \Big\}.$$

We claim that

(1) $\Gamma_w^-(G) \leq \Gamma_w(G) - \frac{2\epsilon_w(G)}{(n-1)(n-2)}$.

To justify this, let $c(f) = w(f) \cdot \Pi_{e \in E} q(e)$ for all $f \in E$. Then (1) is equivalent to saying that

(2) $\Gamma_c^-(G) \leq \Gamma_c(G) - \frac{2}{(n-1)(n-2)}$,

which follows instantly from Lemma 2.9. So (1) holds.

Define $w'(e) = w(e) - \frac{\epsilon_w(G)}{(n-1)(n-2)}$ if $e \in M$ and $w'(e) = w(e)$ otherwise for each edge e. Note that for each odd set U, if U is an OoS, then $\frac{2w'(U)}{|U|-1} = \Gamma_w(G) - \frac{\epsilon_w(G)}{(n-1)(n-2)} \cdot \frac{2|E(U) \cap M|}{|U|-1} \geq \Gamma_w(G) - \frac{\epsilon_w(G)}{(n-1)(n-2)}$; otherwise, $\frac{2w'(U)}{|U|-1} \leq \frac{2w(U)}{|U|-1} \leq \Gamma_w^-(G) \leq \Gamma_w(G) - \frac{2\epsilon_w(G)}{(n-1)(n-2)}$, here the last inequality follows from (1). Thus, we deduce that M restricts to a near-perfect matching on $G[U]$ for all OoS's U iff $\Gamma_{w'}(G) = \Gamma_w(G) - \frac{\epsilon_w(G)}{(n-1)(n-2)}$. By Theorem 1.3, an OoS U of G with respect to $\boldsymbol{w'}$ can be found in time $O(n^4 \ell \log(n^2/\ell))$. We thus conclude that

- if $\frac{2w'(U)}{|U|-1} > \Gamma_w(G) - \frac{\epsilon_w(G)}{(n-1)(n-2)}$, then U is not saturated by M;
- otherwise, all OoS's of G are saturated by M. ∎

Let V be a set and let X, Y be two subsets of V. We say that X and Y are *crossing* if the sets $X \backslash Y$, $Y \backslash X$, and $X \cap Y$ are all nonempty. A family $\mathcal{C}$ of subsets of V is called *laminar* if no two of them are crossing. It is well known that a laminar family $\mathcal{C}$ has a *Venn-diagram* representation: the ith level consists of all sets $X \in \mathcal{C}$, such that there are sets Y at the $(i-1)$th level with $X \subset Y$. Thus each level consists of disjoint sets, and for each set X of level $i+1$ there is a unique set of level i containing X. It follows that $\mathcal{C}$ has a rooted tree-representation as well (see Schrijver [20], pages 214 and 215, for details).

The following lemma (see Schrijver [20], Theorem 3.5) gives an upper bound on the size of a laminar family.

Lemma 4.2. *If $\mathcal{C}$ is a laminar family and $V \neq \emptyset$, then $|\mathcal{C}| \leq 2|V|$.*

Laminar families will play an important role in our search for desired matchings.

Lemma 4.3. *Let G be a simple graph with a positive rational weight $w(e)$ on each edge e and with $\Delta_w(G) < \Gamma_w(G)$, and let $\mathcal{O}$ be a laminar family of OoS's of G with a given Venn-diagram representation. Then we can find an $\mathcal{O}$-matching of G in time $O(\sqrt{n}\ell)$, where $n = n(G)$ and $\ell = \ell(G)$.*

Proof. We aim to find an $\mathcal{O}$-matching M of G in time $O(\sqrt{n}\ell)$ by using a recursive algorithm, where $n = n(G)$ and $\ell = \ell(G)$.

Consider the case when $|\mathcal{O}| = 1$. Let $\mathcal{O} = \{X\}$. Then Lemma 2.4(i) guarantees the existence of a near-perfect matching M in $G[X]$, which can be found in time $O(\sqrt{p}q)$ (see Schrijver [20], page 423), where p (resp. q) is the number of vertices (resp. edges) in $G[X]$.

Suppose $|\mathcal{O}| \geq 2$. Let X be a minimal (with respect to set inclusion) OoS in $\mathcal{O}$, let H be the multigraph obtained from G by contracting X into a single vertex x, and let H^* be the simplification of H. We use $w^*(e)$ to denote the weight on each edge e of H^*. By Lemma 2.6(i), we have $\Delta_{w^*}(H^*) \leq \Delta_w(G)$ and $\Gamma_{w^*}(H^*) = \Gamma_w(G)$. Let $\mathcal{O}^*$ be obtained from $\mathcal{O}$ by contracting X; that is, for each Y in $\mathcal{O}$ with $X \cap Y = \emptyset$, the set $Y \in \mathcal{O}^*$; for each Y in $\mathcal{O}$ with $X \subset Y$ and $X \neq Y$, the set $(Y \backslash X) \cup \{x\} \in \mathcal{O}^*$. From the Venn-diagram representation of $\mathcal{O}$, we see that $\mathcal{O}^*$ also admits such a representation. Hence, by Lemma 2.6(ii), $\mathcal{O}^*$ is a laminar family of OoS's in H^*. Note that $|\mathcal{O}^*| = |\mathcal{O}| - 1$ and that $\Delta_{w^*}(H^*) < \Gamma_{w^*}(H^*)$, because $\Delta_w(G) < \Gamma_w(G)$.

Let M^* be an $\mathcal{O}^*$-matching in H^* outputted by our recursive algorithm. Let u be the vertex in X incident with an edge in M^*, if any, and an arbitrary vertex in X otherwise that has degree less than $\Delta_w(G)$ in G (see Lemma 2.3(i)). By Lemma 2.4(i), $G[X]$ contains a near-perfect matching N that is disjoint from u. Clearly, $M = M^* \cup N$ is an $\mathcal{O}$-matching in G.

Let n_1 (resp. n_2) be the number of vertices H^* (resp. $G[X]$), and let m_1 (resp. m_2) be the number of edges of H^* (resp. $G[X]$). Then M^* (resp. N) can be found in time $O(\sqrt{n_1}m_1)$ (resp. $O(\sqrt{n_2}m_2)$). Therefore M can be found in time $O(\sqrt{n_1}m_1) + O(\sqrt{n_2}m_2) = O(\sqrt{n}(m_1 + m_2)) = O(\sqrt{n}\ell)$. ∎

Lemma 4.4. *Let G be a simple graph with a positive rational weight $w(e)$ on each edge e and with $\Delta_w(G) = \Gamma_w(G)$, and let $\mathcal{O}$ be a laminar family of OoS's of G with a given Venn-diagram representation. Then we can find an $\{\Omega(G), \mathcal{O}\}$-matching of G in time $O(\sqrt{n}\ell)$, where $n = n(G)$ and $\ell = \ell(G)$.*

Proof. We aim to find an $\{\Omega(G), \mathcal{O}\}$-matching M in time $O(\sqrt{n}\ell)$ by using a recursive algorithm, which is a slight modification of the one employed in the proof of the preceding lemma, where $n = n(G)$ and $\ell = \ell(G)$. At each iteration, we consider an intermediate graph H^* and a laminar family $\mathcal{O}^*$ of OoS's in H^*, and aim to find an $\mathcal{O}^*$-matching M^* in H^* that covers all vertices of degree $\Delta_w(G)$ (rather than $\Delta_{w^*}(H^*)$) in H^*, if any.

Consider the case when $|\mathcal{O}| = 1$. Let $\mathcal{O} = \{X\}$, let H be the multigraph obtained from G by contracting X into a single vertex x, and let H^* be the simplification of H. We use $w^*(e)$ to denote the weight on each edge e of H^*. If H^* contains no vertex of degree $\Delta_w(G)$, set $M^* = \emptyset$; otherwise, we have $\Delta_{w^*}(H^*) = \Delta_w(G)$, and by Lemma 2.3(iii) $\Gamma_w(G) \geq \Gamma_{w^*}(H^*)$, so Theorem

1.4 guarantees the existence of a matching M^* in H^* that covers all vertices of degree $\Delta_w(G)$. Let u be the vertex in X incident with an edge in M^*, if any, and an arbitrary vertex in X otherwise that has weighted degree less than $\Delta_w(G)$ in G (see Lemma 2.3(i)). By Lemma 2.4, $G[X]$ contains a near-perfect matching N that is disjoint from u. Then $M = M^* \cup N$ is an $\{\Omega(G), \mathcal{O}\}$-matching in G, which can be found in time $O(\sqrt{n}\ell)$.

Suppose $|\mathcal{O}| \geq 2$. Let X be a minimal (with respect to set inclusion) OoS in $\mathcal{O}$, let H be the multigraph obtained from G by contracting X into a single vertex x, and let H^* be the simplification of H. We again use $w^*(e)$ to denote the weight on each edge e of H^*. By Lemma 2.6(i), we have $\Delta_{w^*}(H^*) \leq \Delta_w(G)$ and $\Gamma_{w^*}(H^*) = \Gamma_w(G)$. As in the proof of the preceding lemma, let $\mathcal{O}^*$ be obtained from $\mathcal{O}$ by contracting X. Once again, by Lemma 2.6(i), $\mathcal{O}^*$ is a laminar family of OoS's in H^* with a Venn-diagram representation, and $|\mathcal{O}^*| = |\mathcal{O}| - 1$.

If H^* contains no vertex of degree $\Delta_w(G)$, then $\Delta_{w^*}(H^*) < \Gamma_{w^*}(H^*)$; in this case, our recursive algorithm returns an $\mathcal{O}^*$-matching M^* in H^* (see also Lemma 4.3). If H^* contains a vertex of degree $\Delta_w(G)$, then $\Delta_{w^*}(H^*) = \Gamma_{w^*}(H^*)$; let M^* be an $\mathcal{O}^*$-matching that covers all vertices of degree $\Delta_w(G)$ in H^* outputted by our recursive algorithm. Let u be the vertex in X incident with an edge in M^*, if any, and a vertex in X that has degree less than $\Delta_w(G)$ in G (see Lemma 2.3(i)) otherwise. By Lemma 2.4(i), $G[X]$ contains a near-perfect matching N that is disjoint from u. Clearly, $M = M^* \cup N$ is an $\{\Omega(G), \mathcal{O}\}$-matching in G, which can be found in time $O(\sqrt{n}\ell)$. ∎

Let $G = (V, E)$ be a multigraph with a positive rational weight $w(e)$ on each edge e, and let U_1 and U_2 be two OoS's of G with $|U_1 \cap U_2| > 0$ and with $U_1 \backslash U_2 \neq \emptyset \neq U_2 \backslash U_1$.

If $|U_1 \cap U_2|$ is odd, then the operation of replacing $\{U_1, U_2\}$ with $\{U_1 \cap U_2, U_1 \cup U_2\}$ (deleting $U_1 \cap U_2$ if its size is one) is our *type-I uncrossing* technique. By Lemma 2.7(iii), if a matching of G restricts to a near-perfect matching on both $G[U_1 \cap U_2]$ and $G[U_1] \cup G[U_2]$, then it also restricts to a near-perfect matching on $G[U_i]$ for $i = 1, 2$.

If $|U_1 \cap U_2|$ is even, then the operation of replacing $\{U_1, U_2\}$ with $\{U_1 \backslash U_2, U_2 \backslash U_1\}$ (deleting $U_i \backslash U_{3-i}$ if its size is one for $i = 1, 2$) is our *type-II uncrossing* technique. By Lemma 2.8(iv), if a matching of G covers all vertices in $U_1 \cap U_2$ and restricts to a near-perfect matching on both $G[U_1] \backslash U_2$ and $G[U_1] \backslash U_2$, then it also restricts to a near-perfect matching on $G[U_i]$ for $i = 1, 2$.

Let $\mathcal{O}$ be a laminar family of OoS's in G, let M be an $\mathcal{O}$-matching in G that covers all vertices of degree $\Delta_w(G)$ if $\Delta_w(G) = \Gamma_w(G)$, and let U be an OoS that is not saturated by M. We apply the following algorithm to uncross the triple $(\mathcal{O}, U, M)$ and generate a larger laminar family of OoS's.

Algorithm 4.1 for uncrossing the triple $(\mathcal{O}, U, M)$

Step 0. Set $U_0 = U$ and $k = 0$.

Step 1. If $\mathcal{O}$ contains no set S such that S and U_k are crossing, stop: $\mathcal{O} = \mathcal{O} \cup \{U_k\}$ is a larger laminar family of OoS's. Else, let S be such a set in $\mathcal{O}$, go to Step 2 if $|S \cap U_k|$ is odd, and go to Step 3 otherwise.

Step 2. Set $U_{k+1} = S \cup U_k$ if $|E(S \cup U_k) \cap M| < \frac{|S \cup U_k| - 1}{2}$ and $U_{k+1} = S \cap U_k$ if $|E(S \cup U_k) \cap M| = \frac{|S \cup U_k| - 1}{2}$ otherwise. Set $k = k + 1$, return to Step 1.

Step 3. Set $U_{k+1} = S \backslash U_k$ if $|E(S \backslash U_k) \cap M| < \frac{|S \backslash U_k| - 1}{2}$ and $U_{k+1} = U_k \backslash S$ if $|E(S \backslash U_k) \cap M| =$

$\frac{|S\backslash U_k|-1}{2}$ otherwise. Set $k = k + 1$, return to Step 1.

Let us make some remarks on this algorithm.

When $|S \cap U_k|$ is odd, by Lemma 2.7, $S \cup U_k$ is an OoS, and so is $S \cap U_k$ if $|S \cap U_k| \geq 3$. Furthermore, one of the inequalities $|E(S \cup U_k) \cap M| < \frac{|S \cup U_k|-1}{2}$ and $|E(S \cap U_k) \cap M| < \frac{|S \cap U_k|-1}{2}$ holds, because U_k is not saturated by M. Thus a type-I uncrossing technique applies; that is, we may replace $\{S, U_k\}$ with $\{S \cup U_k, S \cap U_k\}$. For ease of implementation, at each iteration we only replace U_k by $S \cup U_k$ if the first inequality holds and replace U_k by $S \cap U_k$ otherwise.

When $|S \cap U_k|$ is even, by Lemma 2.8(i), we have $\Delta_w(G) = \Gamma_w(G)$. So M covers all vertices in $\Omega(G)$ by hypothesis on M. By Lemma 2.8(iii), $S \cap U_k \subseteq \Omega(G)$ and no vertex in $S \cap U_k$ is adjacent to any vertex outside $S \cup U_k$. Thus M covers all vertices in $S \cap U_k$. As U_k is not saturated by M, one of the inequalities $|E(S\backslash U_k) \cap M| < \frac{|S\backslash U_k|-1}{2}$ and $|E(U_k\backslash S) \cap M| < \frac{|U_k\backslash S|-1}{2}$ holds. So a type-II uncrossing technique applies; we replace U_k by $S\backslash U_k$ in the former case and replace U_k by $U_k\backslash S$ in the latter.

Lemma 4.5. *Algorithm 4.1 correctly finds a larger laminar family of OoS's of G than the original $\mathcal{O}$ in time $O(n^3)$, where $n = n(G)$.*

Proof. From the algorithm, it is easy to see that the number of crossing pairs is decreased by at least one at each iteration. So the total number of iterations is at most $|\mathcal{O}|$ (original size), which is $O(n)$ by Lemma 4.2. Since the resulting family $\mathcal{O}$ contains no crossing pairs, it is laminar as well. Clearly, the running time of this uncrossing algorithm is $O(n^3)$. ∎

Proof of Theorem 1.5. We start with a family $\mathcal{O} = \{S\}$, where S is an arbitrary OoS in G. Clearly, $\mathcal{O}$ is a laminar family with a trivial Venn-diagram representation. At a general step, suppose we have had a laminar family $\mathcal{O}$ of OoS's in G, with a given Venn-diagram representation. By Lemmas 4.3 and 4.4, we can first find an $\mathcal{O}$-matching M in G in time $O(\sqrt{n}\ell)$ that is also an $\Omega(G)$-matching if $\Delta_w(G) = \Gamma_w(G)$. By Lemma 4.1, we can then find an OoS U of G that is not saturated by M, if any, in time $O(n^4 \ell \log(n^2/\ell))$. If there is no such U, then M is as desired. So we assume that U is available. Given the triple $(\mathcal{O}, U, M)$, Algorithm 4.1 enables us to generate a larger laminar family $\mathcal{O}$ in time $O(n^3)$. Note that, as a by-product, we can also produce a Venn-diagram representation of the resulting $\mathcal{O}$ in time $O(n^3)$.

The process is repeated with this new $\mathcal{O}$. By Lemma 4.2, $|\mathcal{O}| \leq 2n$, so the whole algorithm terminates in $O(n)$ iterations, and therefore runs in time $O(n^5 \ell \log(n^2/\ell))$. ∎

5 Fractional Edge-Colorings

In this section we present a strongly polynomial-time algorithm for the weighted fractional edge-coloring problem (WFECP).

Let $G = (V, E)$ be a multigraph with a positive rational weight $w(e)$ on each edge e, and let M be a matching of G. We reserve the symbol $b(M)$ for $\min\{w(e) : e \in M\}$. For any $0 \leq c \leq b(M)$, we use $G - cM$ to denote the weighted multigraph obtained from G by replacing $w(e)$ with $w(e) - c$ for each $e \in M$ (we delete all edges with zero weight in $G - cM$), and use $\boldsymbol{w} - cM$ or $\boldsymbol{w}(c)$ to denote the weight function associated with $G - cM$.

Lemma 5.1. *The following two statements are true:*

(i) For any c with $0 \le c \le b(M)$, the inequality $\chi^*_{w(c)}(G - cM) \ge \chi^*_w(G) - c$ holds.

(ii) If $\chi^*_{w(t)}(G-tM) = \chi^*_w(G)-t$ for some positive constant t, then $\chi^*_{w(s)}(G-sM) = \chi^*_w(G)-s$ for all constants s with $0 < s < t$.

Proof. (i) For each vertex $v \in \Omega(G)$, if v is covered by M, then $d_{G-cM,w(c)}(v) = \Delta_w(G) - c$; otherwise, $d_{G-cM,w(c)}(v) = \Delta_w(G)$. For each OoS U, if $|E(U) \cap M| = \frac{|U|-1}{2}$, then $\frac{2w(c)(U)}{|U|-1} = \Gamma_w(G) - c$; otherwise, $\frac{2w(c)(U)}{|U|-1} = \Gamma_w(G) - \frac{2c|E(U)\cap M|}{|U|-1} > \Gamma_w(G) - c$. Therefore, $\chi^*_{w(c)}(G - cM) = \max\{\Delta_{w(c)}(G - cM), \Gamma_{w(c)}(G - cM)\} \ge \max\{\Delta_w(G) - c, \Gamma_w(G) - c\} = \chi^*_w(G) - c$, as desired.

(ii) Assume the contrary: $\chi^*_{w(s)}(G - sM) \ne \chi^*_w(G) - s$ for some s with $0 < s < t$. From (i) we deduce that $\chi^*_{w(s)}(G - sM) > \chi^*_w(G) - s$. By definition, there exists a vertex v such that $d_{G-sM,w(s)}(v) = \chi^*_{w(s)}(G - sM)$ or an odd set U such that $\frac{2w(s)(U)}{|U|-1} = \chi^*_{w(s)}(G - sM)$. If the first case occurs, then v is not covered by M, and hence $d_{G-tM,w(t)}(v) = d_{G-sM,w(s)}(v) > \chi^*_w(G) - s > \chi^*_w(G) - t$, a contradiction. If the second case occurs, then $2|E(U) \cap M| < |U| - 1$, and hence $s(1 - \frac{2|E(U)\cap M|}{|U|-1}) > \chi^*_w(G) - \frac{2w(U)}{|U|-1}$. It follows that

$$t(1 - \frac{2|E(U) \cap M|}{|U| - 1}) > s(1 - \frac{2|E(U) \cap M|}{|U| - 1}) > \chi^*_w(G) - \frac{2w(U)}{|U| - 1},$$

which implies $\chi^*_{w(t)}(G - tM) \ge \frac{2w(t)(U)}{|U|-1} = \frac{2w(U)}{|U|-1} - \frac{2t|E(U)\cap M|}{|U|-1}) > \chi^*_w(G) - t$, a contradiction again. ∎

Let $r(M)$ denote the largest value of c satisfying $\chi^*_{w(c)}(G - cM) = \chi^*_w(G) - c$; we call $r(M)$ the *residue* of M in G. From Lemma 5.1 (ii), we see that this term is well defined. Also, it is clear that $0 \le r(M) \le b(M)$.

We call M a *feasible matching* of $(G, \boldsymbol{w})$ if M is

- an $\Omega(G)$-matching when $\Delta_w(G) > \Gamma_w(G)$,
- an $\{\Omega(G), \mathcal{O}(G)\}$-matching when $\Delta_w(G) = \Gamma_w(G)$, and
- an $\mathcal{O}(G)$-matching when $\Delta_w(G) < \Gamma_w(G)$.

Let us now give a description of our algorithm.

Algorithm 5.1 for WFECP

Step 0. Set $G_1 = G$, $\boldsymbol{w}_1 = \boldsymbol{w}$, and $k = 1$.

Step 1. If $\chi^*_{w_k}(G_k) = 0$, stop: $\{(M_i, r_i) : 1 \le i \le k - 1\}$ is an optimal solution. Else, find a feasible matching M_k of $(G_k, \boldsymbol{w}_k)$, and determine the residue r_k of M_k in G_k.

Step 2. Set $G_{k+1} = G_k - r_kM_k$, $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - r_kM_k$, and $k = k + 1$, return to Step 1.

In the preceding section we have designed algorithms for finding feasible matchings. To determine the residue $r(M)$ of any given matching M in G, we propose the following algorithm, where we assume that $G - M$ is a simple graph, otherwise, replace $G - M$ by its simplification.

Algorithm 5.2 for finding residue

Step 0. Set $r_1 = b(M)$, $G_0 = G$, $G_1 = G - r_1M$, $\boldsymbol{w}_0 = \boldsymbol{w}$, $\boldsymbol{w}_1 = \boldsymbol{w} - r_1M$, and $k = 1$.

Step 1. Compute $\Delta_{w_k}(G_k)$ and $\Gamma_{w_k}(G_k)$. If

$$\chi^*_{w_k}(G_k) = \max\{\Delta_{w_k}(G_k), \Gamma_{w_k}(G_k)\} = \chi^*_w(G) - r_k,$$

stop: $r(M) = r_k$.

Step 2. If $\Delta_{w_k}(G_k) > \Gamma_{w_k}(G_k)$, set $r_{k+1} = \chi^*_w(G) - \Delta_{w_k}(G_k)$. Else, find an OoS U_k in G_k with respect to $\boldsymbol{w}_k$, and set r_{k+1} as the solution of the equation

$$\frac{2w(U_k)}{|U_k|-1} - \frac{2r|E(U_k) \cap M|}{|U_k|-1} = \chi^*_w(G) - r.$$

Step 3. Set $G_{k+1} = G - r_{k+1}M$, $\boldsymbol{w}_{k+1} = \boldsymbol{w} - r_{k+1}M$, and $k = k+1$, return to Step 1.

Lemma 5.2. *The following statements hold for the above algorithm:*

(i) $r_k > r_{k+1} \geq 0$ *for all* $k \geq 1$.

(ii) If $\Delta_{w_k}(G_k) > \Gamma_{w_k}(G_k)$ *and* $\Delta_{w_{k+1}}(G_{k+1}) \geq \Gamma_{w_{k+1}}(G_{k+1})$, *then Algorithm 5.2 terminates at iteration* $k+1$, *and* $r(M) = r_{k+1}$.

(iii) If $\Delta_{w_k}(G_k) \leq \Gamma_{w_k}(G_k)$, *then* $\Delta_{w_{k+1}}(G_{k+1}) \leq \Gamma_{w_{k+1}}(G_{k+1})$. *Furthermore, if* U_{k+1} *is defined in iteration* $k+1$, *then*

$$\frac{2|E(U_{k+1}) \cap M|}{|U_{k+1}|-1} > \frac{2|E(U_k) \cap M|}{|U_k|-1}.$$

Proof. For simplicity, write $\gamma_w(U) = \frac{2w(U)}{|U|-1}$ for each odd set U of G. Since r_{k+1} has been defined in the algorithm, from Step 1 and Lemma 5.1(i), we see that

(1) $\chi^*_{w_k}(G_k) = \max\{\Delta_{w_k}(G_k), \Gamma_{w_k}(G_k)\} > \chi^*_w(G) - r_k$.

(i) To prove that $r_k > r_{k+1} \geq 0$, consider Step 2. If $\Delta_{w_k}(G_k) > \Gamma_{w_k}(G_k)$, then $r_{k+1} = \chi^*_w(G) - \Delta_{w_k}(G_k) < r_k$ by (1). If $\Delta_{w_k}(G_k) \leq \Gamma_{w_k}(G_k)$, then $\frac{2w(U_k)}{|U_k|-1} - \frac{2r_{k+1}|E(U_k)\cap M|}{|U_k|-1} = \chi^*_w(G) - r_{k+1}$ by definition. So $r_{k+1} = \frac{(\chi^*_w(G)-\gamma_w(U_k))(|U_k|-1)}{|U_k|-1-2|E(U_k)\cap M|} \geq 0$. Since U_k is an OoS in G_k with respect to $\boldsymbol{w}_k$, by (1), we have $\Gamma_{w_k}(G_k) = \frac{2w(U_k)}{|U_k|-1} - \frac{2r_k|E(U_k)\cap M|}{|U_k|-1} > \chi^*_w(G) - r_k$, which implies $r_k > \frac{(\chi^*_w(G)-\gamma_w(U_k))(|U_k|-1)}{|U_k|-1-2|E(U_k)\cap M|} = r_{k+1}$.

(ii) Let v be a vertex of G_k with $\Delta_{w_k}(G_k) = d_{G_k,w_k}(v)$. Observe that v is disjoint from M, for otherwise, $\Delta_{w_k}(G_k) = d_{G,w}(v) - r_k \leq \Delta_w(G) - r_k \leq \chi^*_w(G) - r_k$, contradicting (1). So $d_{G_{k+1},w_{k+1}}(v) = d_{G,w}(v) = d_{G_k,w_k}(v) = \Delta_{w_k}(G_k) = \chi^*_w(G) - r_{k+1}$, where the last equality can be seen from Step 2. For each vertex u covered by M, we have $d_{G_{k+1},w_{k+1}}(u) \leq \Delta_w(G) - r_{k+1} \leq \chi^*_w(G) - r_{k+1}$. So $\Delta_{w_{k+1}}(G_{k+1}) = d_{G,w}(v) = \Delta_{w_k}(G_k) = \chi^*_w(G) - r_{k+1}$. If $\Delta_{w_{k+1}}(G_{k+1}) \geq \Gamma_{w_{k+1}}(G_{k+1})$, then

$$\chi^*_{w_{k+1}}(G_{k+1}) = \max\{\Delta_{w_{k+1}}(G_{k+1}), \Gamma_{w_{k+1}}(G_{k+1})\} = \chi^*_w(G) - r_{k+1}.$$

Thus from Step 1 we see that Algorithm 5.2 terminates at iteration $k+1$ and returns $r(M) = r_{k+1}$.

(iii) By the definition of r_{k+1}, we have

(2) $\Gamma_{w_{k+1}}(G_{k+1}) \geq \frac{2w_{k+1}(U_k)}{|U_k|-1} = \frac{2w(U_k)}{|U_k|-1} - \frac{2r_{k+1}|E(U_k)\cap M|}{|U_k|-1} = \chi_w^*(G) - r_{k+1}$,
which together with (i) implies

(3) $\chi_w^*(G) - r_{k+1} \geq \frac{2w(U_k)}{|U_k|-1} - \frac{2r_k|E(U_k)\cap M|}{|U_k|-1} = \Gamma_{w_k}(G_k)$.

Let us show that

(4) $\Delta_{w_{k+1}}(G_{k+1}) \leq \Gamma_{w_{k+1}}(G_{k+1})$.

Assume the contrary: $\Delta_{w_{k+1}}(G_{k+1}) > \Gamma_{w_{k+1}}(G_{k+1})$. Then $\Delta_{w_{k+1}}(G_{k+1}) > \chi_w^*(G) - r_{k+1}$ by (2). Let v be a vertex of maximum weighted degree in G_{k+1}. If v is covered by M, then $d_{G_{k+1},w_{k+1}}(v) \leq \Delta_w(G) - r_{k+1} \leq \chi_w^*(G) - r_{k+1}$, a contradiction. So v is disjoint from M, and hence $d_{G_{k+1},w_{k+1}}(v) = d_{G,w}(v)$. Therefore

(5) $\Delta_{w_k}(G_k) \geq d_{G,w}(v) = \Delta_{w_{k+1}}(G_{k+1}) > \chi_w^*(G) - r_{k+1}$.

Combining (3) and (5), we get $\Delta_{w_k}(G_k) > \Gamma_{w_k}(G_k)$; this contradiction to the hypothesis establishes (4).

It remains to prove that

(6) $\frac{2|E(U_{k+1})\cap M|}{|U_{k+1}|-1} > \frac{2|E(U_k)\cap M|}{|U_k|-1}$.

Indeed, since U_{k+1} has been defined, by Step 1 and (4) we have

(7) $\Gamma_{w_{k+1}}(G_{k+1}) = \chi_{w_{k+1}}^*(G_{k+1}) > \chi_w^*(G) - r_{k+1}$.

As U_{k+1} is an OoS in G_{k+1} with respect to $\boldsymbol{w}_{k+1}$, there holds $\Gamma_{w_{k+1}}(G_{k+1}) = \frac{2w_{k+1}(U_{k+1})}{|U_{k+1}|-1}$. From (2) and (7) it follows that

(8) $\gamma_w(U_{k+1}) - \frac{2r_{k+1}|E(U_{k+1})\cap M|}{|U_{k+1}|-1} > \gamma_w(U_k) - \frac{2r_{k+1}|E(U_k)\cap M|}{|U_k|-1}$.

Since U_k is an OoS in G_k with respect to $\boldsymbol{w}_k$, the inequality $\frac{2w_k(U_k)}{|U_k|-1} \geq \frac{2w_k(U_{k+1})}{|U_{k+1}|-1}$ holds; that is,

(9) $\gamma_w(U_k) - \frac{2r_k|E(U_k)\cap M|}{|U_k|-1} \geq \gamma_w(U_{k+1}) - \frac{2r_k|E(U_{k+1})\cap M|}{|U_{k+1}|-1}$.

Combining (8) and (9), we obtain

(10) $(r_k - r_{k+1})\frac{2|E(U_{k+1})\cap M|}{|U_{k+1}|-1} > (r_k - r_{k+1})\frac{2|E(U_k)\cap M|}{|U_k|-1}$.

Thus (6) follows from (i) and (10). ∎

Lemma 5.3. *Suppose $G - M$ is a simple graph. Then Algorithm 5.2 correctly finds $r(M)$ in time $O(n^5\ell\log(n^2/\ell))$, where $n = n(G)$ and $\ell = \ell(G)$.*

Proof. By Lemma 5.2(ii) and (iii), we may assume that $\Delta_{w_k}(G_k) \leq \Gamma_{w_k}(G_k)$ for all $k \geq 2$. Thus the inequality $\frac{2|E(U_{k+1})\cap M|}{|U_{k+1}|-1} > \frac{2|E(U_k)\cap M|}{|U_k|-1}$ holds for all $k \geq 2$. Let $\boldsymbol{c}$ be the weight function defined on E such that $c(e) = 1$ if $e \in M$ and 0 otherwise. Applying Lemma 2.9 to $(G, \boldsymbol{c})$, we find that $\frac{2|E(U_{k+1})\cap M|}{|U_{k+1}|-1} - \frac{2|E(U_k)\cap M|}{|U_k|-1} \geq \frac{2}{(n-1)(n-2)}$ for all $k \geq 1$. Since $0 \leq \frac{2|E(U)\cap M|}{|U|-1} \leq 1$ for any odd set U, Algorithm 5.2 terminates within $O(n^2)$ iterations. From the algorithm, we see that the output is the largest constant r, with $\chi_{w(r)}^*(G - rM) = \max\{\Delta_{w(r)}(G - rM), \Gamma_{w(r)}(G - rM)\} = \chi_w^*(G) - r$, where $\boldsymbol{w}(r) = \boldsymbol{w} - rM$, and hence it is the residue of M.

Since each of Step 1 and Step 2 takes time $O(n^3\ell\log(n^2/\ell))$ by Lemma 3.4, the algorithm runs in time $O(n^5\ell\log(n^2/\ell))$. ∎

The following two lemmas can be seen directly from the definition of residue, and the first one can be proved easily by contradiction.

Lemma 5.4. *Let M be a matching of G with $0 \le r(M) < b(M)$, let $G' = G - r(M)M$, and let $\boldsymbol{w}' = \boldsymbol{w} - r(M)M$. Then one of the following statements holds:*

(i) there exists a vertex v disjoint from M, such that $\Delta_{w'}(G') = d_{G',w'}(v) = \chi^*_w(G) - r(M)$;

(ii) there exists an odd set U with $|E(U) \cap M| < \frac{|U|-1}{2}$, such that $\Gamma_{w'}(G') = \frac{2w'(U)}{|U|-1} = \chi^*_w(G) - r(M)$.

Proof. It suffices to show (ii) holds when (i) does not hold. Clearly, from the definition of $r(M)$, we have $\chi^*_{w'}(G') = \max\{\Delta_{w'}(G'), \Gamma_{w'}(G')\} = \chi^*_w(G) - r(M)$. Since (i) does not hold, for each vertex v disjoint from M, we see that $d_{G,w}(v) = d_{G',w'}(v) < \chi^*_w(G) - r(M)$. Let t be the largest value such that $d_{G,w}(v) + t \le \chi^*_w(G)$ for each vertex v disjoint from M. Write $\boldsymbol{w}(t) = \boldsymbol{w} - tM$. By the choice of t, we deduce that $t > r(M)$ and $\Delta_{w(t)}(G - tM) = \chi^*_w(G) - t$.

Assume the contrary that (ii) does not hold. Then for each odd set U with $|E(U) \cap M| < \frac{|U|-1}{2}$, we have $\frac{2w(U)}{|U|-1} - r(M)\frac{2|E(U)\cap M|}{|U|-1} = \frac{2w'(U)}{|U|-1} < \chi^*_w(G) - r(M)$. Let s be the largest value such that $\frac{2w(U)}{|U|-1} - s\frac{2|E(U)\cap M|}{|U|-1} \le \chi^*_w(G) - s$ for each odd set U with $|E(U) \cap M| < \frac{|U|-1}{2}$. Write $\boldsymbol{w}(s) = \boldsymbol{w} - sM$. From the choice of s, we see that $s > r(M)$ and $\Gamma_{w(s)}(G - sM) = \chi^*_w(G) - s$. Let $\lambda = \min\{t, s\}$ and $\boldsymbol{w}(\lambda) = \boldsymbol{w} - \lambda M$. Then $\lambda > r(M)$ and $\chi^*_{w(\lambda)}(G - \lambda M) = \max\{\Delta_{w(\lambda)}(G - \lambda M), \Gamma_{w(\lambda)}(G - \lambda M)\} = \chi^*_w(G) - \lambda$, which again contradicts the definition of $r(M)$. Therefore, when (i) does not hold, (ii) must hold. ∎

Lemma 5.5. *Let M be a feasible matching of G with residue $r(M)$, let $G' = G - r(M)M$, and let $\boldsymbol{w}' = \boldsymbol{w} - r(M)M$. Then the following statements hold:*

(i) If $\Delta_w(G) \ge \Gamma_w(G)$, then $\Delta_{w'}(G') \ge \Gamma_{w'}(G')$.

(ii) If $\Delta_w(G) = \Gamma_w(G)$, then $\Delta_{w'}(G') = \Gamma_{w'}(G')$.

(iii) If $\Delta_w(G) \le \Gamma_w(G)$, then $\Delta_{w'}(G') \le \Gamma_{w'}(G')$.

Let us now analyze the computational complexity of our algorithm for WFECP.

Lemma 5.6. *Algorithm 5.1 terminates in at most $2m + 5n$ iterations and hence runs in time $O(m^2 + n^5 m\ell \log(n^2/\ell))$, where $n = n(G)$, $m = m(G)$, and $\ell = \ell(G)$.*

Proof. Recall that in the algorithm $r_k = r(M_k)$ for all $k \ge 1$. We break the proof into some simple observations.

(1) Suppose i and j are two subscripts such that $\Delta_{w_k}(G_k) > \Gamma_{w_k}(G_k)$ for all k with $i \le k \le j$. Then $j - i + 1 \le m + n$.

To justify this, note that, for each k with $i \le k \le j-1$, if $r(M_k) = b(M_k)$, then G_{k+1} is obtained from G_k by deleting at least one edge in M_k. If $r(M_k) < b(M_k)$, then there exists a vertex v disjoint from M_k, such that $\Delta_{w_{k+1}}(G_{k+1}) = d_{G_{k+1},w_{k+1}}(v) = \chi^*_w(G) - r_k$ by Lemma 5.4. Thus the number of vertices with maximum weighted degree $\Delta_{w_{k+1}}(G_{k+1})$ in G_{k+1} is strictly greater than the number of vertices with maximum weighted degree $\Delta_{w_k}(G_k)$ in G_k if $k \le j-2$.

Since each vertex with maximum weighted degree in G_k also has maximum weighted degree in $G_{k'}$ for any pair $\{k, k'\}$ with $i \le k < k' \le j-1$, and since G has m edges in total and each G_k has at most n vertices with maximum weighted degree, we conclude that $j - i + 1 \le m + n$.

(2) Suppose i and j are two subscripts such that $\Delta_{w_k}(G_k) < \Gamma_{w_k}(G_k)$ for all k with $i \le k \le j$. Then $j - i + 1 \le m + 2n$.

To justify this, note that, for each k with $i \le k \le j-1$, if $r(M_k) = b(M_k)$, then G_{k+1} is obtained from G_k by deleting at least one edge in M_k. If $r(M_k) < b(M_k)$, then there exists an odd set U with $|E(U) \cap M_k| < \frac{|U|-1}{2}$, such that $\Gamma_{w_{k+1}}(G_{k+1}) = \frac{2w_{k+1}(U)}{|U|-1} = \chi_w^*(G) - r(M_k)$ by Lemma 5.4. Let $\mathcal{O}_k$ be a laminar family of OoS's in G_k with maximum size. Then each set in $\mathcal{O}_k \cup \{U\}$ is an OoS in G_{k+1}. Thus we can apply Algorithm 4.1 to uncross the triple $(\mathcal{O}_k, U, M)$ and generate a laminar family $\mathcal{O}_{k+1}$ of OoS's in G_{k+1} with $|\mathcal{O}_{k+1}| > |\mathcal{O}_k|$. Let a_k denote the maximum size of a laminar family of OoS's in G_k. Then $a_{k+1} > a_k$.

Since each OoS in G_k is also an OoS in $G_{k'}$ for any pair $\{k, k'\}$ with $i \le k < k' \le j-1$, and since $a_k \le 2n$ by Lemma 4.2, we obtain $j - i + 1 \le m + 2n$.

(3) Suppose i and j are two subscripts such that $\Delta_{w_k}(G_k) = \Gamma_{w_k}(G_k)$ for all k with $i \le k \le j$. Then $j - i + 1 \le m + 3n$.

Using the same arguments as employed in the proofs of (2) and (3) and using Lemma 5.4, we can prove that if $r(M_k) < b(M_k)$, then the number of vertices with maximum weighted degree $\Delta_{w_{k+1}}(G_{k+1})$ in G_{k+1} is strictly greater than the number of vertices with maximum weighted degree $\Delta_{w_k}(G_k)$ in G_k if $k \le j-2$, or $a_{k+1} > a_k$. So $j - i + 1 \le m + 3n$.

From (1)-(3) and Lemma 5.5, we conclude that the total number of iterations of Algorithm 5.1 is bounded above by the maximum possible value of $j-i+1$ in (1) or (2) plus that of $j-i+1$ in (3), which is at most $2m + 5n$.

To determine the residue r_k of M_k in G_k contained in Step 1, we may replace $G - M$ by its simplification; or equivalently, we may assume that $G - M$ is a simple graph. Thus, by Lemma 5.3, at each iteration k the residue r_k can be found in time $O(m + n^5 \ell \log(n^2/\ell))$, which dominates the complexity of each iteration (see Theorems 1.4 and 1.5). Therefore Algorithm 5.1 runs in time $O(m^2 + n^5 m \ell \log(n^2/\ell))$. ∎

It remains to prove that WFECP can actually be solved in a more efficient way by using a two-phase algorithm.

Let $G^* = (V, E^*)$ be the simplification of G and let $w^*(e)$ be the weight of each edge e of G^*. In Phase I, we apply Algorithm 5.1 to WFECP on G^* with respect to the weight function $\boldsymbol{w}^*$, obtaining an optimal solution $\{(M_i, r_i) : 1 \le i \le t\}$. By Lemma 5.6, $t = O(2\ell + 5n) = O(\ell)$ and the time taken by this phase is $O(n^5 \ell^2 \log(n^2/\ell))$.

In Phase II, our objective is to express each M_i as the "sum" of some matchings in G; that is, to find some matchings $M_{i,1}, M_{i,2}, \ldots, M_{i,s(i)}$ in G, each of which projects into M_i in G^*, and find some positive numbers $x(M_{i,1}), x(M_{i,2}), \ldots, x(M_{i,s(i)})$, such that $\sum_{j=1}^{s(i)} x(M_{i,j}) = r_i$ for $1 \le i \le t$ and that $\sum_{e \in M_{i,j}} x(M_{i,j}) = w(e)$ for each edge e of G. The algorithm is described below, where $E_G(e)$ stands for the set of all edges between u and v in G for each edge $e = uv$ of G^*.

Algorithm 5.3 for matching decomposition

Step 0. Set $i = 1$.

Step 1. If $i > t$, stop: $\{(M_{i,j}, x(M_{i,j})) : 1 \leq i \leq t,\ j = 1, 2, \ldots\}$ is as desired. Else, set $j = 1$.

Step 2. If $r_i = 0$, go to Step 4. Else, let e' be an edge in $E_G(e)$ for each $e \in M_i$, set $M_{i,j} = \{e' : e \in M_i\}$, and set $x(M_{i,j}) = \min\{r_i, b(M_{i,j})\}$.

Step 3. Replace G by $G - x(M_{i,j})M_{i,j}$ and replace $\boldsymbol{w}$ by $\boldsymbol{w} - x(M_{i,j})M_{i,j}$ (remember to delete edges with zero weight in the resulting G). Set $r_i = r_i - x(M_{i,j})$ and $j = j + 1$, return to Step 2.

Step 4. Set $i = i + 1$, return to Step 1.

We can establish the main result on fractional edge-colorings now.

Proof of Theorem 1.2. Let us look back at Algorithm 5.3. From the definition of $x(M_{i,j})$, we see that at each iteration of generating $M_{i,j}$, either M_i becomes fully decomposed (which corresponds to the case when $r_i = 0$ in Step 2) or some edge in $M_{i,j}$ gets deleted from G because its weight becomes zero. So the total number of iterations of our algorithm (and hence the total number of matchings $M_{i,j}$ used) is $O(t + m) = O(m)$. Therefore Algorithm 5.3 runs in time $O(mn)$. Combining the complexities of the two phases, we conclude that WFECP on G can be solved in time $O(mn + n^5\ell^2 \log(n^2/\ell))$. ∎

References

[1] J. Asplund and J. McDonald, On a limit of the method of Tashkinov trees for edge-coloring, *Discrete Math.* **339** (2016), 2231-2238.

[2] A. Caprara and R. Rizzi, Improving a family of approximation algorithms to edge color multigraphs, *Inform. Process. Lett.* **68** (1998), 11-15.

[3] J. Edmonds, Maximum matching and a polyhedron with 0, 1-vertices, *J. Res. Nat. Bur. Standards Sect. B* **69** (1965), 125-130.

[4] M. X. Goemans and V. S. Ramakrishnan, Minimizing submodular functions over families of sets, *Combinatorica* **15** (1995), 499-513.

[5] A. Goldberg and R. Tarjan, A new approach to the maximum-flow problem, *J. ACM* **35** (1988), 921-940.

[6] M. Goldberg, On multigraphs of almost maximal chromatic class (Russian), *Discret. Analiz.* **23** (1973), 3-7.

[7] M. Goldberg, Edge-coloring of multigraphs: recoloring technique, *J. Graph Theory* **8** (1984), 123-137.

[8] R. Gomory and T.C. Hu, Multi-terminal network flows, *J. Soc. Indust. Appl. Math.* **9** (1961), 551-570.

[9] T. Jensen and B. Toft, Unsolved graph edge coloring problems, in: *Topics in Chromatic Graph Theory* (L. Beineke and R. Wilson, eds.), pp. 327-357, Cambridge University Press, 2015.

[10] M. Grötschel, L. Lovász, A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* **1** (1981), 169-197.

[11] M. Grötschel, L. Lovász, A. Schrijver, Corrigendum to our paper: "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* **4** (1984), 291-295.

[12] I. Holyer, The NP-completeness of edge-colorings, *SIAM J. Comput.* **10** (1981), 718-720.

[13] J. R. Isbell and H. Marlow, Attrition games, *Naval Res. Logist. Quart.* **2** (1956), 71-93.

[14] W. Kennedy, *Fractional Edge and Total Coloring*, Ph.D. Thesis, McGill University, 2011.

[15] H. Kierstead, On the chromatic index of multigraphs without large triangles, *J. Combin. Theory Ser. B* **36** (1984), 156-160.

[16] A. Letchford, G. Reinelt, and D. Theis, Odd matching cut sets and b-matchings revisited, *SIAM J. Discrete Math.* **22** (1987), 1480-1487.

[17] G. Nemhauser and S. Park, A polyhedral approach to edge coloring, *Oper. Res. Lett.* **10** (1991), 315-322.

[18] M. Padberg and R. Rao, Odd minimum cutsets and b-matchings, *Math. Oper. Res.* **7** (1982), 67-80.

[19] M. Padberg and L. Wolsey, Fractional covers for forests and matchings, *Math. Programming* **29** (1984), 1-14.

[20] A. Schrijver, *Combinatorial Optimization – Polyhedra and Efficiency*, Springer-Verlag, Berlin, 2003.

[21] P. Seymour, On multi-colorings of cubic graphs, and conjectures of Fulkerson and Tutte, *Proc. London Math. Soc.* **38** (1979), 423-460.

[22] M. Stiebitz, D. Scheide, B. Toft, and L. Favrholdt, *Graph Edge Colouring: Vizing's Theorem and Goldberg's Conjecture*, John Wiley & Sons, 2012.

[23] V. Tashkinov, On an algorithm for the edge coloring of multigraphs (Russian), *Diskretn. Anal. Issled. Oper. Ser. 1* **7** (2000), 72-85.

[24] V. Vizing, On an estimate of the chromatic class of a p-graph (Russian), *Discret. Analiz.* **3** (1964), 25-30.

[25] D. West, A short proof of the Berge-Tutte formula and the Gallai-Edmonds structure theorem, *European J. Combin.* **32** (2011), 674-676.

6 Appendix: Comments on Kennedy's Algorithm for FECP

The following is taken from Kennedy's thesis [14] (see Section 8.4, pages 154 and 155).

8.4 A Combinatorial Algorithm for Fractional Edge Colouring

We remark that one can use an iterative approach to construct an optimal factional edge coloring of any multigraph in polynomial time. (To the author's knowledge, this algorithm has not previously appeared in print, though Meagher notes in his M.Sc. thesis that such an algorithm exists.) As in Kahn's proof that the Goldberg-Seymour Conjecture is asymptotically true (see Theorem 3.2), this is more complicated than edge colouring bipartite graphs since we need to worry about reducing both the maximum degree and the edge-density of any odd overfull subgraph. We modify the simple iterative approach and apply the following two reductions. We show that if there exists a subgraph H satisfying $1 < |H| < |G|$ and $\frac{2|E(H)|}{|H|-1} = \chi'_f(G)$, then we can reduce our problem to edge coloring H and G/H, the graph obtained by contracting the vertices of H into a single vertex. In doing so, we exploit the fact that $|\delta(H)| < \Delta(G)$, from which it follows that $\chi'_f(G) = \max\{\chi'_f(H), \chi'_f(G/H)\}$. If no such subgraph exists, then we can find a matching M and scalar $\epsilon > 0$ such that when we remove weight ϵ of M the fractional chromatic index drops by ϵ. We remove weight ϵ of M and repeat the procedure on the reduced graph.

In the above description, $\chi'_f(G)$ is the fractional chromatic index of G, which is $\chi^*_w(G)$ defined in our paper when $w(e) = 1$ for each edge e. The symbol $|G|$ stands for the number of vertices in G. Moreover, $\delta(H)$ is the set of all edges with precisely one end in H.

Comment 1. Following Kennedy's algorithm, we need to first determine if there exists a subgraph H of G satisfying $1 < |H| < |G|$, $\frac{2|E(H)|}{|H|-1} = \chi'_f(G)$, and $|\delta(H)| < \Delta(G)$. If yes, find such a subgraph.

Suppose H is available. In this case, $\Gamma(G) > \Delta(G)$. Kennedy's algorithm proceeds by using the fact $\chi'_f(G) = \max\{\chi'_f(H), \chi'_f(G/H)\}$. Implicitly, it is assumed here that an optimal fractional edge-coloring ϕ_1 of H can be combined with an optimal fractional edge-coloring ϕ_2 of G/H to yield an optimal fractional edge-coloring of G.

Let U stand for the vertex set of H. To ensure that ϕ_2 can be extended to an optimal fractional edge-coloring of the whole multigraph G, what we need to combine with ϕ_2 is an optimal fractional edge-coloring ϕ_3 of $G/\overline{U}$ (rather than ϕ_1). However, the problem with this approach is that the inequality $\Gamma(G/\overline{U}) \leq \Gamma(G)$ may not hold. So the reduction employed here does not seem to work.

Comment 2. When $\Gamma(G) \leq \Delta(G)$, Kennedy's algorithm proceeds by finding a matching M and scalar $\epsilon > 0$ such that when we remove weight ϵ of M the fractional chromatic index drops by ϵ.

Consider the case when $\Gamma(G) = \Delta(G)$. To ensure that this approach works, M must cover all vertices of maximum degree and saturate all optimal odd sets. However, there is no known algorithm for finding such an M efficiently. (The reader is referred to Theorem 1.5 and its proof in the current paper.) Even though such a matching M can be obtained in polynomial time, to ensure that the whole algorithm runs in polynomial time, we have to carefully choose the scalar ϵ involved in Kennedy's algorithm, which is a very technical issue. Moreover, after removing weights of matchings involved repeatedly, the original fractional edge-coloring problem (FECP) may become the weighted fractional edge-coloring problem (WFECP) with diversified edge weights.

We also wish to point out that the Padberg-Rao algorithm cannot be used directly to find a subgraph H of G satisfying $1 < |H| < |G|$ and $\frac{2|E(H)|}{|H|-1} = \chi'_f(G)$ when $\Gamma(G) = \Delta(G)$.

So it does not seem that the problems with Kennedy's algorithm can be fixed by using only slight modifications.