A Scaling Algorithm for Weighted f-Factors in General Graphs

Ran Duan
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
duanran@mail.tsinghua.edu.cn

Haoqing He
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
hehq13@mails.tsinghua.edu.cn

Tianyi Zhang
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
tianyi-z16@mails.tsinghua.edu.cn

Abstract

We study the maximum weight perfect f-factor problem on any general simple graph $G = (V, E, \omega)$ with positive integral edge weights w, and $n = |V|$,$m = |E|$. When we have a function $f : V \to \mathbb{N}_+$ on vertices, a perfect f-factor is a generalized matching so that every vertex u is matched to exactly $f(u)$ different edges. The previous best results on this problem have running time $O(mf(V))$ [Gabow 2018] or $\tilde{O}(W(f(V))^{2.373}))$ [Gabow and Sankowski 2013], where W is the maximum edge weight, and $f(V) = \sum_{u\in V} f(u)$. In this paper, we present a scaling algorithm for this problem with running time $\tilde{O}(mn^{2/3} \log W)$. Previously this bound is only known for bipartite graphs [Gabow and Tarjan 1989]. The advantage is that the running time is independent of $f(V)$, and consequently it breaks the $\Omega(mn)$ barrier for large $f(V)$ even for the unweighted f-factor problem in general graphs.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Scaling Algorithm, f-Factors, General Graphs

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.41

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at `https://arxiv.org/abs/2003.07589`.

1 Introduction

Suppose we are given an undirected simple graph $G = (V, E, \omega)$ on n vertices and m edges, with positive integer edge weights $\omega : E \to \{1, 2, \cdots, W\}$. Let $f : V \to \mathbb{N}_+$ be a function that maps vertices to positive integers. An f-factor is a subset of edges $F \subseteq E$ such that $\deg_F(u) \le f(u)$ for all $u \in V$, and F is a *perfect* f-factor if $\deg_F(u) = f(u)$ for all $u \in V$. In this paper we are concerned with computing a perfect f-factor with maximum edge weights[1]

For polynomial running time algorithms, the previous best result on this problem has running time [2]$\tilde{O}(mf(V))$ [8], where conventionally $f(V) = \sum_{v\in V} f(v)$. When edge weights are small integers, a pseudo-polynomial running time of $\tilde{O}(W\,(f(V))^{2.373})$ was obtained using algebraic approaches by [9]. For unweighted graphs, one can achieve $\tilde{O}(m\sqrt{f(V)})$ running time using algorithms from [13, 6]. Faster algorithms with running time independent of $f(V)$ were obtained previously but only in bipartite graphs: [11] gave a scaling algorithm that runs in time $\tilde{O}(m^{2/3}n^{5/3} \log W)$ solving the more general min-cost unit-capacity max-flow problem,

[1] Of course, original definition of f-factors means perfect ones (e.g. in [16]), but we follow the definition from [13] which does not require perfectness for convenience.
[2] In this paper $\tilde{O}(\cdot)$ hides $\log n$ factors.

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 41; pp. 41:1–41:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and the time bound for bipartite f-factor was later improved to $\tilde{O}(m\min\{n^{2/3}, m^{1/2}\}\cdot\log W)$ in [10], and the time bound for min-cost flow was further improved to $\tilde{O}(mn^{1/2})$ and $\tilde{O}(m^{10/7}\log W)$ using algebraic approaches [15, 1]. For the maximum weight f-factor problem, if one is willing to settle for approximate solutions instead of the exact maximum, linear time algorithms can be found from [13, 2]. A closely related problem is the min-cost perfect b-matching, in which every edge can be matched multiple times. There are several classical results for b-matchings [10, 4, 7, 8].

In this paper we prove the following result, which is the first one to break the $\Omega(mn)$ barrier for perfect f-factors in general graphs.

▶ **Theorem 1.** *There is a deterministic algorithm that computes a maximum weight perfect f-factor in $\tilde{O}(mn^{2/3}\log W)$ time.*

1.1 Technical overview

Our algorithm is based on the scaling approach for maximum weight matching in general graphs that runs in time $\tilde{O}(m\sqrt{n}\log W)$ from [3] and the blocking flow method in [5, 14, 12]. Here we begin with a sketch of our idea on finding a perfect f-factor in an unweighted graph. To generalize it to weighted graphs, we will adapt the scaling algorithmic framework for maximum weight perfect matching from [3].

The algorithm for the unweighted case uses primal-dual approach for f-factors which was presented in [8]. It maintains a set of dual variables $y : V \to \mathbb{Z}$ and $z : 2^V \to \mathbb{N}$, as well as a laminar family of blossoms $\Omega \subseteq 2^V$ and a compatible f-factor F, which are initialized as $y = 0, z = 0, F = \Omega = \emptyset$. Basically, the algorithm invokes for $Cn^{2/3}$ times the Edmonds search procedure under an approximate complementary slackness constraint on F, y, z, Ω, where C is a sufficiently large constant. The key idea is that when G is a simple graph, after that we wish to prove that the total deficiency of the current f-factor F is bounded by $O(n^{2/3})$, namely $\sum_{v\in V}(f(v) - \deg_F(v)) \le O(n^{2/3})$. If this is true, then we only need extra $O(n^{2/3})$ rounds of Edmonds searches to reach a perfect f-factor.

Let F^* be an arbitrary perfect f-factor. To upper bound the total deficiency $\sum_{v\in V}(f(v) - \deg_F(v)) \le O(n^{2/3})$, we need to bound the total number of edge-disjoint augmenting walks in $F^* \oplus F$. Consider any augmenting walk which is specified by a sequence of consecutive edges $(u_1, u_2), (u_2, u_3), \cdots, (u_{2s-1}, u_{2s})$, where $(u_{2i-1}, u_{2i}) \in F^*, (u_{2i}, u_{2i+1}) \in F$, and all u_i's but u_1, u_{2s} are saturated vertices ($\deg_F(u_i) = f(u_i)$). If we start the search for y-values of all vertices equal to some positive constant, then y-values of unsaturated vertices remain equal. Since u_1, u_{2s} are both unsaturated vertices, we have $y(u_1) = y(u_{2s}) = -Cn^{2/3}$.

No blossoms. For bipartite graphs, we do not need to consider blossoms, so we can use the idea from [12, 5]. By approximate complementary slackness we know: $y(u_{2i-1}) + y(u_{2i}) \ge -2, y(u_{2i}) + y(u_{2i+1}) \le 0$. Then we have $y(u_{2i+1}) - y(u_{2i-1}) \le 2$, $y(u_{2s-1}) \ge Cn^{2/3}$. Consider the sequence of duals: $y(u_1), y(u_3), \cdots, y(u_{2s-1})$. This sequence starts with a small value $y(u_1) = -Cn^{2/3}$ but ends with a large value $y(u_{2s-1}) \ge Cn^{2/3}$, and so intuitively many of the differences $y(u_{2i+1}) - y(u_{2i-1})$ should be positive. However, given the upper bound $y(u_{2i+1}) - y(u_{2i-1}) \le 2$, we would know many differences $y(u_{2i+1}) - y(u_{2i-1})$ can only belong to a very narrow range $\{1, 2\}$. In this case, since $y(u_{2i-1}) + y(u_{2i}) \ge -2, y(u_{2i}) + y(u_{2i+1}) \le 0$, it must be $-1 - y(u_{2i+1}) \le y(u_{2i}) \le -y(u_{2i+1})$. In words, this augmenting walk contains an edge in $V_q \times V_{-q}$, where $V_x = \{|y(u) - x| \le 1 \mid u \in V\}$, $q = y(u_{2i})$.

Since there are many different such pairs $y(u_{2i-1}), y(u_{2i+1})$, intuitively we can imagine this augmenting walk contains edges in $V_q \times V_{-q}$ for $\Omega(n^{2/3})$ different integer q's. If the number of augmenting walks is $\omega(n^{2/3})$, there will be $\Omega(n^{2/3})$ different $V_q \times V_{-q}$'s intersecting

$\omega(n^{2/3})$ augmenting walks each. By the pigeon-hole principle, there exists one such q such that $|V_q \cup V_{-q}| \le O(n^{1/3})$. As G is a simple graph, the total number of edge-disjoint augmenting walks that contains an edge in $V_q \times V_{-q}$ is at most $|V_q \cup V_{-q}|^2 = O(n^{2/3})$, which comes to a contradiction.

Handling blossoms. The major difficulty for general graphs comes from the blossoms. We use the generalized blossoms introduced in [8], and utilize the blossom dissolution technique from [3], but it will become much more complicated for f-factors. To analyze the influence of blossoms, let us divide Ω into two categories: large and small. A blossom $B \in \Omega$ is large if $|B| \ge n^{1/3}$. For small blossoms, we know by definition, the total number of edges contained in any small blossoms is bounded by $n^{4/3}$. So if $F^* \oplus F$ contains $\ge Cn^{2/3}$ augmenting walks, then most augmenting walks contain $O(n^{2/3})$ small blossom edges. To restore the argument we discussed in previous paragraphs, we could safely remove those vertices incident to any edges belonging to small blossoms from the sequence $u_1, u_3, u_5, \cdots, u_{2s-1}$, and we could still work with a very long sequence of vertices that are not removed (if C is large).

As for large blossoms, we could prove that $\sum_{\text{large } B\in\Omega} z(B) \le O(n^{4/3})$. Basically, this is because the total number of root large blossoms is always bounded by $n^{2/3}$, and so each round of Edmonds search could increase this sum by at most $n^{2/3}$, and therefore the algorithm could raise $\sum_{\text{large } B\in\Omega} z(B)$ to at most $Cn^{4/3}$ during $Cn^{2/3}$ executions of Edmonds search. Once we have a good handle of the total sum $\sum_{\text{large } B\in\Omega} z(B) \le O(n^{4/3})$, we could argue that the "average influence" of large blossoms on each augmenting walk is bounded by $O(n^{2/3})$, if $F^* \oplus F$ has more than $Cn^{2/3}$ augmenting walks.

1.2 Structure of our paper

In Section 2 we define the notations and basic concepts we will use in the paper, while in Section 3 the algorithm is given. A brief analysis of the running time of the algorithm is given in Section 4. Due to page limit, many details of the proofs are omitted, which can be found in the full version of this paper.

2 Preliminaries

2.1 Notations

Our input is a weighted simple graph $G = (V, E, \omega)$ without self-loops and a function $f : V \to \mathbb{N}_+$. For $S \subseteq V$, define $f(S) = \sum_{v\in S} f(v)$, and let $\delta(S)$ and $\gamma(S)$ be sets of edges with exactly one endpoint and both endpoints in S, respectively, and $\delta(v)$ is an abbreviation for $\delta(\{v\})$. For any edge subset $F \subseteq E$, define $\delta_F(S) = \delta(S) \cap F$, $\omega(F) = \sum_{e\in F} \omega(e)$, and $\deg_F(u) = |F \cap \{(u, v) \in E\}|$. $F \subseteq E$ is called an f-factor if $\deg_F(u) \le f(u)$ for all $u \in V$. For an f-factor F, the **deficiency** of u in F is defined as $f(u) - \deg_F(u)$ and u is **saturated** by F if $f(u) - \deg_F(u) = 0$. When all vertices are saturated, F is called a perfect f-factor. Edges in F are called **matching edges**. For a graph G and a subset of vertices U, $G[U]$ denotes the subgraph of G induced by U.

2.2 Blowup graphs

Instead of running on the original graph, our algorithm will be operating on an auxiliary weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mu)$ which is called the *blowup graph.* $\mathcal{V}$ contains all original vertices in V. For each edge $e = (u, v)$ in the original graph, add two vertices e_u, e_v to $\mathcal{V}$ and add three edges $(u, e_u), (e_u, e_v), (e_v, v)$ to $\mathcal{E}$. All vertices in V are called *original* vertices, and the newly added vertices are called *auxiliary* vertices. Then assign $\mu(u, e_u) = \mu(v, e_v) = \omega(u, v)$, $\mu(e_u, e_v) = 0$ and $f(e_u) = f(e_v) = 1$.

The purpose of transferring original graph G to $\mathcal{G}$ is mainly to avoid edges contained in the sets $I(B)$ for disjoint blossoms B. (See subsection Blossom.) Note that the number of vertices in $\mathcal{G}$ is $n + 2m$, so we need to carefully analyze the running time. It is easy to see the following, whose proof is in the full version of this paper.

▶ **Lemma 2.** *Computing maximum weight perfect f-factor in G and $\mathcal{G}$ are equivalent.*

2.3 LP formulation

Computing maximum weight perfect f-factor on the blowup graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mu)$ can be expressed as a linear program [8]:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{e \in \mathcal{E}} \mu(e)x(e) \\
\text{subject to} \quad & \sum_{e \in \delta(v)} x(e) = f(v), \forall v \in \mathcal{V} \\
& \sum_{e \in \gamma(B) \cup I} x(e) \le \left\lfloor \frac{f(B) + |I|}{2} \right\rfloor, \forall B \subseteq \mathcal{V}, I \subseteq \delta(B) \\
& 0 \le x(e) \le 1, \forall e \in \mathcal{E}
\end{aligned}
$$

Its dual LP is written as the following.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in \mathcal{V}} f(v)y(v) + \sum_{B \subseteq \mathcal{V}, I \subseteq \delta(B)} \left\lfloor \frac{f(B) + |I|}{2} \right\rfloor z(B, I) + \sum_{e \in \mathcal{E}} u(e) \\
\text{subject to} \quad & yz(e) + u(e) \ge \mu(e), \forall e \in \mathcal{E} \\
& z(B, I) \ge 0, u(e) \ge 0
\end{aligned}
$$

Here $yz(u, v)$ is defined as: $yz(u, v) = y(u) + y(v) + \sum_{B, I : (u,v) \in \gamma(B) \cup I, I \subseteq \delta(B)} z(B, I)$.

2.4 Blossoms

We follow the definitions and the terminology of [8, 13] for f-factor blossoms. A blossom is specified by a tuple $(B, \mathcal{E}_B, \beta(B), \eta(B))$, where $B \subseteq \mathcal{V}$ is a subset of vertices, $\mathcal{E}_B \subseteq \mathcal{E}$ a subset of edges, $\beta(B) \in B$ a special vertex which is called the **base**, and $\eta(B)$ is either null or an edge from $\delta(\beta(B)) \cap \delta(B)$. Blossoms follow an inductive definition below.

▶ **Definition 3** (Blossom, [8, 13]). *A single vertex v forms a trivial blossom, also called a singleton. Here $B = \{v\}$, $\mathcal{E}_B = \emptyset$, $\beta(B) = v$, and $\eta(B)$ is null. Inductively, let $B_0, B_1, \cdots, B_{l-1}$ be a sequence of disjoint singletons or non-trivial blossoms. Suppose there exists a closed walk $C_B = \{e_0, e_1, \cdots, e_{l-1}\}$ starting and ending with B_0 such that $e_i \in B_i \times B_{i+1}$, $(B_l = B_0)$. The vertex set $B = \bigcup_{i=0}^{l-1} B_i$ is identified as a blossom if the following are satisfied.*

1. *Base. If B_0 is a singleton, the two edges incident to B_0 on C_B, i.e., e_0 and e_{l-1}, must both be matched or both be unmatched.*
2. *Alternation. Fix a $B_i, i \neq 0$. If B_i is a singleton, exactly one of e_{i-1} and e_i is matched. If B_i is a non-trivial blossom, $\eta(B_i) = e_{i-1}$ or e_i.*

The edge set of the blossom B is $\mathcal{E}_B = C_B \cup (\cup_{i=0}^{l-1} \mathcal{E}_{B_i})$ and its base is $\beta(B) = \beta(B_0)$. If B_0 is not a singleton, $\eta(B) = \eta(B_0)$. Otherwise, $\eta(B)$ may either be null or one edge in $\delta(B) \cap \delta(B_0)$ that is the opposite type of e_0 and e_{l-1}.

A blossom is called **root blossom** if it is not contained in any other blossom. Blossoms have two different types: **light** and **heavy**. If B_0 is a singleton, B is light/heavy if e_0 and e_{l-1} are both unmatched/matched. Otherwise, B is light/heavy if B_0 is light/heavy.

▶ **Definition 4.** *Given an f-factor F, an alternating walk on $\mathcal{G}$ is a sequence of consecutive edges $(u_1, u_2), (u_2, u_3), \cdots, (u_{l-1}, u_l)$ such that:*
- *$(u_i, u_{i+1}) \in \mathcal{E}$ are different edges $1 \le i < l$.*
- *exactly one of $(u_{i-1}, u_i), (u_i, u_{i+1})$ belongs to F, $1 < i < l$.*

This walk is called an augmenting walk if both $(u_1, u_2), (u_{l-1}, u_l) \notin F$.

When searching for an augmenting walk, a blossom behaves as a unit in the graph. These properties are formally stated by the following lemma.

▶ **Lemma 5** ([8, 13])**.** *Let v be an arbitrary vertex in B. There exists an even length alternating walk $P_0(v)$ and an odd length alternating walk $P_1(v)$ from $\beta(B)$ to v using edges in $\mathcal{E}_B$. Moreover, the terminal edge of $P_{0,1}(v)$ incident to $\beta(B)$ must have a different type than $\eta(B)$, if $\eta(B)$ is defined.*

We also introduce the notion of **maturity** of blossoms below.

▶ **Definition 6** (Mature Blossom, [8, 13])**.** *A blossom is mature with respect to an f-factor F if the following requirements are satisfied.*
1. *Every vertex $v \in B \setminus \{\beta(B)\}$ is saturated, namely $\deg_F(v) = f(v)$.*
2. *The deficiency of $\beta(B)$ is at most 1. Furthermore, if it is 1, then B must be a light blossom and $\eta(B)$ is null; otherwise, $\eta(B)$ is defined.*

Our algorithm always keeps a set Ω of mature blossoms and maintains a non-negative value $z(B)$ for each $B \in \Omega$. For each blossom B, define a set $I(B) \subseteq \delta(B)$ as $I(B) = \delta_F(B) \oplus \{\eta(B)\}$.

2.5 Augmenting path

To find augmentations, we need to work with the contraction graph $\widehat{\mathcal{G}}$ where every root blossom is contracted to a single node.

▶ **Definition 7** ([8, 13])**.** *Let F, Ω and $\widehat{\mathcal{G}}$ be an f-factor, a set of blossoms and the graph obtained by contracting every root blossom in the Ω, respectively. $\widehat{P} = \langle B_0, e_0, B_1, e_1, \cdots, B_l \rangle \in \widehat{\mathcal{G}}$ is called an augmenting path if the following requirements are satisfied.*
1. *The terminals B_0 and B_l must be unsaturated singletons or unsaturated light blossoms. If $\widehat{P}$ is a closed path ($B_0 = B_l$), B_0 must be a singleton and the deficiency of $\beta(B_0)$ is at least 2. Otherwise B_0 and B_l can be either singletons or blossoms and their deficiency must be positive.*
2. *If the terminal vertex B_0 (B_l) is a singleton, then the incident terminal edges e_0 (e_{l-1}) must be unmatched. Otherwise, they can be either matched or unmatched.*
3. *Let B_i, $0 < i < l$ be an internal singleton or blossom. If B_i is a singleton, then exactly one of e_{i-1} and e_i is matched. If B_i is a non-trivial blossom, then $\eta(B_i) = e_{i-1}$ or e_i.*

To avoid misunderstanding, we emphasize the difference between the augmenting path and the augmenting walk. First they are defined on $\widehat{\mathcal{G}}$ and $\mathcal{G}$ respectively. Second, an augmenting walk can pass through a vertex in $\mathcal{G}$ several times but an augmenting path can pass through a vertex in $\widehat{\mathcal{G}}$ (except the endpoint) only once. In the following parts, these two concepts are used in different scenarios.

Next we define a concept of the alternating path, which is weaker than the concept of the augmenting path.

▶ **Definition 8.** *Let F, Ω and $\widehat{\mathcal{G}}$ be an f-factor, a set of blossoms and the graph obtained by contracting every root blossom in the Ω. A simple path $\widehat{P} = \langle B_0, e_0, B_1, e_1, \cdots, B_l \rangle$ is called an alternating path if it satisfies the following requirements.*

1. *The terminals B_0 must be unsaturated singletons or unsaturated light blossoms.*
2. *If the terminal vertex B_0 is a singleton, then the incident terminal edges e_0 must be non-matching. Otherwise, they can be either matching or non-matching.*
3. *For each $1 \leq i < l$, if B_i is a singleton, then exactly one of e_{i-1}, e_i is matched. Otherwise, $\eta(B_i) = e_{i-1}$ or e_i.*

2.6 Complementary slackness

Throughout the algorithm, we will be maintaining an f-factor F, a set of mature blossoms Ω, dual functions $y : \mathcal{V} \to \mathbb{N}$, $z : \Omega \to \mathbb{N}_{\geq 0}$ and $yz : \mathcal{E} \to \mathbb{N}$. For an f-factor F, we define two kinds of complementary slackness: complementary slackness and approximate complementary slackness.

▶ **Definition 9** (Complementary Slackness)**.** *In the blowup graph $\mathcal{G}$, an f-factor F, duals y, z, as well as a laminar family of blossoms Ω satisfy* complementary slackness *if the following requirements hold.*

1. *Dominance. For each $e \in \mathcal{E}$, $yz(e) \geq \mu(e)$.*
2. *Tightness. For each $e \in F$, $yz(e) = \mu(e)$.*

▶ **Definition 10** (Approximate Complementary Slackness)**.** *In the blowup graph $\mathcal{G}$, an f-factor F, duals y, z, as well as a laminar family of blossoms Ω satisfy* approximate complementary slackness *if the following requirements hold.*

1. *Dominance. For each $e \in \mathcal{E}$, $yz(e) \geq \mu(e) - 2$.*
2. *Tightness. For each $e \in F$, $yz(e) \leq \mu(e)$.*

▶ **Lemma 11** ([13])**.** *Let F be a perfect f-factor associated with duals y, z and blossoms Ω, and define F^* to be a maximum weight perfect f-factor. Suppose F, Ω, y, z satisfy approximate complementary slackness, then*

$$\mu(F) \geq \mu(F^*) - f(\mathcal{V})$$

Proof. We first define $u : \mathcal{E} \to \mathbb{N}$ as

$$u(e) = \begin{cases} \mu(e) - yz(e), & \text{if } e \in F \\ 0, & \text{otherwise} \end{cases}$$

According to the approximate domination and tightness properties, we have $u(e) \geq 0$ for all $e \in \mathcal{E}$. Moreover, $yz(e) + u(e) \geq \mu(e) - 2$ for all $e \in \mathcal{E}$. This gives the following:

$$\begin{aligned}
\mu(F) &= \sum_{e \in F} (yz(e) + u(e)) = \sum_{v \in V} deg_F(v) y(v) + \sum_{B \in \Omega} |F \cap (\gamma(B) \cup I(B))| z(B) + \sum_{e \in F} u(e) \\
&= \sum_{v \in V} f(v) y(v) + \sum_{B \in \Omega} \left\lfloor \frac{f(B) + |I(B)|}{2} \right\rfloor z(B) + \sum_{e \in \mathcal{E}} u(e) \\
&\geq \sum_{v \in V} deg_{F^*}(v) y(v) + \sum_{B \in \Omega} |F^* \cap (\gamma(B) \cup I(B))| z(B) + \sum_{e \in F^*} u(e) \\
&\geq \sum_{e \in F^*} (\mu(e) - 2) \geq \mu(F^*) - f(\mathcal{V})
\end{aligned}$$

◀

2.7 Edmonds search

In this subsection, we introduce two different implementations of Edmonds search. Suppose we have an f-factor F, a set of blossoms Ω, and duals y, z satisfying some kind of slackness condition. The purpose of Edmonds search is to reduce total deficiency of F by eligible augmenting paths. We need two different notions of eligibility, namely eligibility and approximate eligibility, compatible with Definition 9 or Definition 10.

▶ **Definition 12** (Eligibility, [8]). *An edge $e \in \mathcal{E}$ is eligible if $yz(e) = \mu(e)$.*

▶ **Definition 13** (Approximate Eligibility, [13]). *An edge $e \in E$ is approximately eligible if it satisfies one of the following.*
1. *$e \in \mathcal{E}_B$ for some $B \in \Omega$.*
2. *$e \notin F$ and $yz(e) = \mu(e) - 2$.*
3. *$e \in F$ and $yz(e) = \mu(e)$.*

Let $\widehat{\mathcal{G}_{\text{elig}}}$ be the subgraph of $\widehat{\mathcal{G}}$ consisting of eligible edges. A root blossom $B' \in \Omega$ is called reachable from an unsaturated root blossom B via an alternating path in $\widehat{\mathcal{G}_{\text{elig}}}$, if there is an alternating path that starts at B and ends at B'. To find augmenting paths and blossoms in $\widehat{\mathcal{G}_{\text{elig}}}$, we start from any unsaturated node u_0 in the contraction graph $\widehat{\mathcal{G}}$ and grow a search tree $\widehat{T}$ rooted at u_0; this method was also described in [8, 13]. All nodes in $\widehat{T}$ are classified as **outer/inner**. Initially the root is outer. Next we use a DFS-like approach to build the entire $\widehat{T}$. During the process, we keep track of a tree path $\langle u_0, e_0, u_1, \cdots, e_{l-1}, u_l \rangle$ from the root, which is guaranteed to be an alternating path. According to the type of u_l, the next edge e_l and node u_{l+1} are selected by the rules below:
1. u_l is outer. If u_l is a singleton, then scan the next non-matching edge e_l and find the other endpoint u_{l+1}. If u_l is a nontrivial blossom, then scan the next edge e_l and find the other endpoint u_{l+1}.
2. u_l is inner. If u_l is a singleton, then scan the next matching edge e_l and find the other endpoint u_{l+1}. If u_l is nontrivial blossom, then assign $e_l = \eta(u_l)$ (if it was not scanned before) and find the other endpoint u_{l+1}.

After finding u_{l+1}, we try to classify it as outer or inner: if u_{l+1} is a singleton, then u_{l+1} is outer if e_l is matched; otherwise, u_{l+1} is outer if $e_l = \eta(u_{l+1})$. Issues may arise if (1) u_{l+1} was already classified by previous tree searches and there is a conflict between the new label and the old label; or (2) u_{l+1} is an unsaturated then the tree search has found a new augmenting path. In either case we can construct a new blossom or reduce the total deficiency.

In the end, when all reachable singletons or root blossoms are classified as outer or inner, let $\widehat{\mathcal{V}}_{\text{out}}$ be the set of all outer singletons or root blossoms, and let $\widehat{\mathcal{V}}_{\text{in}}$ be the set of all inner singletons or root blossoms. Define $\mathcal{V}_{\text{out}}, \mathcal{V}_{\text{in}}$ to be the set of all vertices in $\mathcal{V}$ contained in outer and inner root blossoms, respectively. Next we introduce a meta procedure that will be a basic building block, which is dual adjustment. A dual adjustment performs the following step: decrement $y(v)$ for all $v \in \mathcal{V}_{\text{out}}$, and increment $y(v)$ for all $v \in \mathcal{V}_{\text{in}}$; after that, increment by 2 all $z(B)$ for all $B \in \widehat{\mathcal{V}}_{\text{out}}$, and decrement by 2 all $z(B)$ for all $B \in \widehat{\mathcal{V}}_{\text{in}}$. This is summarized as the AdjustDuals algorithm 1.

We introduce two different implementations of Edmonds search: the EdmondsSearch algorithm 2 and the PQ-Edmonds algorithm 3, which both rely on the AdjustDuals subroutine 1. The EdmondsSearch algorithm searches from all unsaturated root blossoms, and it requires that the y-values of all unsaturated vertices have the same parity. It reserves approximate complementary slackness under the approximate eligibility, so it only needs to perform one step of augmentation before dual-adjustment:

Algorithm 1 AdjustDuals(F, Ω, y, z).

1 classify every root blossom in Ω as outer or inner;
2 let $\widehat{\mathcal{V}_{\text{out}}}/\widehat{\mathcal{V}_{\text{in}}}$ be the set of all outer/inner root blossoms in $\widehat{\mathcal{G}_{\text{elig}}}$ including singletons; let $\mathcal{V}_{\text{out}}/\mathcal{V}_{\text{in}}$ be the set of all vertices in $\mathcal{V}$ contained in outer/inner root blossoms;
3 adjust the duals y, z as follows:

$$y(v) \leftarrow y(v) - 1, v \in \mathcal{V}_{\text{out}}$$
$$y(v) \leftarrow y(v) + 1, v \in \mathcal{V}_{\text{in}}$$
$$z(B) \leftarrow z(B) + 2, \text{for non-singleton } B \in \widehat{\mathcal{V}_{\text{out}}}$$
$$z(B) \leftarrow z(B) - 2, \text{for non-singleton } B \in \widehat{\mathcal{V}_{\text{in}}}$$

Algorithm 2 EdmondsSearch(F, Ω, y, z).

```
/* Precondition:  y-values of unsaturated vertices must all be of the
   same parity                                                     */
```
1 find a maximal set $\widehat{\Psi}$ of a vertex-disjoint augmenting paths in $\widehat{\mathcal{G}_{\text{elig}}}$ and extend $\widehat{\Psi}$ to a set Ψ of vertex-disjoint augmenting walks in $\mathcal{G}_{\text{elig}}$;
2 update $F \leftarrow F \oplus \bigcup_{P \in \Psi} P$;
3 find a maximal set Ω' of mature blossoms reachable from unsaturated vertices in $\widehat{\mathcal{G}_{\text{elig}}}$;
4 update $\Omega \leftarrow \Omega \cup \Omega'$ and $\widehat{\mathcal{G}_{\text{elig}}}$;
5 run AdjustDuals(F, Ω, y, z);
6 for every matching edge (u, v) that does not satisfy the dominance condition, choose an auxiliary node u, $y(u) \leftarrow \mu(u, v) - y(v) - \sum_B z(B)$;
7 recursively remove all root blossoms whose dual values are zero;

▶ **Lemma 14** ([13]). *In the EdmondsSearch algorithm, after augmentation and blossom formation, $\widehat{\mathcal{G}_{elig}}$ does not contain any augmenting paths.*

Proof. Suppose that, after the augmentation and blossom formation, there is an augmenting path P in $\widehat{\mathcal{G}_{\text{elig}}}$. Since $\widehat{\Psi}$ is maximal, P must intersect some augmenting path $P' \in \widehat{\Psi}$ at a vertex v. However, after the augmentation and blossom formation every edge in P' will become ineligible, so the matching edge $(v, v') \in P$ is no longer in $\widehat{\mathcal{G}_{\text{elig}}}$, contradicting the fact that P consists of eligible edges. ◀

The PQ-Edmonds algorithm searches for augmenting paths only from a set U of unsaturated vertices whose y-values share the same parity, halting after finding an augmenting path from vertices in U or making D dual adjustments.

The following two lemmas describe the properties of the EdmondsSearch algorithm and the PQ-Edmonds algorithm[3] which are from [8, 13]. Their proofs are presented in the full version of this paper.

[3] In the original paper [8], their algorithm actually searches from all unsaturated root blossoms. This slack can be remedied by the following reduction. For each unsaturated vertex $v \notin U$, match v to $f(v) - \deg_F(v)$ new temporary vertices whose duals are equal to $-y(v)$ and the matching edges have zero weight. The proof is described in the full version of this paper.

Algorithm 3 PQ-Edmonds(F, Ω, y, z, U, D).

/* Precondition: $\{y(u) | u \in U\}$ must all be of the same parity */
while *less than D dual adjustments have been made so far* **do**
  if *an augmenting path P from vertices in U is found* **then**
    update $F \leftarrow F \oplus P$;
    break;
  end
  find a maximal set Ω' of mature blossoms reachable from U in $\widehat{\mathcal{G}_{\text{elig}}}$;
  update $\Omega \leftarrow \Omega \cup \Omega'$ and $\widehat{\mathcal{G}_{\text{elig}}}$;
  run AdjustDuals(F, Ω, y, z);
  recursively remove all root blossoms whose dual values are zero;
end
for every matching edge (u, v) that does not satisfy the dominance condition, choose an auxiliary node u, $y(u) \leftarrow \mu(u, v) - y(v) - \sum_B z(B)$;

▶ **Lemma 15.** *The EdmondsSearch algorithm preserves* ***approximate complementary slackness*** *under the* ***approximate eligibility*** *definition. Furthermore, one execution can be implemented in $O(m)$ time.*

▶ **Lemma 16.** *The PQ-Edmonds algorithm preserves the* ***complementary slackness*** *under the* ***eligibility*** *definition. Furthermore, one execution can be implemented in $O(m \log n)$ time. Moreover, the $y(u)$ for unsaturated vertex u not in U will not be increased during the algorithm.*

3 The Scaling Algorithm

Our algorithm follows the idea of the scaling algorithm in [3] for maximum weight perfect matching. The scaling algorithm maintains an f-factor F, a family of blossoms Ω, as well as duals y, z, and it is divided into $\lceil \log(2f(\mathcal{V})W) \rceil$ iterations. Edge weights $\mu(e)$ are rescaled to $2f(\mathcal{V})\mu(e)$ and they all have $\lceil \log(2f(\mathcal{V})W) \rceil$ bits. Throughout the algorithm we assume y always assigns integer values and z always assigns even non-negative integers. For any $B \in \Omega$, B is called a *large blossom* if $|B \cap V| \geq n^{1/3}$, i.e., the number of original vertices in B is at least $n^{1/3}$; otherwise it is deemed a *small blossom*.

Let $\bar{\mu}$ be the edge weight function that keeps track of the scaled edge weights in each iteration. Initially before the first iteration, assign $F, \Omega = \emptyset$, $y, z, \bar{\mu} = 0$. At the beginning of each iteration, define F_0 to be the f-factor from the previous iteration. Empty the matching $F \leftarrow \emptyset$, and update weights and duals as following.

$$\bar{\mu}(e) \leftarrow 2\left(\bar{\mu}(e) + \text{the next bit of } 2f(\mathcal{V})\mu(e)\right)$$
$$y(u) \leftarrow 2y(u) + 3$$
$$z(B) \leftarrow 2z(B)$$

The whole procedure is shown in the Scaling algorithm 5, involving an important subroutine: the Dissolve algorithm 4. Note that here we only provide a stretch of the proof ideas, while the whole proofs can be found in the full version of this paper.

Algorithm 4 Dissolve(B, y, z, Ω).

---

for $u \in B$ *or there exists* $v \in B$ *such that* $(u, v) \in I(B)$ **do**
    $y(u) \leftarrow y(u) + z(B)/2$;
end
$z(B) \leftarrow 0$ and remove it from Ω;

3.1 Correctness

In this subsection, we will show that the Scaling algorithm indeed returns the maximum weight perfect f-factor in G, and in the next subsection we will analyze its time complexity. Some proofs of the statements here are omitted and can be found in the full version of this paper.

Define $\zeta(B) = (e_u, e_v)$, if $(u, e_u) = \eta(B)$ and e_v is the auxiliary vertex adjacent to e_u. It is not hard to see that:

▶ **Lemma 17.** *For any blossom $B \in \Omega$ in $\mathcal{G}$, the edge $e \in \delta(B)$ has the form of (u, e_u) where $u \in B$ is an original vertex and e_u is an auxiliary vertex.*

▶ **Lemma 18.** *There are two properties right after the scaling step (Line 3-6):*
1. *For each $e \in \mathcal{E}$, $\bar{\mu}(e) \leq yz(e)$.*
2. *For each $e \in F_0$, $\bar{\mu}(e) \geq yz(e) - 6$.*

▶ **Lemma 19.** *There are two properties right after the blossom dissolution step (Line 7-15):*
1. *For each $(u, v) \notin F_0 \cup \bigcup_{i=1}^{l} \gamma(B_i) \cup I(B_i)$, $\bar{\mu}(u, v) \leq 0$.*
2. *For each $(u, v) \in \mathcal{E}$, $\bar{\mu}(u, v) \leq 2\min\{y(u), y(v)\}$.*

Proof. Let $\bar{\mu}_1, y_1, z_1, \Omega_1$ be the edge weights, duals and blossoms at the beginning of the step of blossom dissolution, respectively. Let $\Omega'_1 \subseteq \Omega_1$ be the set of all blossoms that are dissolved in Line 7-9 before the step of reweighting. By Lemma 18: (Note that for auxiliary edges adjacent to edges in $I(B)$, blossom dissolution can only cause $\bar{\mu}(u, v)$ to become smaller.)

$$\begin{aligned}
\bar{\mu}(u, v) &\leq \bar{\mu}_1(u, v) - y_1(u) - y_1(v) - \sum_{\substack{B \in \Omega'_1 \\ (u,v) \in \gamma(B) \cup I(B)}} z_1(B) \\
&\leq yz_1(u, v) - y_1(u) - y_1(v) - \sum_{\substack{B \in \Omega'_1 \\ (u,v) \in \gamma(B) \cup I(B)}} z_1(B) \\
&= \sum_{\substack{B \in \Omega_1 \setminus \Omega'_1 \\ (u,v) \in \gamma(B) \cup I(B)}} z_1(B)
\end{aligned}$$

The last term is zero when $(u, v) \notin \bigcup_{i=1}^{l} \gamma(B_i) \cup I(B_i)$.

Hence, by the end of the step of blossom dissolution,

$$y(u) = \frac{1}{2} \sum_{\substack{B \in \Omega_1 \setminus \Omega'_1 \\ \exists (u,w) \in \gamma(B) \cup I(B)}} z_1(B) \geq \frac{1}{2} \sum_{\substack{B \in \Omega_1 \setminus \Omega'_1 \\ (u,v) \in \gamma(B) \cup I(B)}} z_1(B) \geq \frac{1}{2} \bar{\mu}(u, v)$$

By symmetry, we can also prove $y(v) \geq \frac{1}{2}\bar{\mu}(u, v)$. Then $\bar{\mu}(u, v) \leq 2\min\{y(u), y(v)\}$. ◀

Next we study what happens during the step of augmentation within small blossoms. If a matching edge (u, e_u) is newly added to F in line 17, for some small blossom B_i, $(u, e_u) \in I_{F_0}(B_i) \setminus \{\eta(B_i)\}$. The following statements will be proved in the full version.

Algorithm 5 Scaling($\mathcal{V}, \mathcal{E}, \mu, f$).

1 $y, z \leftarrow 0,\ F, \Omega \leftarrow \emptyset$;
2 **for** *iter* $= 1, \cdots, \lceil \log(2f(\mathcal{V})W) \rceil$ **do**
   `/* scaling */`
3   $\bar{\mu}(e) \leftarrow 2\,(\bar{\mu}(e) +$ the next bit of $2f(\mathcal{V})\mu(e))$;
4   $y(u) \leftarrow 2y(u) + 3$;
5   $z(B) \leftarrow 2z(B)$;
6   $F_0 \leftarrow F,\ F \leftarrow \emptyset$;
   `/* blossom dissolution (Line 7-15) */`
7   **while** *exists a large blossom* $B \in \Omega$*, or a root blossom* B *with* $z(B) \leq 12$ **do**
8   run Dissolve(B, y, z, Ω);
9   **end**
10   $\bar{\mu}(u, v) \leftarrow \bar{\mu}(u, v) - y(u) - y(v), \forall (u, v) \in \mathcal{E}$;
11   $y(u) \leftarrow 0, \forall u \in \mathcal{V}$;
12   let $B_1, B_2, \cdots, B_l$ be all the root small blossoms not dissolved yet;
13   **while** *exists a blossom* $B \in \Omega$ **do**
14   run Dissolve(B, y, z, Ω);
15   **end**
   `/*` Ω `now becomes empty. */`
   `/* augmentation within small blossoms (Line 16-27) */`
16   **if** $(u, v) \in I_{F_0}(B_j) \setminus \{\eta(B_j)\}$ *for some previous root small blossom* B_j **then**
17     $F \leftarrow F \cup \{(u, v)\}$;
18   if $u \in B_j, v \notin B_j$, $y(v) \leftarrow \bar{\mu}(u, v) - y(u)$;
19   **end**
20   **for** $i = 1, 2, \cdots, l$ **do**
21   **while** $\max\{y(u) \mid \deg_F(u) < f(u), u \in B_i\} > 6$ **do**
22   let Y_1, Y_2 be the largest and second largest y values of unsaturated vertices in B_i;
23   define $U \subseteq B_i$ to be the set of unsaturated vertices whose y values equal to Y_1;
24   define $H_i = \mathcal{G}[B_i \cup$ all the endpoints of $I_{F_0}(B_i) \setminus \{\eta(B_i)\}]$;
25   run PQ-Edmonds$(F, \Omega, y, z, U, Y_1 - Y_2)$ in subgraph H_i;
26   **end**
27   **end**
   `/* deficiency reduction */`
28   run EdmondsSearch(F, Ω, y, z) on the entire graph G for $\lceil Cn^{2/3} \rceil + 6$ times;
29 **end**
`/* weight adjustment */`
30 **for** *an edge* $(u, v) \in \mathcal{E}$ *such that* $yz(u, v) < \mu(u, v)$ **do**
31   $\mu(u, v) \leftarrow yz(u, v)$;
32 **end**
`/* PQ-deficiency reduction */`
33 repeat PQ-Edmonds$(F, \Omega, y, z, \{u \mid u \in \mathcal{V}$ is unsaturated$\}, \infty)$ on the entire graph G until the total deficiency becomes zero;

- At the beginning of the step of augmentation within small blossoms, $\bar{\mu}(u, e_u) \geq y(u) + y(e_u) - 6$.
- After we have added (u, e_u) to F and reassigned $y(e_u) \leftarrow \bar{\mu}(u, e_u) - y(u)$, the complementary slackness is preserved. Plus, $y(e_u) \geq \frac{1}{2}\bar{\mu}(u, e_u) - 6$.
- Within the while-loop, for any $u \in B_i$ reachable via alternating paths from U, $y(u) \geq Y_1$. Plus, $y(e_u) \geq 0$ at any moment. Also from Lemma 16, the y-values of unsaturated vertices outside current U cannot increase.

Then we can conclude the correctness of the algorithm:

▶ **Lemma 20.** *The* Scaling *algorithm 5 returns a maximum weight perfect f-factor in G.*

Proof. First we claim that approximate complementary slackness is maintained until the step of weight adjustment (Line 30-32). By Lemma 19, the tightness of complementary slackness is satisfied after the step of blossom dissolution. For each edge e newly added to F, $yz(e) = \mu(e)$ and the dominance of complementary slackness is satisfied. For the edges in H_i, the PQ-Edmonds preserve complementary slackness by Lemma 16. For the edge (u, v) where u and v does not belong to any H_i, the duals does not change. For the edge (u, v) where $u \in H_i$ and v does not belong to any H_j, we have $\mu(u, v) \leq 0$, $\mu(u, v) \leq 2y(v)$ and $y(u) \geq 0$ by Lemma 19. The complementary slackness is maintained after the step of augmentation within small blossoms. Since complementary slackness is stronger than approximate complementary slackness and the EdmondsSearch algorithm preserves approximate complementary slackness by Lemma 15, approximate complementary slackness is maintained at the end of each iteration.

Let μ, μ' be the edge weights respectively before and after the step of weight adjustment on line 30-32. As y, z, μ, Ω satisfy approximate complementary slackness, y, z, μ', Ω satisfy complementary slackness, we know for each edge e, $\mu(e) - \mu'(e) \in [0, 2]$. Since PQ-Edmonds algorithm preserves complementary slackness by Lemma 16, complementary slackness is maintained with respect to edge weights μ' after Algorithm 5. Now, again by $\mu(e) - \mu'(e) \in [0, 2], \forall e \in \mathcal{E}$, we know, y, z, F, Ω still satisfy approximate complementary slackness with respect to μ after Algorithm 5 is completed.

After the step of PQ-deficiency reduction, the total deficiency becomes zero. Then, according to Lemma 11, $\mu(F^*) - \mu(F) \leq f(\mathcal{V})$. Since for every edge $e \in \mathcal{E}$, $\mu(e)$ is an integral multiple of $2f(\mathcal{V})$, therefore it must be $\mu(F) = \mu(F^*)$. Hence F is a maximum weight perfect f-factor of $\mathcal{G}$. ◀

4 Running Time Analysis

Recall that an alternating walk on $\mathcal{G}$ is a sequence of edges $(u_1, u_2), (u_2, u_3), \cdots, (u_{l-1}, u_l)$ such that: (1) $(u_i, u_{i+1}) \in \mathcal{E}$ are different edges; (2) exactly one of $(u_{i-1}, u_i), (u_i, u_{i+1})$ belongs to F, $1 < i < l$. And this walk is called an augmenting walk if both $(u_1, u_2), (u_{l-1}, u_l) \notin F$.

▶ **Lemma 21.** *The running time of each iteration is $\tilde{O}(mn^{2/3})$, thus $\tilde{O}(mn^{2/3} \log W)$ for all iterations.*

Proof. We analyze the running time of the t-th iteration, where $t \geq 1$. Clearly the scaling step and the blossom dissolution step only take linear time. By Lemma 15, the deficiency reduction step takes $\tilde{O}(mn^{2/3})$ in total. So the only technical part is the running time of the augmentations within small blossoms.

For each small blossom B_i, $|B_i \cap V|$, i.e. the number of original vertices in B_i, is less than $n^{1/3}$. According to the properties of the blowup graph, the number of vertices in B_i is $O(n^{2/3})$. When we add edges in $I_{F_0}(B_i) \setminus \{\eta(B_i)\}$ to F, the overall deficiency of vertices in

B_i is at most $1 + 3\binom{|B_i \cap V|}{2} < 1.5n^{2/3}$. After one execution of the PQ-Edmonds algorithm, the overall deficiency is reduced by one, or the largest y value of unsaturated vertices in B_i is equal to Y_2; the former case could happen at most $1.5n^{2/3}$ times, while the latter case could happen at most $|B_i| = O(n^{2/3})$ times since every time this case happens we add at least one more unsaturated vertex to U. Thus the PQ-Edmonds algorithm is invoked for at most $O(n^{2/3})$ times. By Lemma 16, for each small blossom B_i, each instance of the PQ-Edmonds algorithm takes $O(m(B_i) \log n)$ time, where $m(B_i)$ denotes the number of edges in B_i. Thus, the total running time of the augmentations within small blossoms is $\tilde{O}(mn^{2/3})$. ◀

Now we only need to analyze the running time for the PQ-deficiency reduction step at the end of the Scaling algorithm 5.

▶ **Lemma 22.** *Let F_t denote the f-factor at the end of the t-th scaling iteration. For any $t \geq 1$, $F_{t-1} \oplus F_t$ contains at most $O(n^{2/3})$ edge-disjoint augmenting walks in $\mathcal{G}$, where augmenting walks are w.r.t. F_t. (For the F_1, we can imagine an arbitrary perfect f-factor F_0, but do not need to compute it explicitly.)*

Then we can see the total deficiency of F_t is at most $O(n^{2/3}t)$ for any $t \geq 1$, and the overall running time of our algorithm is bounded by $\tilde{O}(mn^{2/3} \log W)$ by Lemma 16.

The rest of this subsection is devoted to the proof of Lemma 22 in the t-th iteration. With a slight abuse of notations, let $F_0 = F_{t-1}$ and $F = F_t$ and when talking about augmenting walks, we always mean augmenting walks in $F_0 \oplus F$ w.r.t. F. (F-edges are considered as matching edges and F_0-edges are considered as non-matching edges.) Let $\bar{\mu}_{\text{old}}, y_{\text{old}}, z_{\text{old}}, \Omega_{\text{old}}$ denote the edge weights, duals, and blossoms at the beginning of the blossom dissolution step, respectively; and let $\Omega_{\text{old}}^{\text{large}}$ be the set of all blossoms in Ω_{old} that were dissolved in the blossom dissolution phase before the reweighting step. Similarly, $\bar{\mu}, y, z, \Omega$ denote the edge weights, duals, and blossoms at the end of the t-th iteration, respectively; and Ω^{large} denotes the set of all large blossoms in Ω.

Instead of directly working with duals y, define variables $\widehat{y}$ for vertices as follows:

$$\widehat{y}(u) = y(u) + \frac{1}{2} \sum_{\substack{X \in \Omega^{\text{large}} \text{ s.t.} \\ \exists (u,v) \in \gamma(X) \cup I(X)}} z(X)$$

When $\eta(B) \in I(B)$, $\eta(B)$ is not a matching edge. Then $\zeta(B)$ may be a matching edge and its yz-value will increase by $z(B)/2$ after the dissolution of blossom B. (Recall that $\zeta(B) = (e_u, e_v)$, if $(u, e_u) = \eta(B)$.)

Consider any subwalk $\rho = \langle u_1, u_2, \cdots, u_{2s+1} \rangle$ of an augmenting walk in $F_0 \oplus F$ starting with an edge not in F. Then, for $1 \leq i \leq s$, since $(u_{2i-1}, u_{2i}) \notin F$ and $(u_{2i}, u_{2i+1}) \in F$, by approximate complementary slackness we have

$$y(u_{2i-1}) + y(u_{2i}) + \sum_{\substack{X \in \Omega \\ (u_{2i-1}, u_{2i}) \in \gamma(X) \cup I(X)}} z(X) \quad \geq \quad \bar{\mu}(u_{2i-1}, u_{2i}) - 2 \tag{1}$$

$$y(u_{2i}) + y(u_{2i+1}) + \sum_{\substack{X \in \Omega \\ (u_{2i}, u_{2i+1}) \in \gamma(X) \cup I(X)}} z(X) \quad \leq \quad \bar{\mu}(u_{2i}, u_{2i+1}) \tag{2}$$

Plugging in the definition of $\widehat{y}$, we get:

$$\widehat{y}(u_{2i-1}) + \widehat{y}(u_{2i}) + \sum_{\substack{X \in \Omega \setminus \Omega^{\text{large}} \\ (u_{2i-1}, u_{2i}) \in \gamma(X) \cup I(X)}} z(X) \geq \bar{\mu}(u_{2i-1}, u_{2i}) - 2 \tag{3}$$

and since $(u_{2i}, u_{2i+1}) \in F$, we have:

$$\widehat{y}(u_{2i}) + \widehat{y}(u_{2i+1}) = y(u_{2i}) + y(u_{2i+1}) + \sum_{\substack{X \in \Omega^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \gamma(X) \cup I(X)}} z(X) + \frac{1}{2} \sum_{\substack{X \in \Omega^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \{\eta(X),\zeta(X)\}}} z(X)$$

$$\widehat{y}(u_{2i}) + \widehat{y}(u_{2i+1}) + \sum_{\substack{X \in \Omega \setminus \Omega^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \gamma(X) \cup I(X)}} z(X) \le \bar{\mu}(u_{2i}, u_{2i+1}) + \frac{1}{2} \sum_{\substack{X \in \Omega^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \{\eta(X),\zeta(X)\}}} z(X)$$

Taking a subtraction we have

$$\begin{aligned}\widehat{y}(u_{2i+1}) - \widehat{y}(u_{2i-1}) \le 2 + \bar{\mu}(u_{2i}, u_{2i+1}) - \bar{\mu}(u_{2i-1}, u_{2i}) + \frac{1}{2} \sum_{\substack{X \in \Omega^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \{\eta(X),\zeta(X)\}}} z(X) \\ + \sum_{\substack{X \in \Omega \setminus \Omega^{\text{large}} \\ (u_{2i-1},u_{2i}) \in \gamma(X) \cup I(X)}} z(X) - \sum_{\substack{X \in \Omega \setminus \Omega^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \gamma(X) \cup I(X)}} z(X)\end{aligned} \tag{4}$$

By the derivation presented in Lemma 19,

$$\begin{aligned}\bar{\mu}(u_{2i}, u_{2i+1}) &\le yz_{\text{old}}(u_{2i}, u_{2i+1}) - y_{\text{old}}(u_{2i}) - y_{\text{old}}(u_{2i+1}) - \sum_{\substack{X \in \Omega_{\text{old}}^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \gamma(X) \cup I(X)}} z_{\text{old}}(X) \\ &= \sum_{\substack{X \in \Omega_{\text{old}} \setminus \Omega_{\text{old}}^{\text{large}} \\ (u_{2i},u_{2i+1}) \in \gamma(X) \cup I(X)}} z_{\text{old}}(X)\end{aligned} \tag{5}$$

Now, as $(u_{2i-1}, u_{2i}) \in F_0$, again by Lemma 18 we are able to prove:

$$\bar{\mu}(u_{2i-1}, u_{2i}) \ge -6 + \sum_{\substack{X \in \Omega_{\text{old}} \setminus \Omega_{\text{old}}^{\text{large}} \\ (u_{2i-1},u_{2i}) \in \gamma(X) \cup I(X)}} z_{\text{old}}(X) - \frac{1}{2} \sum_{\substack{X \in \Omega_{\text{old}}^{\text{large}} \\ (u_{2i-1},u_{2i}) \in \{\eta(X),\zeta(X)\}}} z_{\text{old}}(X) \tag{6}$$

So for (u_{2i-1}, u_{2i}) not an η-edge or ζ-edge for old large blossoms,

$$\bar{\mu}(u_{2i-1}, u_{2i}) \ge -6 \tag{7}$$

Also define the function Z for an augmenting walk ρ as:

$$\mathrm{Z}(\rho) = \frac{1}{2} \sum_{\substack{e \in \rho, X \in \Omega \\ e \in \{\eta(X),\zeta(X)\}}} z(X) + \frac{1}{2} \sum_{\substack{e \in \rho, X \in \Omega_{\text{old}} \\ e \in \{\eta(X),\zeta(X)\}}} z_{\text{old}}(X)$$

Let $\widehat{F}, \widehat{\Omega}, \widehat{z}$ denote any f-factor together with a compatible set of blossoms as well as their duals, and let ρ be an arbitrary alternating walk. For any blossom $X \in \widehat{\Omega}$, define the following quantity:

$$\mathrm{Diff}(\rho, X, \widehat{F}) \stackrel{\text{def}}{=} |\rho \cap \widehat{F} \cap (\gamma(X) \cup I(X))| - |\rho \cap (\gamma(X) \cup I(X)) \setminus \widehat{F}|$$

By a summation of (4) plugging in (5) and (6) over all $1 \le i \le s$,

$$\begin{aligned}\widehat{y}(u_{2s+1}) - \widehat{y}(u_1) \le& 8s + \mathrm{Z}(\rho) - \sum_{X \in \Omega \setminus \Omega^{\text{large}}} z(X) \cdot \mathrm{Diff}(\rho, X, F) \\ &- \sum_{X \in \Omega_{\text{old}} \setminus \Omega_{\text{old}}^{\text{large}}} z_{\text{old}}(X) \cdot \mathrm{Diff}(\rho, X, F_0)\end{aligned} \tag{8}$$

We consider augmenting walks $\rho = \langle u_1, u_2, \cdots, u_{2s}\rangle$ in $F_0 \oplus F$ starting and ending with edges not equal to $\eta(X)$ or $\zeta(X)$ for all $X \in \Omega^{\text{large}} \cup \Omega^{\text{large}}_{\text{old}}$, and u_1, u_{2s} are not equal to $\beta(X)$ for all $X \in \Omega^{\text{large}} \cup \Omega^{\text{large}}_{\text{old}}$. This only excludes $O(n^{2/3})$ augmenting walks. Thus, $\widehat{y}(u_1) = \widehat{y}(u_{2s}) = -Cn^{2/3}$. For convenience, we only consider augmenting walks starting and ending with non-matching edges ($\notin F$) not in $\gamma(X) \cup I(X)$ for current small blossoms X, otherwise we can just choose the first and last such edges on every augmenting walk and consider the subwalk between them, and we can get similar results.

In an augmenting walk satisfying those conditions, since its first and last edges are in F_0 and not equal to η or ζ edges of old large blossoms, $\bar{\mu}(u_{2s-1}, u_{2s}) \geq -6$, and also (u_{2s-1}, u_{2s}) cannot be in a large blossom, by (1) we have $\widehat{y}(u_{2s-1}) \geq -\widehat{y}(u_{2s}) - 8 = Cn^{2/3} - 8$. Then by (8), if $\mathrm{Z}(\rho)$ is less than $n^{2/3}$, and $\mathrm{Diff}(\rho, X, F)$ and $\mathrm{Diff}(\rho, X, F_0)$ are non-negative, then we can see the length of such an augmenting walk is $\Omega(n^{2/3})$ when C is a large constant. We have the following statement for $\mathrm{Diff}(\rho, X, F)$ and $\mathrm{Diff}(\rho, X, F_0)$, which is proven in the full version.

▶ **Lemma 23.** *Consider any blossom $X \in \Omega \cup \Omega_{old}$ and any augmenting walk ρ in $F_0 \oplus F$, in any maximal consecutive subwalk of $\rho \cap (\gamma(X) \cup I(X))$, the number of matching edges is at least the number of non-matching edges. (For $X \in \Omega$, edges in F are matching edges, and for $X \in \Omega_{old}$, edges in F_0 are matching edges.)*

So $\mathrm{Diff}(\rho, X, F)$ and $\mathrm{Diff}(\rho, X, F_0)$ are both nonnegative. It is also not hard to see the following observations:

- For old and new large blossoms, $\sum_{B \in \Omega^{\text{large}}_{\text{old}}} z_{\text{old}}(B)$ and $\sum_{B \in \Omega^{\text{large}}} z(B)$ are both $O(n^{4/3})$. This is because the number of root large blossoms at a given time are bounded by $O(n^{2/3})$ and large blossoms can only be formed and dual-adjusted in the $O(n^{2/3})$ EdmondsSearch steps in the deficiency reduction step.
- The total number of non-matching edges in $\gamma(B) \cup I(B)$ for all small blossoms B is bounded by $O(n^{4/3})$, that is, $\sum_{B \in \Omega \setminus \Omega^{\text{large}}} |(\gamma(B) \cup I(B)) \setminus F|$ and $\sum_{B \in \Omega_{\text{old}} \setminus \Omega^{\text{large}}_{\text{old}}} |(\gamma(B) \cup I(B)) \setminus F_0|$ are both $O(n^{4/3})$. This is because the number of edges in $\gamma(B)$ for every blossom B is bounded by $O(|B|^2)$, the size of root small blossoms is less than $n^{1/3}$, and every root blossom has at most one non-matching edge in $I(B) \setminus \gamma(B)$,

Therefore, if we assume the number of augmenting walks in $F_0 \oplus F$ is larger than $K \cdot n^{2/3}$ for a large constant K, then we still have $\Omega(n^{2/3})$ augmenting walks ρ satisfying the following conditions:

(a) Starting and ending with edges not equal to η-edge or ζ-edge for all old large blossoms.
(b) Starting and ending with vertices not equal to the base for all old large blossoms.
(c) $\mathrm{Z}(\rho) < n^{2/3}$
(d) The total number of non-matching edges on ρ in $\gamma(X) \cup I(X)$ for all old and new small blossoms X is less than $n^{2/3}$. As in Lemma 23, the number of maximal consecutive subwalks in $\rho \cap (\gamma(X) \cup I(X))$ containing those edges is also bounded by $n^{2/3}$. We call all the edges in such subwalks "skip edges".

Among vertices $u_1, u_2, \cdots, u_{2s-1}$, we pick the vertices with odd subscript which are original vertices in G, plus the two endpoints, and obtain the list: $u_1 (= v_1), u_p (= v_2), u_{p+6} (= v_3), \cdots, u_{p+6q} (= v_{q+2}), u_{2s-1} (= v_{q+3})$, where p is 3,5 or 7. Consider all the differences $\widehat{y}(v_{i+1}) - \widehat{y}(v_i)$ for $i = 1, \cdots, q+2$. For the subwalk $[v_i, v_{i+1}]$ in ρ containing skip edges or the η or ζ-edges of old or new large blossoms, the sum of the differences $\widehat{y}(v_{i+1}) - \widehat{y}(v_i)$ is at most $O(n^{2/3})$, by (c),(d) and Lemma 23. For subwalk $[v_i, v_{i+1}]$ in ρ which does not contain skip edges or η or ζ-edges, from (4), $\widehat{y}(u_{2j+1}) - \widehat{y}(u_{2j-1}) \leq 8$ for (u_{2j+1}, u_{2j-1}) in the subwalk, so $\widehat{y}(v_{i+1}) - \widehat{y}(v_i) \leq 24$, Since the sum of all $\widehat{y}(v_{i+1}) - \widehat{y}(v_i)$ is $\widehat{y}(u_{2s-1}) - \widehat{y}(u_1)$, which is at least $2Cn^{2/3}$, if C is a large constant, the number of subwalks $[v_i, v_{i+1}]$ in ρ which does not contain skip edges or η, ζ-edges and satisfies $0 < \widehat{y}(v_{i+1}) - \widehat{y}(v_i) \leq 24$ is $\Omega(n^{2/3})$. Moreover, there are $\Omega(n^{2/3})$ different values of $\widehat{y}(v_i)$ in ρ. When v_i is not u_1 and v_{i+1} is not u_{2s-1}, then $v_i = u_{p+6k}$ and $v_{i+1} = u_{p+6k+6}$. Then we can see $\widehat{y}(u_{p+6k+2}) - \widehat{y}(u_{p+6k})$, $\widehat{y}(u_{p+6k+4}) - \widehat{y}(u_{p+6k+2})$ and $\widehat{y}(u_{p+6k+6}) - \widehat{y}(u_{p+6k+4})$ are at most 8, and also by (1),(2),(7) and Lemma 19, (remember p is odd)

$$\widehat{y}(u_{p+6k+3}) + \widehat{y}(u_{p+6k+4}) \leq 0$$
$$\widehat{y}(u_{p+6k+2}) + \widehat{y}(u_{p+6k+3}) \geq -8$$

So we have $\widehat{y}(u_{p+6k+3}) \geq -\widehat{y}(u_{p+6k}) - 16$ and $\widehat{y}(u_{p+6k+3}) \leq -\widehat{y}(u_{p+6k+6}) + 8 < -\widehat{y}(u_{p+6k}) + 8$. Note that u_{p+6k} and u_{p+6k+3} are original vertices in G, and their $\widehat{y}$-values are $\geq -Cn^{2/3}$, so their $\widehat{y}$-values are also $< Cn^{2/3} + 8$.

Given an interval $[a, b]$ of integers, define $V_{[a,b]} = \{v \in V | \widehat{y}(v) \in [a, b]\}$. For an edge $(u, v) \in E$ such that $u \in V_{[a,b]}$ and $v \in V_{[a',b']}$, we say the auxiliary edges $(u, e_u), (e_u, e_v), (e_v, v)$ are "between" the pair of intervals $[a, b]$ and $[a', b']$. If we divide the original vertices in G by their $\widehat{y}$-values into intervals of length 48, then every augmenting walk we consider will go through auxiliary edges between $\Omega(n^{2/3})$ pairs of intervals of the form $[a, a+48], [-a, -a+48]$. Any constant fraction of those $\Theta(n^{2/3})$ pairs of intervals contains $O(n^{1/3})$ vertices each, so there are at most $O(n^{2/3})$ auxiliary edges between any of such pair of intervals. Thus the number of augmenting walks satisfying (a),(b),(c),(d) is bounded by $O(n^{2/3})$.

References

1 Michael B Cohen, Aleksander Madry, Piotr Sankowski, and Adrian Vladu. Negative-weight shortest paths and unit capacity minimum cost flow in $\tilde{O}(m^{10/7} \log w)$. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 752–771. SIAM, 2017.

2 Ran Duan and Seth Pettie. Linear-time approximation for maximum weight matching. *Journal of the ACM (JACM)*, 61(1):1, 2014.

3 Ran Duan, Seth Pettie, and Hsin-Hao Su. Scaling algorithms for weighted matching in general graphs. *ACM Transactions on Algorithms (TALG)*, 14(1):8, 2018.

4 Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

5 Shimon Even and R Endre Tarjan. Network flow and testing graph connectivity. *SIAM journal on computing*, 4(4):507–518, 1975.

6 Harold N Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 448–456. ACM, 1983.

7 Harold N Gabow. Scaling algorithms for network problems. In *24th Annual Symposium on Foundations of Computer Science (SFCS 1983)*, pages 248–258. IEEE, 1983.

8 Harold N Gabow. Data structures for weighted matching and extensions to b-matching and f-factors. *ACM Transactions on Algorithms (TALG)*, 14(3):39, 2018.

9 Harold N Gabow and Piotr Sankowski. Algebraic algorithms for b-matching, shortest undirected paths, and f-factors. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 137–146. IEEE, 2013.

10 Harold N Gabow and Robert E Tarjan. Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18(5):1013–1036, 1989.

11 Andrew Goldberg and Robert Tarjan. Solving minimum-cost flow problems by successive approximation. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 7–18. ACM, 1987.

12 Andrew V Goldberg and Satish Rao. Beyond the flow decomposition barrier. *Journal of the ACM (JACM)*, 45(5):783–797, 1998.

13 Dawei Huang and Seth Pettie. Approximate generalized matching: f-factors and f-edge covers. *arXiv preprint*, 2017. `arXiv:1706.05761`.

14 Alexander V Karzanov. On finding maximum flows in networks with special structure and some applications. *Matematicheskie Voprosy Upravleniya Proizvodstvom*, 5:81–94, 1973.

15 Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{rank})$ iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433. IEEE, 2014.

16 A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.