A polynomial time algorithm for the fractional f-density

Guantao Chen*[1] and Guoning Yu†[1]

[1]Dept. of Math and Stat, Georgia State University , Atlanta, GA 30303

Abstract

Let G be a loopless multigraph and f be a function from $V(G)$ to $\mathbb{Z}_+$ the set of positive integers. The *fractional f-density* $\mathcal{W}_f$ is defined below:

$$\mathcal{W}_f^*(G) = \max_{U \subseteq V, |U| \geq 2} \frac{|E(U)|}{\lfloor f(U)/2 \rfloor}.$$

We give a strongly polynomial-time algorithm for calculating $\mathcal{W}_f^*(G)$ in terms of the number of vertices of G. Consequently, our algorithm extends the one developed by Chen, Zang and Zhao for $f = 1$, i.e. $f(v) \equiv 1$ for every vertex $v \in V(G)$ [Densities, matchings, and fractional edge-colorings, SIAM Journal on Optimization, 29 (2019), pp. 240-261].

An *f-(edge)-coloring* of G is an assignment of a color to each edge of G such that each color appears at each vertex $v \in V(G)$ at most $f(v)$ times. The *f-chromatic index* of a graph G, denoted by $\chi'_f(G)$, is the least integer k such that G admits an f-coloring using k colors. Clearly, a proper graph edge-coloring, where $f \equiv 1$, is a special f-coloring. The f-coloring problem has much more broader applications than traditional edge-coloring such as the file transfer problem in computer networks.

Let $\Delta_f(G) = \max_{v \in V(G)} \left\lceil \frac{d(v)}{f(v)} \right\rceil$. Then, $\Delta_f(G)$ and $\lceil \mathcal{W}_f^*(G) \rceil$ provide two lower bounds for $\chi'_f(G)$. The Goldberg-Seymour Conjecture for f-coloring states that $\chi'_f(G) \leq \max\{\Delta_f(G)+1, \lceil \mathcal{W}_f^*(G) \rceil\}$, which was recently confirmed by Chen, Hao, and Yu. Combining their results, our algorithm implies that f-chromatic number can be approximated by 1 in polynomial time.

Keywords: f-coloring; fractional edge-coloring; density

*Partially supported by NSF grants DMS-1855716 and DMS-2154331 gchen@gsu.edu

†Partially supported by the University Graduate Fellowship of Georgia State University, gyu6@gsu.edu

1 Introduction

Let $G = (V, E)$ be a multigraph without any loop, and let $f : V \to \mathbb{Z}_+$ be a function which gives a positive integral label to each vertex of G. We call a spanning subgraph M of G a *degree-f subgraph* (or an *f-matching*) if $d_M(v) \leq f(v)$ for any $v \in V(M)$, where $d_M(v)$, called the degree of v in graph M, is the number of edges in M incident with v. If every vertex $v \in M$ has degree exactly $f(v)$, such subgraph M is commonly known as an f-factor. An *f-edge-coloring* (or an *f-coloring*) of graph G is an assignment of colors to each edge such that each color class is an f-matching of G. The minimum number of colors needed for an f-coloring is called the *f-chromatic index* and denoted by $\chi'_f(G)$. The *f-coloring problem* is to determinate $\chi'_f(G)$ for a given graph G. By definition, the traditional edge coloring is an f-coloring for which $f \equiv 1$, i.e., $f(v) = 1$ for every vertex $v \in V(G)$.

By considering vertices that can be incident with more than one edge, the f-coloring problem demonstrates a broader range of practical applications compared to traditional edge coloring. One such example is the file transfer problem in computer networks, where computers often need to send or receive multiple files simultaneously. For further details on this topic, we recommend referring to the works of Choi and Hakimi (1988), Coffman et al. (1985), Krawczyk et al. (1985), and Nakano and Nakano (1993) cited in [3, 4, 8, 10].

It's worth to note that the fundamental tool used in traditional edge-coloring is the concept of a *Kempe change* applied to a component of a subgraph induced by two colors, represented as an alternating path or cycle. However, when dealing with a subgraph where the degree of vertex v is bounded by $f(v)$, the structure becomes significantly more complex. As a result, studying f-coloring poses considerably greater challenges.

Let the *fractional f-maximum degree* be $\Delta^*_f(G) = \max\limits_{v\in V} \frac{d(v)}{f(v)}$. For each $U \subseteq V$, let $f(U) = \sum_{v\in U} f(v)$. Let the *fractional f-density* of G be

$$\mathcal{W}^*_f(G) = \max_{U\subseteq V, |U|\geq 2} \frac{w(U)}{\lfloor f(U)/2 \rfloor}.$$

We call $\Delta_f(G) = \lceil \Delta^*_f(G) \rceil$ and $\mathcal{W}_f(G) = \lceil \mathcal{W}^*_f(G) \rceil$ the *f-maximum degree* and *f-density* of G, respectively. Clearly, $\Delta_f(G)$ and $\mathcal{W}_f(G)$ are both lower bounds of $\chi'_f(G)$, in other words, $\chi'_f(G) \geq \max\{\Delta_f(G), \mathcal{W}_f(G)\}$.

For the upper bound, let $\mu(u,v) = \sum\limits_{e=uv} w(e)$ be the multiplicity of a pair of distinct vertices $u, v \in V$. And for any $v \in V$, we denote $\mu(v) = \max\limits_{u\in V, u\neq v} \mu(u,v)$. Without involving f-density, Hakimi and Kariv [9] proved $\Delta_f(G) \le \chi'_f(G) \le \max_{v\in V} \left\lceil \frac{d(v)+\mu(v)}{f(v)} \right\rceil$. The disadvantage of this result is the gap between the upper bound and the lower bound could be larger than any given constant since $\mu(v)$ can be very large. The Goldberg-Seymour Conjecture for f-coloring states that

$$\max\{\Delta_f(G), \mathcal{W}_f(G)\} \le \chi'_f(G) \le \max\left\{\Delta_f(G)+1, \mathcal{W}_f(G)\right\}.$$

Consequently, the f-chromatic index is completely determined or bounded between two consecutive integers. The conjecture was recently confirmed by Chen, Hao and Yu [1]. Clearly, $\Delta^*_f(G)$ can be computed in polynomial time. In this paper, we develop an algorithm showing that $\mathcal{W}_f(G)$ can be computed in polynomial time as follows. Consequently, both lower bound and upper bound of the Goldberg-Seymour Conjecture for f-coloring can be computed in polynomial time.

Theorem 1.1. *The fractional f-density $\mathcal{W}^*_f(G)$ of a multigraph G can be computed in polynomial time in terms of the numbers of vertices and the numbers of edges of G.*

When $f \equiv 1$, the *fractional density* $\mathcal{W}^*(G)$ is defined as

$$\mathcal{W}^*(G) = \max_{U\subseteq V, |U|\ge 3\, odd} \frac{2w(U)}{|U|-1}.$$

As a consequence, we have the following.

Corollary 1.2 (Chen, Zang and Zhao [2])**.** *The fractional density $\mathcal{W}^*(G)$ of a multigraph G can be computed in polynomial time in terms of the numbers of vertices and the numbers of edges of G.*

2 Algorithm Description

We shall use the classical method proposed by Isbell and Marlow [7] that the problem of linear-fractional programming as

$$\alpha(\boldsymbol{x}^*) = \max_{\boldsymbol{x}\in S} \left\{ \alpha(\boldsymbol{x}) = \frac{g(\boldsymbol{x})}{h(\boldsymbol{x})} \right\}, \tag{1}$$

where g and h are real-valued functions on a subset S of $\mathbb{R}^n$, and $h(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in S$, is closely related to the problem

$$z\left(\boldsymbol{x}^*, \alpha\right) = \max_{\boldsymbol{x} \in S}\{z(\boldsymbol{x}, \alpha) = g(\boldsymbol{x}) - \alpha h(\boldsymbol{x})\}, \tag{2}$$

where α is a real constant, in the sense that $\boldsymbol{x}^*$ solves Problem (1) if and only if $(\boldsymbol{x}^*, \alpha^*)$ solves Problem (2) for $\alpha = \alpha^* = \alpha\left(\boldsymbol{x}^*\right)$, giving the value $z\left(\boldsymbol{x}^*, \alpha^*\right) = 0$. They also proposed an iterative method for the case when both g and h are linear, which generates a sequence of solutions to the latter problem until the above optimality criterion is satisfied. When restricted to the calculation of $\mathcal{W}_f^*(G)$, S is the family of all subsets of V with at least two vertices,

$$g(U) = 2w(U), \text{ and}$$
$$h(U) = f(U) - \sigma_f(U)$$

for each $U \in S$, where $\sigma_f(U)$ is the *parity function* indicating the oddness of $f(U)$, i.e.,

$$\sigma_f(U) = \begin{cases} 1, & \text{if } f(U) \text{ is odd} \\ 0, & \text{if } f(U) \text{ is even} \end{cases}$$

Therefore, the objective function is

$$z(U, \alpha) = 2w(U) - \alpha(f(U) - \sigma_f(U)). \tag{3}$$

Our algorithm goes as follows.

Algorithm 1 The fractional f-density algorithm

Step 0. Arbitrarily take any edge $e = u_0v_0 \in E$. Set $k = 0$, $U_0 = \{u_0, v_0\}$, and $\alpha_0 = \frac{w(U_0)}{\lfloor f(U_0)/2 \rfloor}$.
Step 1. Find the solution U_{k+1} to the problem

$$z\left(U_{k+1}, \alpha_k\right) = \max_{U \in V, |U| \geq 2} z\left(U, \alpha_k\right). \tag{4}$$

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

3 Complexity Analysis

In this section, we prove that the time complexity of Algorithm 1 is a polynomial.

Theorem 3.1. *Given a multigraph $G = (V, E)$, denote $n = |V|$ and $m = |E|$. The time complexity of the fractional f-density algorithm on G is $O\left(n^3 m^2 \log\left(n^2/m\right)\right)$.*

We first prove that Algorithm1 terminates in polynomial steps. In fact, we observe the strict monotonicity of α_k, $f(U_k)$ and $w(U_k)$.

Claim 3.2. *In Algorithm 1, we have for each $k \geq 1$,*

$$\alpha_k > \alpha_{k-1}, \tag{5}$$

$$f(U_{k+1}) < f(U_k), \text{ and} \tag{6}$$

$$w(U_{k+1}) < w(U_k). \tag{7}$$

Proof. Consider the k iteration for some $k \geq 1$. We prove for (5) first. By definition, we have $\alpha_{k-1} = \frac{w(U_{k-1})}{\lfloor f(U_{k-1})/2 \rfloor}$, and so $z(U_{k-1}, \alpha_{k-1}) = 0$. Note that $z(U_k, \alpha_{k-1}) \neq 0$ since otherwise $U^* = U_k$ is the optimal solution and the algorithm stops at the $k-1$ iteration. Hence, $z(U_k, \alpha_{k-1}) > z(U_{k-1}, \alpha_{k-1}) = 0$, which in turn gives the following.

$$w(U_k) - \alpha_{k-1} \left\lfloor \frac{f(U_k)}{2} \right\rfloor > 0$$

Therefore,

$$\alpha_k = \frac{w(U_k)}{\lfloor f(U_k)/2 \rfloor} > \alpha_{k-1}.$$

Now we prove for (6). By definition, we have

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

Since U_k is the solution for (4) at the $k-1$ iteration, we have $z(U_k, \alpha_{k-1}) \geq z(U, \alpha_{k-1})$ for any $U \in V$ with $|U| \geq 2$. It follows that

$$2w(U_{k+1}) - \alpha_{k-1}(f(U_{k+1}) - \sigma_f(U_{k+1})) \leq z(U_k, \alpha_{k-1}).$$

Then,

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

Recall that $z(U_{k+1}, \alpha_k) > 0$ and $\alpha_k > \alpha_{k-1}$, we have

$$f(U_{k+1}) - \sigma_f(U_{k+1}) < f(U_k) - \sigma_f(U_k) \tag{8}$$

For the cases that $f(U_{k+1})$ and $f(U_k)$ are of the same parity, the inequality 8 clearly gives $f(U_{k+1}) < f(U_k)$. If $f(U_{k+1})$ is even and $f(U_k)$ is odd, then $f(U_{k+1}) < f(U_k)-1 < f(U_k)$, and so we reach the conclusion $f(U_k) > f(U_{k+1})$. Otherwise, $f(U_{k+1})$ is odd, $f(U_k)$ is even, so that $f(U_{k+1})-1 < f(U_k)$. Since both sides are even integers, we have $f(U_{k+1}) - 1 \leq f(U_k) - 2$, and so $f(U_{k+1}) < f(U_k)$.

Moreover, since $z(U_k, \alpha_{k-1})$ is the maximum when fixing α_{k-1} for any $k \geq 1$, we have $z(U_k, \alpha_{k-1}) \geq z(U_{k+1}, \alpha_{k-1})$, and so

$$2w(U_k) - \alpha_{k-1}(f(U_k) - \sigma_f(U_k)) \geq 2w(U_{k+1}) - \alpha_{k-1}(f(U_{k+1}) - \sigma_f(U_{k+1})).$$

From the fact that $\alpha_{k-1} > 0$ and the inequality 8, we have

$$w(U_k) - w(U_{k+1}) \geq \frac{\alpha_{k-1}}{2}[(f(U_k) - \sigma_f(U_k)) - (f(U_{k+1}) - \sigma_f(U_{k+1}))] > 0$$

Thus, $w(U_{k+1}) < w(U_k)$. $\square$

Lemma 3.3. *Algorithm 1 terminates in $O(m)$ steps.*

Proof. Let w^* be the least common multiple of all denominators as in the fraction form of each edge weight. Note that the difference of any two different sums of edge weights is at least w^*. Since $w(U_{k+1}) < w(U_k)$ for any $k \geq 1$, Algorithm 1 terminates in at most $\frac{w(G)}{w^*} = O(m)$ steps. $\square$

Since the solution for Problem (4) is not explicitly written in Algorithm 1, we first prove that Problem (4) can be solved in polynomial time. An approach of solving it is to convert it to a minimum cut problem. Here we introduce some definitions needed. Fix a simple graph $G = (V, E)$ and an edge weight function $w : E \to \mathbb{Q}$. For any two disjoint vertex sets $X, Y \subseteq V$, we denote $E[X, Y] = E_G[X, Y] = \{xy \in E : x \in X \text{ and } y \in Y\}$, i.e., the set of edges of G connecting X and Y, and $w[X, Y] = w(E[X, Y]) = \sum_{e \in E[X,Y]} w(e)$. A subset F of E is called a *cut* if $F = E[X, \overline{X}]$ for some $X \subseteq V$. Moreover, if $\emptyset \neq X \neq V$, then $E[X, \overline{X}]$ is called a nontrivial cut. Clearly, $\emptyset$ is a cut, and it is nontrivial if and only if G is disconnected. We call $w[X, \overline{X}]$ the *capacity* of a cut $E[X, \overline{X}]$. For two distinct vertices $s, t \in V$, we call $E[X, \overline{X}]$ an s-t

cut if $|\{s,t\} \cap X| = 1$. The minimum cut problem is to find an s-t cut with minimum capacity for all $s,t \in V$.

Let $|V| = n$ and $|E| = m$. Note that Menger's Theorem implies that a minimum s-t cut problem can be converted into finding a maximum flow between s and t in the weighted graph G. Let τ be the time for finding a minimum s-t cut for given $s,t \in V$. By Goldberg-Tarjan algorithm [5] for the maximum flow problem gives that $\tau = O(nm\log(n^2/m))$. Furthermore, it is shown by Gomory and Hu [6] that by constructing a Gomory-Hu tree of G, a minimum cut can be found in time $O(n\tau)$. Therefore, we have the following lemma.

Lemma 3.4. *Let $G = (V,E)$ be a simple graph with a rational weight $w(e)$ (possibly negative) on each edge $e \in E$. A minimum cut of G can be found in time $O(n^2 m\log(n^2/m))$.*

Let $T \subseteq V$ with $|T|$ even. For a vertex set $X \subseteq V$, the cut $E[X,\overline{X}]$ is called a *T-cut* if $|T \cap X|$ is odd. The minimum T-cut problem is to find a T-cut with minimum capacity. By applying Padberg-Rao algorithm [11], Chen, Zang and Zhao [2] proved that the minimum T-cut problem can be solved in polynomial time.

Lemma 3.5 (Chen, Zang and Zhao [2])**.** *Let $G = (V,E)$ be a simple graph with a rational weight $w(e)$ (possibly negative) on each edge $e \in E$, 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 c can be found in time $O\left(n^2 m\log\left(n^2/m\right)\right)$ if all weights are nonnegative and in time $O\left(n^3 m\log\left(n^2/m\right)\right)$ otherwise.*

Now, we derive the time complexity of Problem (4).

Lemma 3.6. *Problem* (4) *can be solved in time $O(n^3 m\log(n^2/m))$.*

Proof. Noticing that for each fixed α_k, solving Problem (4) is equivalent to finding an *optimal set* U_{k+1} with at least two vertices such that $-z(U,\alpha_k)$ reaches the minimum. Since $2w(U) = d(U) - w[U,\overline{U}]$ for any $U \subseteq V$, we have

$$\begin{aligned}-z(U,\alpha_k) &= w[U,\overline{U}] - d(U) + \alpha_k(f(U) - \sigma_f(U))\\ &= w[U,\overline{U}] + \sum_{v\in U}(\alpha_k f(v) - d(v)) - \alpha_k\sigma_f(U).\end{aligned}$$

Inspired by Chen, Zang and Zhao [2], we add a dummy vertex r to the weighted graph $G = (V, E)$ and an edge between r and each vertex in $v \in V$, and obtain a vertex-labeled edge-weighted graph $G' = (V', E')$ such that

- the weight of each edge $e \in E$ in G' is the same $w(e)$ as in G;
- for each vertex $v \in V$, $w(rv) = \alpha_k f(v) - d_G(v)$;
- the label of each vertex $v \in V$ is $f(v)$; and
- the label of r is $f(V) = \sum_{v \in V} f(v)$.

Since every cut of G' is of the form $E_{G'}[U, \overline{U} \cup \{r\}]$, where $U \subseteq V$. The capacity of such a cut is

$$w[U, \overline{U} \cup \{r\}] = w[U, \overline{U}] + \sum_{v \in U} (\alpha_k f(v) - d(v)).$$

We apply Lemma 3.4, by Gomory-Hu algorithm on G', regardless of the vertex labels, we can find a nonempty minimum cut $E[M, \overline{M} \cup \{r\}]$ in time $O(n^2 m \log(n^2/m))$. If $f(M)$ is odd, then for any $X \subseteq V$,

$$\begin{aligned} -z(M, \alpha_k) &= w[M, \overline{M}] + \sum_{v \in M} (\alpha_k f(v) - d(v)) - \alpha_k \sigma_f(M) \\ &= w[M, \overline{M}] + \sum_{v \in M} (\alpha_k f(v) - d(v)) - \alpha_k \\ &\leq w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)) - \alpha_k \sigma_f(X). \end{aligned}$$

We are done by letting $U_{k+1} = M$ which is an optimal set.

Suppose $f(M)$ is even. Noticing that the total label $f(V') = f(r) + \sum_{v \in V} f(v) = 2f(V)$ is even, we can apply Lemma 3.5 and find an odd cut $E[N, \overline{N} \cup \{r\}]$ of V' whose capacity is the minimum among all odd cut of V'. The time complexity of finding such odd cut is $O(n^2 m \log(n^2/m))$ when all edge weights are nonnegative, i.e., for all $v \in V$,

$$\alpha_k f(v) - d_G(v) \geq 0,$$

which is satisfied if $\alpha_k \geq \Delta_f(G)$. Otherwise, it can be found in time $O(n^3 m \log(n^2/m))$. Then, for any $X \subseteq V$ with $f(X)$ being odd,

$$w[N, \overline{N}] + \sum_{v \in N} (\alpha_k f(v) - d(v)) \leq w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)).$$

Clearly, since $[M, \overline{M} \cup \{r\}]$ has the minimum capacity among all cuts, we have

$$w[M, \overline{M}] + \sum_{v \in M} (\alpha_k f(v) - d(v)) \leq w[N, \overline{N}] + \sum_{v \in N} (\alpha_k f(v) - d(v)).$$

Moreover, if we have

$$w[M, \overline{M}] + \sum_{v \in M} (\alpha_k f(v) - d(v)) \leq w[N, \overline{N}] + \sum_{v \in N} (\alpha_k f(v) - d(v)) - \alpha_k, \tag{9}$$

then for any $X \subseteq V$ with $f(X)$ being odd,

$$\begin{aligned}
-z(M, \alpha_k) &= w[M, \overline{M}] + \sum_{v \in M} (\alpha_k f(v) - d(v)) \\
&\leq w[N, \overline{N}] + \sum_{v \in N} (\alpha_k f(v) - d(v)) - \alpha_k \\
&\leq w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)) - \alpha_k \\
&= w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)) - \alpha_k \sigma_f(X).
\end{aligned}$$

Meanwhile, for any $X \subseteq V$ with $f(X)$ being even, we also have

$$\begin{aligned}
-z(M, \alpha_k) &= w[M, \overline{M}] + \sum_{v \in M} (\alpha_k f(v) - d(v)) \\
&\leq w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)) \\
&= w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)) - \alpha_k \sigma_f(X).
\end{aligned}$$

Therefore, $U_{k+1} = M$ is still an optimal solution.

If Inequality (9) dose not hold, then for any $X \subseteq V$ with $f(X)$ even,

$$\begin{aligned}
-z(N, \alpha_k) &= w[N, \overline{N}] + \sum_{v \in N} (\alpha_k f(v) - d(v)) - \alpha_k \\
&< w[M, \overline{M}] + \sum_{v \in M} (\alpha_k f(v) - d(v)) \\
&\leq w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)) \\
&= w[X, \overline{X}] + \sum_{v \in X} (\alpha_k f(v) - d(v)) - \alpha_k \sigma_f(X).
\end{aligned}$$

And for any $X \subseteq V$ with $f(X)$ odd,

$$\begin{aligned}
-z(N,\alpha_k) &= w[N,\overline{N}] + \sum_{v\in N}(\alpha_k f(v) - d(v)) - \alpha_k \\
&\le w[X,\overline{X}] + \sum_{v\in X}(\alpha_k f(v) - d(v)) - \alpha_k \\
&= w[X,\overline{X}] + \sum_{v\in X}(\alpha_k f(v) - d(v)) - \alpha_k \sigma_f(X).
\end{aligned}$$

Thus, $U_{k+1} = N$ is an optimal solution. In conclusion, if the minimum cut of G' is odd, Step 1 in Algorithm 1 takes time $O\left(n^2 m \log\left(n^2/m\right)\right)$, otherwise, it takes time at most $O\left(n^3 m \log\left(n^2/m\right)\right)$. $\square$

References

[1] G. Chen, Y. Hao, and X. Yu, *The goldberg-seymour conjecture on edge-coloring of weighted multigraphs*, Manuscript, (2023).

[2] X. Chen, W. Zang, and Q. Zhao, *Densities, matchings, and fractional edge-colorings*, SIAM Journal on Optimization, 29 (2019), pp. 240–261.

[3] H.-A. Choi and S. L. Hakimi, *Scheduling file transfers for trees and odd cycles*, SIAM J. Comput., 16 (1987), pp. 162–168.

[4] E. G. Coffman, Jr., M. R. Garey, D. S. Johnson, and A. S. LaPaugh, *Scheduling file transfers*, SIAM J. Comput., 14 (1985), pp. 744–780.

[5] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum-flow problem*, Journal of the Association for Computing Machinery, 35 (1988), pp. 921–940.

[6] R. E. Gomory and T. C. Hu, *Multi-terminal network flows*, Journal of the Society for Industrial and Applied Mathematics, 9 (1961), pp. 551–570.

[7] J. R. Isbell and W. H. Marlow, *Attrition games*, Naval Research Logistics Quarterly, 3 (1956), pp. 71–94.

[8] H. Krawczyk and M. Kubale, *An approximation algorithm for diagnostic test scheduling in multicomputer systems*, IEEE Trans. Comput., C-34 (1985), pp. 869 – 872.

[9] S. Louis Hakimi and O. Kariv, *A generalization of edge-coloring in graphs*, Journal of Graph Theory, 10 (1986), pp. 139–154.

[10] S.-i. Nakano and T. Nishizeki, *Scheduling file transfers under port and channel constraints*, Internat. J. Found. Comput. Sci., 4 (1993), pp. 101–115.

[11] M. W. Padberg and M. R. Rao, *Odd minimum cut-sets and b-matchings*, Mathematics of Operations Research, 7 (1982), pp. 67–80.