RIMS-1718

Optimal matching forests and valuated delta-matroids

By

Kenjiro TAKAZAWA

March 2011

京都大学　数理解析研究所

RESEARCH INSTITUTE FOR MATHEMATICAL SCIENCES

KYOTO UNIVERSITY, Kyoto, Japan

Optimal matching forests and valuated delta-matroids

Kenjiro Takazawa

Research Institute for Mathematical Sciences,
Kyoto University, Kyoto 606-8502, Japan.
takazawa@kurims.kyoto-u.ac.jp

March 16, 2011

Abstract

The matching forest problem in mixed graphs is a common generalization of the matching problem in undirected graphs and the branching problem in directed graphs. Giles presented an $O(n^2m)$-time algorithm for finding a maximum-weight matching forest, where n is the number of vertices and m is that of edges, and a linear system describing the matching forest polytope. Later, Schrijver proved total dual integrality of the linear system.

In the present paper, we reveal another nice property of matching forests: the degree sequences of the matching forests in any mixed graph form a delta-matroid and the weighted matching forests induce a valuated delta-matroid. We remark that the delta-matroid is not necessarily even, and the valuated delta-matroid induced by weighted matching forests slightly generalizes the well-known notion of Dress and Wenzel's valuated delta-matroids. By focusing on the delta-matroid structure and reviewing Giles' algorithm, we design a simpler $O(n^2m)$-time algorithm for the weighted matching forest problem. We also present a faster $O(n^3)$-time algorithm by using Gabow's method for the weighted matching problem.

1 Introduction

The concept of *matching forests* in mixed graphs was introduced by Giles [14, 15, 16] as a common generalization of matchings in undirected graphs and branchings in directed graphs. Let $G = (V, E, A)$ be a mixed graph with vertex set V, undirected edge set E and directed edge set A. Let n and m denote $|V|$ and $|E \cup A|$, respectively. For a vector $x \in \mathbf{R}^{E\cup A}$ and $F \subseteq E \cup A$, let $x(F) := \sum_{e\in F} x(e)$.

We denote a directed edge $a \in A$ from $u \in V$ to $v \in V$ by uv. A directed edge is often called an *arc*. For an arc $a = uv$, the terminal vertex v is called the *head* of a and denoted by $\partial^- a$, and the initial vertex u is called the *tail* of a and denoted by $\partial^+ a$. For a vertex $v \in V$, the set of arcs whose head (resp., tail) is v is denoted by $\delta^- v$ (resp., $\delta^+ v$). For $B \subseteq A$, let $\partial^- B = \bigcup_{a\in B} \partial^- a$. A vertex in $\partial^- B$ is said to be *covered* by B. An arc subset $B \subseteq A$ is a *branching* if the underlying edge set of B is a forest and each vertex $v \in V$ is the head of at most one arc in B. For a branching B, a vertex not covered by B is called a *root* of B, and the set of the roots of B is denoted by $R(B)$, i.e., $R(B) = V \setminus \partial^- B$.

An undirected edge $e \in E$ connecting $u, v \in V$ is denoted by (u, v). We often abbreviate (u, v) as uv, where it obvious that it is undirected. For $e = uv \in E$, both u and v are called as the *head* of e, and the set of heads of e is denoted by ∂e, i.e., $\partial e = \{u, v\}$. For a vertex v, the set of edges incident to v is denoted by δv. For $F \subseteq E$, let $\partial F = \bigcup_{e\in F} \partial e$. A vertex in ∂F is said to be *covered* by F. An undirected edge subset $M \subseteq E$ is a *matching* if each vertex $v \in V$ is the head of at most one edge in M. A vertex not covered by M is called a *root* of M and the set of the roots of M is denoted by $R(M)$, i.e., $R(M) = V \setminus \partial M$.

An edge set $F \subseteq E \cup A$ is a *matching forest* if the underlying edge set of F is a forest and each vertex in V is the head of at most one edge in F. Equivalently, an edge set $F = B \cup M$, where $B \subseteq A$ and $M \subseteq E$, is a matching forest if B is a branching and M is a matching with $\partial M \subseteq R(B)$. A vertex in $\partial^- B \cup \partial M$ are said to be *covered* by F, and a vertex is a *root* of F if it is not covered by F. The set of the roots of F is denoted by $R(F)$. Observe that $R(F) = R(B) \cap R(M)$ and $V = R(B) \cup R(M)$.

1.1 Background

Matching forests inherit the tractability of branchings and matchings. Let $w \in \mathbf{R}^{E \cup A}$ be a weight vector on the edge set of a mixed graph $G = (V, E, A)$. We consider the *weighted matching forest problem*, the objective of which is to find a matching forest F maximizing $w(F)$. For this problem, Giles [15] designed a primal-dual algorithm running in $\mathrm{O}(n^2 m)$ time, which provided a constructive proof for integrality of a linear system describing the matching forest polytope. Later, Schrijver [21] proved that Giles' linear system is totally dual integral. These results commonly extend the polynomial-time solvability and the total dual integrality results for the weighted branchings and weighted matchings [4, 7, 9].

Topics related to matching forests include the following. Using the notion of matching forests, Keijsper [17] gave a common extension of Vizing's theorem [23, 24] on covering undirected graphs by matchings and Frank's theorem [11] on covering directed graphs by branchings. Another aspect of matching forests is that they can be represented as *linear matroid matching* (see [22]). From this viewpoint, however, we do not fully understand the tractability of matching forests, since the weighted linear matroid matching problem is unsolved while the unweighted problem is solved [18].

In the present paper, we reveal a relation between matching forests and *delta-matroids* [1, 3, 5] to offer a new perspective on weighted matching forests which explains their tractability. For a finite set V and $\mathcal{F} \subseteq 2^V$, the pair $(V, \mathcal{F})$ is a delta-matroid if it satisfies the following exchange property:

(DM) $\forall S_1, S_2 \in \mathcal{F}$, $\forall s \in S_1 \triangle S_2$, $\exists t \in S_1 \triangle S_2$, $S_1 \triangle \{s, t\} \in \mathcal{F}$.

Here, $\triangle$ denotes the symmetric difference, i.e., $S_1 \triangle S_2 = (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$.

A typical example of a delta-matroid is a *matching delta-matroid.* For an undirected graph $G = (V, E)$, let $\mathcal{F}_M = \{\partial M \mid M \text{ is a matching in } G\}$. Then, $(V, \mathcal{F}_M)$ is a delta-matroid [2, 3]. Branchings in a directed graph also induce a delta-matroid, which we call a *branching delta-matroid.* For a directed graph $G = (V, A)$, let $\mathcal{F}_B = \{R(B) \mid B \text{ is a branching in } G\}$. Then, it is not difficult to verify that $\mathcal{F}_B$ is a delta-matroid (see § 2.1).

A delta-matroid $(V, \mathcal{F})$ is called *even* if $|S_1| - |S_2|$ is even for any $S_1, S_2 \in \mathcal{F}$. Note that a matching delta-matroid is an even delta-matroid, whereas a branching delta-matroid is not. Even delta-matroids are characterized by the following simultaneous exchange property [25]:

(EDM) $\forall S_1, S_2 \in \mathcal{F}$, $\forall s \in S_1 \triangle S_2$, $\exists t \in (S_1 \triangle S_2) \setminus \{s\}$, $S_1 \triangle \{s, t\} \in \mathcal{F}$ and $S_2 \triangle \{s, t\} \in \mathcal{F}$.

The concept of *valuated delta-matroids* [6, 26] is a quantitative generalization of even delta-matroids. A function $f : 2^V \to \mathbf{R} \cup \{-\infty\}$ is a valuated delta-matroid if $\mathrm{dom} f \neq \emptyset$ and

(V-EDM) $\forall S_1, S_2 \in \mathrm{dom} f$, $\forall s \in S_1 \triangle S_2$, $\exists t \in (S_1 \triangle S_2) \setminus \{s\}$, $f(S_1 \triangle \{s, t\}) + f(S_2 \triangle \{s, t\}) \geq f(S_1) + f(S_2)$.

Here, $\mathrm{dom}f := \{S \mid S \subseteq V,\ f(S) \neq -\infty\}$. Note that $(V, \mathrm{dom}f)$ is an even-delta matroid. We remark here that weighted matchings in a weighted undirected graph induce a valuated delta-matroid f_M with $\mathrm{dom}f_M = \mathcal{F}_M$ (see § 2.1).

1.2 Contributions

In this paper, we consider delta-matroids commonly extending matching delta-matroids and branching delta-matroids, and also a valuation on those delta-matroids. For this purpose, we introduce a new class of delta-matroids which properly includes even delta-matroids. We call $(V, \mathcal{F})$ a *simultaneous delta-matroid* if it satisfies the following weaker simultaneous exchange property:

(SDM) $\forall S_1, S_2 \in \mathcal{F}$, $\forall s \in S_1 \triangle S_2$, $\exists t \in S_1 \triangle S_2$, $S_1 \triangle \{s,t\} \in \mathcal{F}$ and $S_2 \triangle \{s,t\} \in \mathcal{F}$.

Note that every even delta-matroid is a simultaneous delta-matroid. Also, a branching matroid is a simultaneous delta-matroid (see § 2.1).

The first main result in this paper is that matching forests also induce a simultaneous delta-matroid. For a mixed graph $G = (V, E, A)$, let $\mathcal{F}_{MF} = \{R(F) \mid F \text{ is a matching forest}\}$. We prove that $\mathcal{F}_{MF}$ is a simultaneous delta-matroid.

Theorem 1. *For any mixed graph $G = (V, E, A)$, it holds that $(V, \mathcal{F}_{MF})$ is a simultaneous delta-matroid.*

Furthermore, we generalize the notion of valuated delta-matroids in order to deal with a quantitative extension of Theorem 1. That is, we define valuated delta-matroids on simultaneous delta-matroids, which slightly generalize valuated delta-matroids on even delta-matroids [6]. We call a function $f : 2^V \to \mathbf{R} \cup \{-\infty\}$ a *valuated delta-matroid* if $\mathrm{dom}f \neq \emptyset$ and

(V-SDM) $\forall S_1, S_2 \in \mathrm{dom}f$, $\forall s \in S_1 \triangle S_2$, $\exists t \in S_1 \triangle S_2$, $f(S_1 \triangle \{s,t\}) + f(S_2 \triangle \{s,t\}) \geq f(S_1) + f(S_2)$.

Note that $(V, \mathrm{dom}f)$ is a simultaneous delta-matroid.

For a weighted mixed graph (G, w) with $G = (V, E, A)$ and $w \in \mathbf{R}^{E \cup A}$, define a function $f_{MF} : 2^V \to \mathbf{R} \cup \{-\infty\}$ by

$$f_{MF}(S) = \begin{cases} \max\{w(F) \mid F \text{ is a matching forest with } R(F) = S\} & (S \in \mathcal{F}_{MF}), \\ -\infty & (\text{otherwise}). \end{cases}$$

We prove that f_{MF} satisfies (S-VDM).

Theorem 2. *For any weighted mixed graph (G, w), it holds that f_{MF} is a valuated delta-matroid.*

Proofs for Theorems 1 and 2 will be given in § 2.2. We remark that the relation between valuated delta-matroids in the sense of [6] and those in our sense is similar to that between M-concave functions and M$^\natural$-concave functions [20].

The next contribution of this paper is new algorithms for the weighted matching forest problem: we design a simpler algorithm and a faster algorithm than Giles' algorithm [15]. In § 3, we present a simple $\mathrm{O}(n^2 m)$-time algorithm which focuses on the delta-matroid structure. We also present an $\mathrm{O}(n^3)$-time algorithm in § 4 by using the technique of Gabow [13] for the weighted matching problem.

2 Delta-matroids and matching forests

In this section, we prove Theorems 1 and 2. That is, we show relations between delta-matroids and matching forests, and between valuated delta-matroids and weighted matching forests.

2.1 Matching delta-matroids and branching delta-matroids

In this subsection, we describe basic facts on delta-matroids, including their relations to matchings and branchings. We begin with exhibiting two operations on delta-matroids. The *dual* of a delta-matroid $(V, \mathcal{F})$ is a delta-matroid $(V, \bar{\mathcal{F}})$, defined by $\bar{\mathcal{F}} = \{V \setminus S \mid S \in \mathcal{F}\}$. The *union* of two delta-matroids $(V, \mathcal{F}_1)$ and $(V, \mathcal{F}_2)$ is a pair $(V, \mathcal{F}_1 \vee \mathcal{F}_2)$ defined by $\mathcal{F}_1 \vee \mathcal{F}_2 = \{S_1 \cup S_2 \mid S_1 \in \mathcal{F}_1, S_2 \in \mathcal{F}_2, S_1 \cap S_2 = \emptyset\}$, which is a delta-matroid [2].

The relation between matchings and delta-matroids is well-known. Let (G, w) be a weighted undirected graph with $G = (V, E)$ and $w \in \mathbf{R}^E$. As stated in § 1, the pair $(V, \mathcal{F}_M)$, where $\mathcal{F}_M = \{\partial M \mid M \text{ is a matching in } G\}$, is an even delta-matroid, which we call the matching delta-matroid of G. Moreover, a function $f_M : 2^V \to \mathbf{R} \cup \{-\infty\}$ defined below is a valuated delta-matroid [19]:

$$f_M(S) = \begin{cases} \max\{w(M) \mid M \text{ is a matching with } \partial M = S\} & (S \in \mathcal{F}_M), \\ -\infty & (\text{otherwise}). \end{cases}$$

We now present a relation between branchings and delta-matroids. Let (G, w) be a weighted directed graph with $G = (V, A)$ and $w \in \mathbf{R}^A$. Recall that $\mathcal{F}_B = \{R(B) \mid B \text{ is a branching in } G\}$. It is verified that $(V, \mathcal{F}_B)$ is a delta-matroid as follows. For a directed graph G, a strong component is called a *source component* if it has no arc entering from other strong components. The vertex set and arc set of a strong component K are denoted by VK and AK, respectively. Let $K_1, \ldots, K_l$ be all source components in G. Then, we have that $\mathcal{F}_B = \{S \mid S \subseteq V,\ |S \cap VK_i| \geq 1 \text{ for } i = 1, \ldots, l\}$. Thus, it follows that $(V, \mathcal{F}_B)$ is a generalized matroid [12]. Moreover, it also follows that $(V, \mathcal{F}_B)$ satisfies (SDM). We call $(V, \mathcal{F}_B)$ as the branching delta-matroid of G.

Theorem 3. *For any directed graph G, it holds that $(V, \mathcal{F}_B)$ is a simultaneous delta-matroid.*

Furthermore, this fact extends to weighted branchings. Define $f_B : 2^V \to \mathbf{R} \cup \{-\infty\}$ by

$$f_B(S) = \begin{cases} \max\{w(B) \mid B \text{ is a branching with } R(B) = S\} & (S \in \mathcal{F}_B), \\ -\infty & (\text{otherwise}). \end{cases}$$

Then, f_B is a valuated delta-matroid, which immediately follows from arguments in Schrijver [21, Theorem 1].

Theorem 4. *For any weighted directed graph (G, w), it holds that f_B is a valuated delta-matroid.*

2.2 Delta-matroids and matching forests

In this subsection, we prove Theorems 1 and 2. We begin with a simple proof showing that $(V, \mathcal{F}_{MF})$ is a delta-matroid for a mixed graph (V, E, A). Let $\mathcal{F}_M$ be the matching delta-matroid of (V, E) and $\mathcal{F}_B$ the branching delta-matroid of (V, A). Then, it immediately follows from the definition of matching forests that $\mathcal{F}_{MF}$ is the dual of $\mathcal{F}_M \vee \bar{\mathcal{F}}_B$, and thus $(V, \mathcal{F}_{MF})$ is a delta-matroid.

We now prove Theorem 1, which is a stronger statement. First, Schrijver [21] proved the following exchange property of branchings.

Lemma 5 (Schrijver [21]). *Let $G = (V, A)$ be a directed graph, and B_1 and B_2 be branchings partitioning A. Let R_1 and R_2 be vertex sets with $R_1 \cup R_2 = R(B_1) \cup R(B_2)$ and $R_1 \cap R_2 = R(B_1) \cap R(B_2)$. Then A can be split into branchings B_1' and B_2' with $R(B_i') = R_i$ for $i = 1, 2$ if and only if each source component K in G satisfies that $|K \cap R_i| \geq 1$ for $i = 1, 2$.*

By using Lemma 5, Schrijver proved an exchange property of matching forests [21, Theorem 2]. Here, we show another exchange property of matching forests, which relates them to simultaneous delta-matroids. The proof below is quite similar to the proof for Theorem 2 in [21]. For completeness, however, we describe a full proof.

Lemma 6. *Let $G = (V, E, A)$ be a mixed graph, F_1 and F_2 be matching forests partitioning $E \cup A$, and $s \in R(F_2) \setminus R(F_1)$. Then, there exist matching forests F_1' and F_2' which partition $E \cup A$ and satisfy one of the following:*

(i) $R(F_1') = R(F_1) \cup \{s\}$ *and* $R(F_2') = R(F_2) \setminus \{s\}$,

(ii) $R(F_1') = R(F_1) \cup \{s, t\}$ *and* $R(F_2') = R(F_2) \setminus \{s, t\}$ *for some* $t \in R(F_2) \setminus (R(F_1) \cup \{s\})$,

(iii) $R(F_1') = (R(F_1) \cup \{s\}) \setminus \{t\}$ *and* $R(F_2') = (R(F_2) \setminus \{s\}) \cup \{t\}$ *for some* $t \in R(F_1) \setminus R(F_2)$.

Proof. Let $M_i := F_i \cap E$ and $B_i := F_i \cap A$ for $i = 1, 2$. Denote the family of the source components in (V, A) by $\mathcal{K}$. If $v \in R(B_1) \cap R(B_2)$ for $v \in V$, then we have $\{v\} \in \mathcal{K}$. Thus, for a source component $K \in \mathcal{K}$ with $|K| \geq 2$, $K \cap R(B_1)$ and $K \cap R(B_2)$ are not empty and disjoint with each other. For each $K \in \mathcal{K}$ with $|K| \geq 2$, choose a pair e_K of vertices, one of which is in $K \cap R(B_1)$ and the other in $K \cap R(B_2)$. Denote $N = \{e_K \mid K \in \mathcal{K}\}$. Note that N is a matching.

Construct an undirected graph $H = (V, M_1 \cup M_2 \cup N)$. We have that H is a disjoint collection of paths and cycles. For, an endpoint u of an edge $e_K \in N$ satisfies that either $u \in \partial^- B_1$ or $u \in \partial^- B_2$, and thus u is not covered by both of M_1 and M_2. Moreover, we have that s is an endpoint of a path P in H. For, since $s \in R(F_2)$, we have that s is not covered by M_2. If s is covered by M_1, then $s \in R(B_1)$, and thus $s \in R(B_1) \cap R(B_2)$. This implies that s is not covered by N.

Denote the set of vertices on P by VP, the set of edges in $M_1 \cup M_2$ on P by EP, and let $M_1' := M_1 \triangle EP$ and $M_2' := M_2 \triangle EP$. Then, both M_1' and M_2' are matchings and

$$R(M_1') = (R(M_1) \setminus VP) \cup (R(M_2) \cap VP), \qquad R(M_2') = (R(M_2) \setminus VP) \cup (R(M_1) \cap VP).$$

Now, by Lemma 5, there exist disjoint branchings B_1' and B_2' such that

$$R(B_1') = (R(B_1) \setminus VP) \cup (R(B_2) \cap VP), \qquad R(B_2') = (R(B_2) \setminus VP) \cup (R(B_1) \cap VP).$$

(Note that $|K \cap R(B_i')| \geq 1$ for $i = 1, 2$ for every source component K.)

Since $R(B_i) \cup R(M_i) = V$, we have that $F_i' := B_i' \cup M_i'$ is a matching forest for $i = 1, 2$, and

$$R(F_1') = (R(F_1) \setminus VP) \cup (R(F_2) \cap VP), \qquad R(F_2') = (R(F_2) \setminus VP) \cup (R(F_1) \cap VP).$$

If $VP = \{s\}$, then Assertion (i) applies. Otherwise, denote the other endpoint of P by t. If $t \in V \setminus (R(F_1) \triangle R(F_2))$, then Assertion (i) applies. If $t \in R(F_2) \setminus R(F_1)$, then Assertion (ii) applies. If $t \in R(F_1) \setminus R(F_2)$, then Assertion (iii) applies. □

Theorem 1 is obvious from Lemma 6. Furthermore, Theorem 2 also follows from Lemma 6.

Proof for Theorem 2. Let $S_1, S_2 \in \mathrm{dom} f$ and $s \in S_1 \triangle S_2$. For $i = 1, 2$, let F_i be a matching forest such that $R(F_i) = S_i$ and $w(F_i) = f_{MF}(S_i)$. Without loss of generality, assume $s \in R(F_2) \setminus R(F_1)$. By applying Lemma 6 to the mixed graph consisting of the edges in F_1 and F_2, we obtain matching forests F_1' and F_2' such that $w(F_1') + w(F_2') = w(F_1) + w(F_2)$ and satisfying one of Assertions (i)–(iii). Now the statement follows from $w(F_1') \leq f_{MF}(R(F_1'))$ and $w(F_2') \leq f_{MF}(R(F_2'))$. □

3 A simpler algorithm

Let (G,w) be a weighted mixed graph with $G=(V,E,A)$ and $w \in \mathbf{R}^{E\cup A}$. In this section, we describe a primal-dual algorithm for finding a matching forest F maximizing $w(F)$. This algorithm is a slight modification of Giles' algorithm [15]. The main difference results from focusing the delta-matroid structure of branchings (Theorem 3).

3.1 LP formulation for the weighted matching forest problem

For a subpartition $\mathcal{L}$ of V, let $\cup\mathcal{L}$ denote the union of the sets in $\mathcal{L}$ and let

$$\gamma(\mathcal{L}) := \{e \mid e \in E,\ e \text{ is contained in } \cup\mathcal{L}\} \cup \{a \mid a \in A,\ a \text{ is contained in some set in } \mathcal{L}\}.$$

Let Λ denote the collection of subpartition $\mathcal{L}$ of V with $|\mathcal{L}|$ odd. The following is a linear programming relaxation of an integer program describing the weighted matching forest problem:

$$\begin{array}{llll}
\text{(P)} & \text{maximize} & \displaystyle\sum_{e\in E\cup A} w(e)x(e) & \\
 & \text{subject to} & x(\delta^{\text{head}}(v)) \leq 1 & (v \in V), \quad (1)\\
 & & x(\gamma(\mathcal{L})) \leq \lfloor |\cup\mathcal{L}| - |\mathcal{L}|/2 \rfloor & (\mathcal{L} \in \Lambda), \quad (2)\\
 & & x(e) \geq 0 & (e \in E\cup A). \quad (3)
\end{array}$$

Here, $\delta^{\text{head}}(v) \subseteq E \cup A$ denotes the set of edges which have v as a head, i.e., $\delta^{\text{head}}(v) = \delta v \cup \delta^{-} v$. Note that the above linear system is a common extension of those describing the weighted matching problem [7] and the weighted branching problem [9]. Giles [15] proved the integrality of the system (1)–(3).

Theorem 7 ([15])**.** *For any weighted mixed graph* (G,w)*, the linear program* (P) *has an integer optimal solution.*

Furthermore, Schrijver [21] proved that the system (1)–(3) is totally dual integral [10], which commonly extends the total dual integrality of those for matchings [4] and for branchings. That is, Schrijver proved that the following dual problem of (P) has an integer optimal solution if w is integer:

$$\begin{array}{llll}
\text{(D)} & \text{minimize} & \displaystyle\sum_{v\in V} y(v) + \sum_{\mathcal{L}\in\Lambda} z(\mathcal{L}) \lfloor |\cup\mathcal{L}| - |\mathcal{L}|/2 \rfloor & \\
 & \text{subject to} & \displaystyle y(u) + y(v) + \sum_{\mathcal{L}\colon e\in\gamma(\mathcal{L})} z(\mathcal{L}) \geq w(e) & (e = uv \in E), \quad (4)\\
 & & \displaystyle y(v) + \sum_{\mathcal{L}\colon a\in\gamma(\mathcal{L})} z(\mathcal{L}) \geq w(a) & (a = uv \in A), \quad (5)\\
 & & y(v) \geq 0 & (v \in V), \quad (6)\\
 & & z(\mathcal{L}) \geq 0 & (\mathcal{L} \in \Lambda). \quad (7)
\end{array}$$

Theorem 8 ([21])**.** *For any weighted mixed graph* (G,w) *with* w *integer, the linear program* (D) *has an integer optimal solution.*

Define the reduced weight $w' \in \mathbf{R}^{E \cup A}$ by

$$w'(e) = y(u) + y(v) + \sum_{\mathcal{L}\colon e \in \gamma(\mathcal{L})} z(\mathcal{L}) - w(e) \quad (e = uv \in E),$$
$$w'(a) = y(v) + \sum_{\mathcal{L}\colon a \in \gamma(\mathcal{L})} z(\mathcal{L}) - w(a) \qquad (a = uv \in A).$$

Below are the complementary slackness conditions of (P) and (D).

$$x(e) > 0 \implies w'(e) = 0 \qquad (e \in E \cup A), \tag{8}$$
$$x(\delta^{\text{head}} v) < 1 \implies y(v) = 0 \qquad (v \in V), \tag{9}$$
$$z(\mathcal{L}) > 0 \implies x(\gamma(\mathcal{L})) = \lfloor |\cup \mathcal{L}| - |\mathcal{L}|/2 \rfloor \quad (\mathcal{L} \in \Lambda). \tag{10}$$

3.2 Algorithm description

3.2.1 Notations

In the algorithm, we keep a matching forest F, which corresponds to an integer feasible solution x of (P), and a dual feasible solution (y, z). We maintain that x and (y, z) satisfy (8) and (10). The algorithm terminates when (9) is satisfied.

Similarly to the classical weighted matching and branching algorithms, we execute shrinking of subgraphs repeatedly. We keep two laminar families $\varDelta$ and $\varUpsilon$ of subsets of V, the former of which results from shrinking a strong component in the directed graph and the latter from shrinking an undirected odd cycle.

We use the following notations to describe the algorithm.

- For a cycle or a path Q in an undirected graph (V, E), let VQ and EQ denote the vertex set and edge set of Q, respectively. We often abbreviate EQ as Q.
- $\varOmega' := \varDelta \cup \varUpsilon$, $\varOmega := \varOmega' \cup \{\{v\} \mid v \in V\}$.
- For each $U \in \varOmega'$, let $G_U = (V_U, E_U, A_U)$ denote the mixed graph obtained from the subgraph induced by U by contracting all maximal proper subsets of U belonging to $\varDelta$. Also, let $\hat{G} = (\hat{V}, \hat{E}, \hat{A})$ denote the mixed graph obtained from G by contracting all maximal sets in $\varOmega'$. We denote a vertex in a shrunk graph by the set of vertices in V which are shrunk into the vertex. Also, we often identify a vertex U in a shrunk graph and the singleton $\{U\}$.
- For $G = (V, E, A)$ and a dual feasible solution (y, z), the *equality subgraph* $G^\circ = (V, E^\circ, A^\circ)$ of G is a subgraph defined by $E^\circ = \{e \mid e \in E,\ w'(e) = 0\}$ and $A^\circ = \{a \mid a \in A,\ w'(a) = 0\}$. We denote the branching delta-matroid in $(\hat{V}, \hat{A}^\circ)$ by $(\hat{V}, \hat{\mathcal{F}}^\circ_B)$, i.e.,

$$\hat{\mathcal{F}}^\circ_B = \{R(\hat{B}) \mid \hat{B} \subseteq \hat{A}^\circ \text{ is a branching in } \hat{G}^\circ\}.$$

The outline of the algorithm is as follows.

- We maintain a matching forest $\hat{F} = \hat{M} \cup \hat{B}$ in $\hat{G}^\circ$, where $\hat{M} \subseteq \hat{E}^\circ$ and $\hat{B} \subseteq \hat{A}^\circ$, in order to maintain (8).
- In contracting a vertex set $U \subseteq V$, we associate a partition $\mathcal{L}_U$ of U such that $x(\gamma(\mathcal{L}_U)) = \lfloor |\cup \mathcal{L}_U| - |\mathcal{L}_U|/2 \rfloor$. The vector z is restricted to subpartitions associated to the sets in $\varOmega'$ in order to maintain (10).

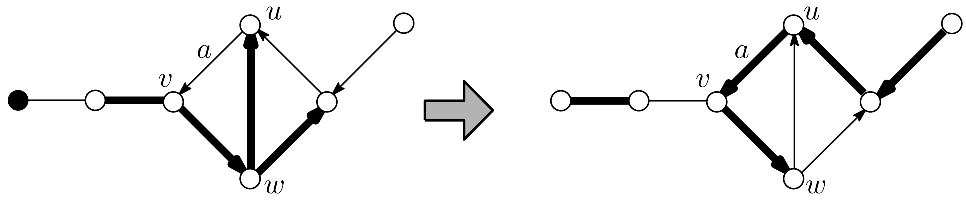

Figure 1: Augmentation. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex.)

- Similarly to Edmonds' matching algorithm [8], we construct an *alternating forest* H, which is a subgraph of $(\hat{V}, \hat{E}^\circ)$. The vertex set and edge set of H are denoted by $\hat{V}H$ and $\hat{E}H$, respectively. We often abbreviate $\hat{E}H$ as H. Each component of H is a tree and contains a unique *source vertex*. Intuitively, a source vertex is a vertex where (9) is not satisfied (see § 3.2.3 for precise definition). For $v \in \hat{V}H$, let P_v denote the path in H connecting a source vertex and v. The edges incident to a source vertex does not belong to $\hat{M}$, and edges in $\hat{M}$ and $\hat{E}^\circ \setminus \hat{M}$ appear alternately on each P_v. We label a vertex v as "even" (resp., "odd") if the length of P_v is even (resp., odd). Here, the length of a path is defined by the number of its edges. The set of vertices labelled as even (resp., odd) is denoted by $\mathrm{even}(H)$ (resp., $\mathrm{odd}(H)$). Also, let $\mathrm{free}(H) := \hat{V} \setminus (\mathrm{even}(H) \cup \mathrm{odd}(H))$.

3.2.2 A rough description of augmentation and shrinking

Before presenting a full description of the algorithm, we briefly sketch how to augment the matching forest, shrink subgraphs and associate a partition with the shrunk vertex set.

To make things easy, let us suppose that no subgraph is shrunk, i.e., $\mathit{\Delta} = \mathit{\Upsilon} = \emptyset$ and $\hat{G} = G$. Denote the current matching forest by $\hat{F} = \hat{M} \cup \hat{B}$, where $\hat{M} \subseteq \hat{E}^\circ$ is a matching and $\hat{B} \subseteq \hat{A}^\circ$ is a branching.

After labeling a vertex v as even in growing the alternating forest H, we search for an arc $a \in \hat{A}^\circ \cap \delta^- v$. Note that arcs in $\delta^- v$ do not belong to $\hat{B}$. If such an arc a is found, our algorithm proceeds as follows.

- If $R(\hat{B}) \setminus \{v\} \in \hat{\mathcal{F}}_B^\circ$, then reset the matching forest $\hat{F} := M' \cup B'$, where $M' := \hat{M} \triangle P_v$ and B' is a branching in $(\hat{V}, \hat{A}^\circ)$ with $R(B') = R(\hat{B}) \setminus \{v\}$. This procedure is one kind of *augmentation*, in which the number of vertices violating (9) decreases. See Figure 1 for an illustration.

- If $R(\hat{B}) \setminus \{v\} \notin \hat{\mathcal{F}}_B^\circ$, it follows that $v \in \hat{V}K$ for some source component K of $(\hat{V}, \hat{A}^\circ)$ and $\hat{V}K \setminus \{v\} \subseteq \partial^- \hat{B}$. If K contains no undirected edge in $\hat{E}^\circ$, we add $X = \hat{V}K$ to $\mathit{\Delta}$ and update $\hat{G}$ by contracting the vertices in X to a single vertex. The newly created vertex is called a *pseudo-vertex*, denoted by $X \in \hat{V}$. We then define a partition $\mathcal{L}_X$ of X by $\mathcal{L}_X = \{X\}$ and set $z(\mathcal{L}_X) = 0$. Note that (10) holds for $\mathcal{L}_X$, since $x(\gamma(\mathcal{L}_X)) = |X| - 1$, $|\cup \mathcal{L}_X| = |X|$ and $|\mathcal{L}_X| = 1$. See Figure 2 for an illustration. The case where K contains an undirected edge in $\hat{E}^\circ$ will be described in § 3.2.3.

Note that we can determine whether $R(\hat{B}) \setminus \{v\} \in \hat{\mathcal{F}}_B^\circ$ or not by decomposing $(\hat{V}, \hat{A}^\circ)$ into strong components. Also, it is not difficult to find a branching $B' \subseteq \hat{A}^\circ$ with $R(B') = R(B) \setminus \{v\}$.

Remark 9. The above bifurcation is the main difference from Giles' algorithm [15]. In Giles' algorithm, we augment the matching forest if $B'' = \hat{B} \cup \{a\}$ is a branching in $\hat{G}$, which is a

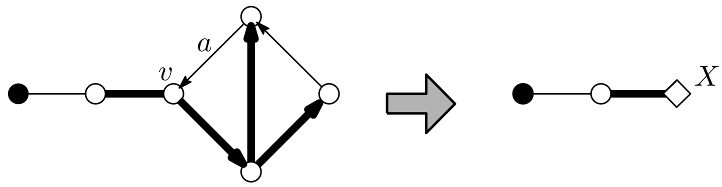

Figure 2: Shrinking of a source component. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The square in the shrunk graph indicates the pseudo-vertex $X \in \hat{V}$.)

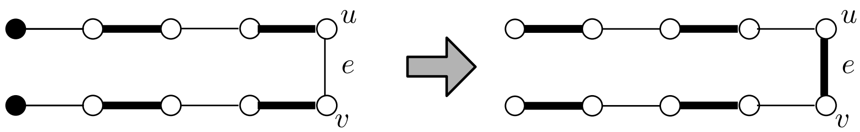

Figure 3: Augmentation. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex.)

sufficient condition for $R(\hat{B}) \setminus \{v\} \in \hat{\mathcal{F}}_B^\circ$. (Here, we reset the matching forest $\hat{F} := M' \cup B''$.) If B'' is not a branching, we have that B'' contains exactly one directed cycle D, and we contract $X' = VD$ to a single vertex. For instance, in Figure 1 we do not augment $\hat{F}$ but contract the directed cycle consisting of u, v, w in Giles' algorithm.

Also, if we find an undirected edge $e \in \hat{E}^\circ$ connecting two even vertices u and v, we do similar procedures as the the classical blossom algorithm [8]. If u and v belong to different components in H, then we augment the matching forest by resetting $\hat{F} := M' \cup \hat{B}$, where $M' := M \triangle (P_u \cup P_v \cup \{e\})$. See Figure 3 for an illustration.

Assume that u and v belong to the same component of H (see Figure 4 for an illustration). Here, $H \cup \{e\}$ contains exactly one odd undirected cycle C. We now add $U = \hat{V}C$ to $\varUpsilon$, and update $\hat{G}$ by contracting the vertices in U to a pseudo-vertex $U \in \hat{V}$. We then define a partition $\mathcal{L}_U$ of U by $\mathcal{L}_U = \{\{v\} \mid v \in U\}$ and set $z(\mathcal{L}_U) = 0$. Note that (10) holds for $\mathcal{L}_U$, since $x(\gamma(\mathcal{L}_U)) = \lfloor |U|/2 \rfloor$ and $|\cup \mathcal{L}_U| = |\mathcal{L}_U| = |U|$.

Dealing with a subgraph containing pseudo-vertices is a bit more complicated. Consider shrinking a source component K in $\hat{G}$ and let $X \subseteq V$ be the union of vertices in $\hat{V}K$. Denote the maximal proper subsets of X belonging to $\varDelta$ and $\varUpsilon$ by $Y_1, \ldots, Y_k \in \varDelta$ and $W_1, \ldots, W_l \in \varUpsilon$, respectively. Here, we can assume that arcs in $\hat{A}K \cap (\delta^+ W_i \cup \delta^- W_i)$ are incident to an identical vertex $v_{W_i} \in V_{W_i}$ for every $i = 1, \ldots, l$, since otherwise we can augment the current matching forest (see Figure 5). Let $X' = (X \setminus (W_1 \cup \cdots \cup W_l)) \cup \{v_{W_1}\} \cup \cdots \cup \{v_{W_l}\}$. Note that X' forms the vertex set of a strong component in (V, A°). We now add X' to $\varDelta$ and let $\mathcal{L}_{X'} = \{X'\}$ be the associated partition with

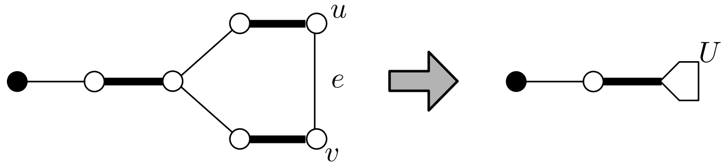

Figure 4: Shrinking of an undirected odd cycle. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The pentagon in the shrunk graph indicates the pseudo-vertex $U \in \hat{V}$.)

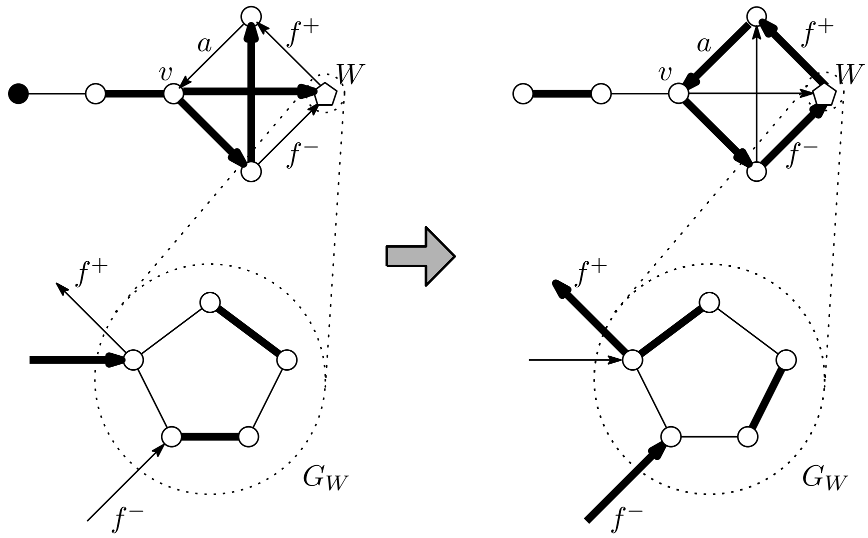

Figure 5: Augmentation. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The graph inside the dotted circle indicates G_W, which is shrunk into the vertex W in $\hat{G}^{\circ}$.)

X'. See Figure 6 for an illustration.

Finally, consider shrinking an odd undirected cycle C. Let $U \subseteq V$ denote the union of vertices in $\hat{V}C$. Denote the maximal proper subsets of U belonging to $\mathit{\Delta}$ by $Y_1 \dots, Y_k \in \mathit{\Delta}$ and the proper subsets of U belonging to $\mathit{\Upsilon}$ $W_1 \dots, W_l \in \mathit{\Upsilon}$, respectively. Let C_U be an odd cycle in G_U which can be obtained by adding even number of edges from each E_{W_j} to C. For $i = 1, \dots, k$, let $f_i^1, f_i^2 \in C_U$ denote the two edges incident to Y_i, and let $v_i^1, v_i^2 \in Y_i$ denote the vertices to which f_i^1 and f_i^2 are incident, respectively. If, for some Y_i, the two vertices v_i^1 and v_i^2 are distinct and $z(\mathcal{L}_{Y_i'}) = 0$ for the minimal subset $Y_i' \in \mathit{\Delta}$ of Y_i such that $\{v_i^1, v_i^2\} \subseteq Y_i'$, we can augment the current matching forest (see Figure 7 for an illustration).

Suppose otherwise. Without loss of generality, assume that v_i^1 and v_i^2 are identical for $i = 1, \dots, j$, and distinct for $i = j+1, \dots, k$. For $i = j+1, \dots, k$, let $Y_i' \in \mathit{\Delta}$ be the minimal subset of Y_i such that $\{v_i^1, v_i^2\} \subseteq Y_i'$. Note that $z(\mathcal{L}_{Y_i'}) > 0$ for $i = j+1, \dots, k$. Let $U' = (U \setminus (Y_1 \cup \dots \cup Y_k)) \cup \{v_{Y_1}, \dots, v_{Y_j}\} \cup (Y_{j+1}' \cup \cdots Y_k')$. We now add U' to $\mathit{\Upsilon}$, and define a partition $\mathcal{L}_{U'}$ of U' by the collection of $\{v_{Y_1}\}, \dots, \{v_{Y_j}\}, Y_{j+1}', \dots, Y_k'$ and singletons of the other vertices in U'. See Figure 8 for an illustration.

3.2.3 A full description of the algorithm

We now present a full description of our algorithm.

Algorithm SIMPLE

Input. A weighted mixed graph (G, w), where $G = (V, E, A)$ and $w \in \mathbf{R}^{E \cup A}$.

Output. A matching forest F in G maximizing $w(F)$.

Step 1. Set $F := \emptyset$, $y(v) := \max\{\{w(e)/2 \mid e \in E\}, \{w(a) \mid a \in A\}\}$ for every $v \in V$, $\mathit{\Delta} := \emptyset$ and $\mathit{\Upsilon} := \emptyset$. (Hence $\mathit{\Omega} = \{\{v\} \mid v \in V\}$, $\hat{G} = G$, $\hat{F} = \emptyset$ and z is void.)

Step 2. Construct the equality subgraph $\hat{G}^{\circ} = (\hat{V}, \hat{E}^{\circ}, \hat{A}^{\circ})$. Define the set of source vertices $\hat{S} = \{U \mid U \in \hat{V},\ y(v) > 0 \text{ and } x(\delta^{\text{head}}(v)) = 0 \text{ for some } v \in U\}$. If $\hat{S} = \emptyset$, deshrink every sets in

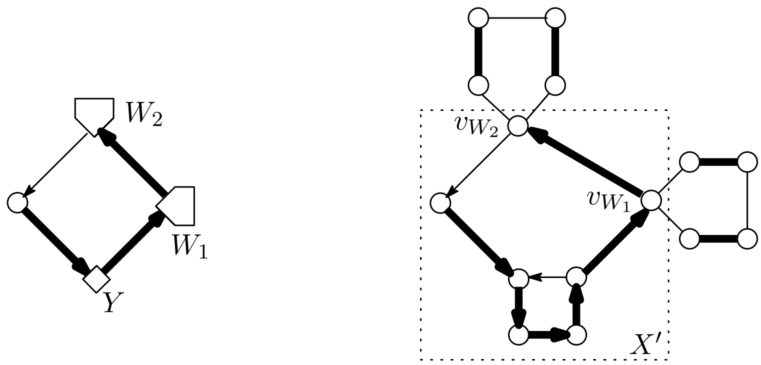

Figure 6: The graph on the left is a strong component K in $\hat{G}$, where $Y \in \Delta$ and $W_1, W_2 \in \Upsilon$. The graph on the right represents a graph obtained by deshrinking Y, W_1 and W_2. The dotted squares indicate X', which is newly added to Δ.

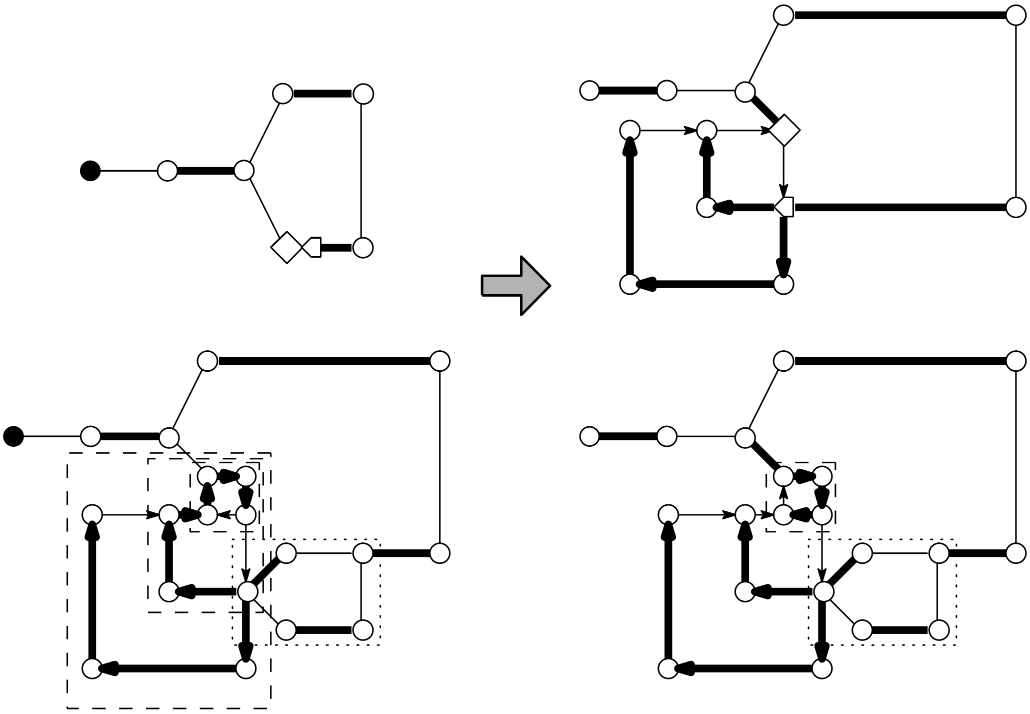

Figure 7: The two graphs above are $\hat{G}$ and those below are G. The dotted box indicates $W \in \Upsilon$ and the nested three dashed boxes indicate $Y, Y', Y'' \in \Delta$, where $Y'' \subseteq Y' \subseteq Y$ and $z(\mathcal{L}_{Y'}) = z(\mathcal{L}_Y) = 0$. In the present step, augmentation and deshrinking of Y and Y' are executed.

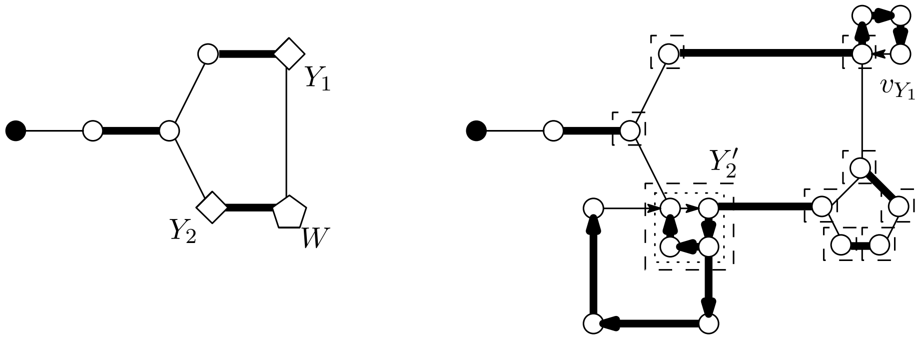

Figure 8: The graph on the left is an odd cycle C in $\hat{G}$, where $Y_1, Y_2 \in \Delta$ and $W \in \Upsilon$. The graph in the right represents G, where the dotted box indicates Y_2' with $z(\mathcal{L}_{Y_2'}) > 0$. The set of vertices inside the dashed boxes is U', where the dashed boxes indicate the partition $\mathcal{L}_{U'}$.

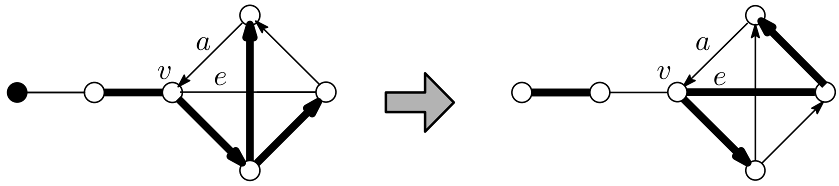

Figure 9: Augmentation in Step 4.2.1. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex.)

Ω' and return F. Otherwise, let H be $(\hat{S}, \emptyset)$, label the vertices in $\hat{S}$ as even, and then go to Step 3.

Step 3. If there exists an arc $a \in \hat{A}^\circ \setminus \hat{B}$ with $\partial^- a \in \mathrm{even}(H)$, then go to Step 4. Otherwise, go to Step 5.

Step 4. Let $v := \partial^- a$. If $R(\hat{B}) \setminus \{v\} \in \hat{\mathcal{F}}_B^\circ$, then go to Step 4.1. Otherwise, go to Step 4.2.

Step 4.1: Augmentation. Reset $\hat{F} := M' \cup B'$, where $M' := \hat{M} \triangle P_v$ and B' is a branching in $(\hat{V}, \hat{A}^\circ)$ with $R(B') = R(\hat{B}) \setminus \{v\}$. Delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figure 1 for an illustration.

Step 4.2. Let K be the source component containing v and let $X \subseteq V$ be the union of vertices in $\hat{V}K$.

- If there exists $e \in \hat{E}^\circ \setminus \hat{M}$ such that $\partial e \subseteq \hat{V}K$, then go to Step 4.2.1.
- Otherwise, go to Step 4.2.2.

Step 4.2.1: Augmentation. Let B_K be a branching in K with $R(B_K) = \partial e$. Reset $\hat{F} := M' \cup B'$, where $M' := (\hat{M} \triangle P_v) \cup \{e\}$ and $B' := (\hat{B} \setminus \hat{A}K) \cup B_K$, delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figure 9 for an illustration.

Step 4.2.2. Let $W_1, \ldots, W_l$ be the maximal proper subsets of X belonging to Υ. If, for some $i \in \{1, \ldots, l\}$, $\hat{A}K$ contains a pair of arcs $f^+ \in \delta^+ W_i$ and $f^- \in \delta^- W_i$ such that $\partial^+ f^+$ and $\partial^- f^-$ belong to distinct vertices in G_{W_i}, then go to Step 4.2.2.1. Otherwise, go to Step 4.2.2.2.

Step 4.2.2.1: Augmentation. Let B_K be a branching in K such that $R(B_K) = \{W_i\}$ and $f^+ \in B_K$. Reset $\hat{F} := M' \cup B'$, where $M' := \hat{M} \triangle P_v$ and $B' := (\hat{B} \setminus \hat{A}(K)) \cup B_K \cup \{f^-\}$. Then, delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω' and go to Step 2. See Figure 5 for an illustration.

Step 4.2.2.2: Shrinking. For each $i = 1, \ldots, l$, let $v_{W_i} \in V_{W_i}$ denote the unique vertex in G_{W_i} to which arcs in $\hat{A}K$ are incident. Let $X' = (X \setminus (W_1 \cup \cdots \cup W_l)) \cup \{v_{W_i}\} \cup \cdots \cup \{v_{W_l}\}$ and add X' to Δ. Let $\mathcal{L}_{X'} = \{X'\}$ be the associated partition with X', set $z(\mathcal{L}_{X'}) := 0$, and then go to Step 3. See Figure 6 for an illustration.

Step 5. Choose an edge $e \in \hat{E}^{\circ} \setminus \hat{E}H$ such that one of its head u is even. Denote the other head of e by v.

- If $v \in \mathrm{even}(H)$ and e connects different components in H, then go to Step 5.1.
- If $v \in \mathrm{even}(H)$ and u and v belong to the same component in H, then go to Step 5.2.
- If $v \in \mathrm{free}(H)$ and $v = \partial^- a$ for some $a \in \hat{B}$, then go to Step 5.3.
- If $v \in \mathrm{free}(H)$ and $v \in \partial e'$ for some $e' \in \hat{M}$, then go to Step 5.4.
- If v is a pseudo-vertex labelled as "saturated," then go to Step 5.5.

If no edge in $\hat{E}^{\circ} \setminus \hat{E}(H)$ satisfies the above conditions, then go to Step 6.

Step 5.1: Augmentation. Reset $\hat{F} := M' \cup \hat{B}$, where $M' := \hat{M} \triangle (P_u \cup P_v \cup \{e\})$, delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figure 3 for an illustration.

Step 5.2. Let C be the cycle in $H \cup \{e\}$ and let $U \subseteq V$ be the union of the vertices in $\hat{V}C$. Denote the maximal proper subsets of U belonging to Δ and Υ by $Y_1, \ldots, Y_k \in \Delta$ and $W_1 \ldots, W_l \in \Upsilon$. Let C_U be an odd cycle in G_U obtained by adding even number of edges from each E_{W_j} to C. For $i = 1, \ldots, k$, let $f_i^1, f_i^2 \in C_U$ denote the two edges incident to Y_i, and let $v_i^1, v_i^2 \in Y_i$ denote the vertices to which f_i^1 and f_i^2 are incident, respectively. If, for some Y_i, the two vertices v_i^1 and v_i^2 are distinct and $z(\mathcal{L}_{Y_i'}) = 0$ for the minimal subset $Y_i' \in \Delta$ of Y_i such that $\{v_i^1, v_i^2\} \subseteq Y_i'$, then go to Step 5.2.1. Otherwise, go to Step 5.2.2.

Step 5.2.1: Deshrinking and augmentation. Delete Y_i' from Δ and reset

$$\hat{M} := \begin{cases} (\hat{M} \triangle P_{Y_i}) \triangle C & (f_i^1, f_i^2 \in E \setminus M), \\ \hat{M} \triangle P_{Y_i}^* & (\text{otherwise}). \end{cases}$$

Here, $P_{Y_i}^*$ denotes the path in $H \cup \{e\}$ from $\hat{S}$ to Y_i consisting of odd number of edges. Delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figure 7 for an illustration.

Step 5.2.2: Shrinking. Without loss of generality, assume that v_i^1 and v_i^2 are identical for $i = 1, \ldots, j$, and distinct for $i = j+1, \ldots, k$. For $i = j+1, \ldots, k$, let $Y_i' \in \Delta$ be the minimal subset of Y_i such that $\{v_i^1, v_i^2\} \subseteq Y_i'$. Let $U' = (U \setminus (Y_1 \cup \cdots \cup Y_k)) \cup \{v_{Y_1}, \ldots, v_{Y_j}\} \cup (Y_{j+1}' \cup \cdots \cup Y_k')$. Add U' to Υ, and define a partition $\mathcal{L}_{U'}$ of U' by the collection of $\{v_{Y_1}\}, \ldots, \{v_{Y_j}\}, Y_{j+1}', \ldots, Y_k'$ and singletons of the other vertices in U'. See Figure 8 for an illustration.

Step 5.3: Augmentation. Reset $\hat{F} := (\hat{M} \triangle P_v) \cup (\hat{B} \setminus \{a\})$, delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figure 10 for an illustration.

Step 5.4: Forest extension. Grow H by adding e and e'. Label v as odd and the other head of e' as even. Then, go to Step 3.

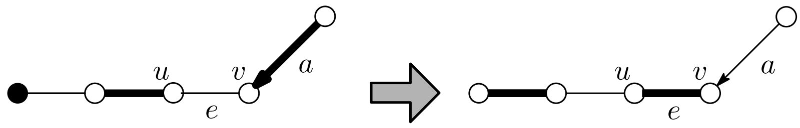

Figure 10: Augmentation in Step 5.3. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex.)

Step 5.5: Augmentation. Reset $\hat{F} := M' \cup \hat{B}$, where $M' := \hat{M} \triangle P_v$, and unlabel v. Delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ and then go to Step 2.

Step 6. Apply Dual_Update described below, delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 3.

Procedure Dual_Update. Define families of vertex subsets of V as follows:

$$\begin{aligned}
\Delta_+ &:= \{\text{maximal set in } \Delta, \text{ contained in some even vertex}\},\\
\Delta_- &:= \{\text{maximal set in } \Delta, \text{ contained in some odd vertex}\},\\
\Upsilon_+ &:= \{\text{maximal set in } \Upsilon, \text{ contained in some even vertex}\},\\
\Upsilon_- &:= \{\text{maximal set in } \Upsilon, \text{ contained in some odd vertex}\}.
\end{aligned}$$

Moreover, let

$$\begin{aligned}
\Delta'_+ &:= \{X \subseteq V \mid X \in \Delta, \text{ maximal proper subset of some element in } \Upsilon_+\},\\
\Delta'_- &:= \{X \subseteq V \mid X \in \Delta, \text{ maximal proper subset of some element in } \Upsilon_-\},\\
V_+ &:= \{v \in V \mid \{v\} \in \mathrm{even}(H) \text{ or } v \text{ is contained in some even vertex}\},\\
V_- &:= \{v \in V \mid \{v\} \in \mathrm{odd}(H) \text{ or } v \text{ is contained in some odd vertex}\}.
\end{aligned}$$

Then, update (y, z) by

$$y(v) := \begin{cases} y(v) - \epsilon & (v \in V_+),\\ y(v) + \epsilon & (v \in V_-),\\ y(v) & (\text{otherwise}), \end{cases}$$

$$z(\mathcal{L}_U) := \begin{cases} z(\mathcal{L}_U) + 2\epsilon & (U \in \Upsilon_+ \cup (\Delta'_- \setminus \Delta_-)),\\ z(\mathcal{L}_U) - 2\epsilon & (U \in \Upsilon_- \cup (\Delta'_+ \setminus \Delta_+)),\\ z(\mathcal{L}_U) + \epsilon & (U \in (\Delta_+ \setminus \Delta'_+) \cup (\Delta_- \cap \Delta'_-)),\\ z(\mathcal{L}_U) - \epsilon & (U \in (\Delta_+ \cap \Delta'_+) \cup (\Delta_- \setminus \Delta'_-)),\\ z(\mathcal{L}_U) & (\text{otherwise}), \end{cases}$$

where $\epsilon \geq 0$ is the maximum value maintaining (4)–(7). That is, ϵ is the minimum of the following:

$\epsilon_1 = \min\{y(v) \mid v \in V_+\}$; $\quad \epsilon_2 = \min\{z(\mathcal{L}_U)/2 \mid U \in \Upsilon_- \cup (\Delta'_+ \setminus \Delta_+)\}$;

$\epsilon_3 = \min\{z(\mathcal{L}_U) \mid U \in (\Delta_+ \cap \Delta'_+) \cup (\Delta_- \setminus \Delta'_-)\}$; $\quad \epsilon_4 = \min\{w'(e)/2 \mid e \in \hat{E},\ \partial e \subseteq V_+\}$;

$\epsilon_5 = \min\{w'(e) \mid e \in \hat{E}$, one of ∂e belongs to V_+, and the other $V \setminus (V_+ \cup V_-)\}$;

$\epsilon_6 = \min\{w'(e) \mid \partial e \subseteq X$ for some $X \in \Delta_+ \cup \Delta'_+\}$;

$\epsilon_7 = \min\{w'(a) \mid a \in A,\ \partial^- a \in V_+,\ a \notin \gamma(\mathcal{L}_U)$ for any $U \in \Delta_+ \cup \Upsilon_+\}$.

Then, apply one of the following.

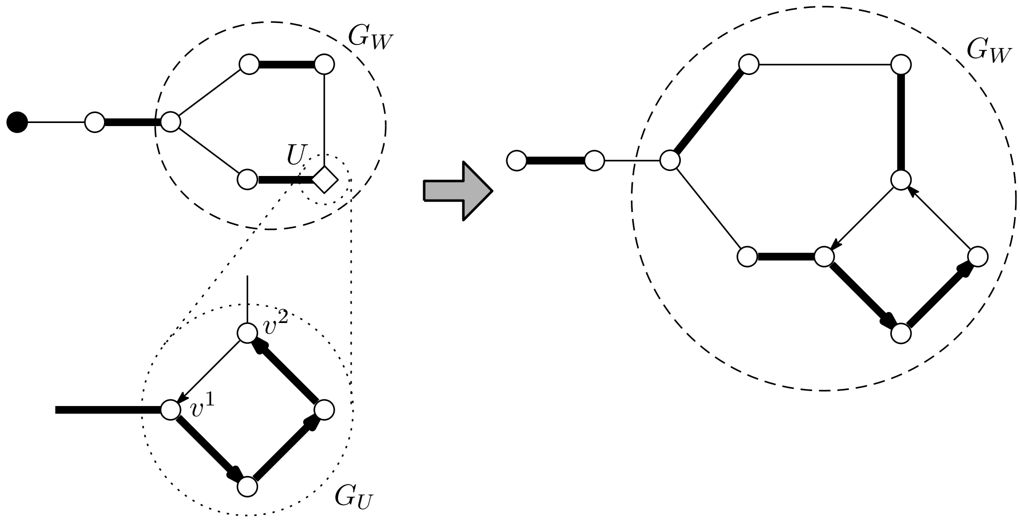

Figure 11: Deshrinking of U and saturation in Case 3.1. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The graph inside the dashed ellipse indicates G_W, and the graph inside the dotted circle indicates G_U. Both G_W and G_U are shrunk in $\hat{G}^\circ$ before the present Dual_Update. After the present Dual_Update, U is deshrunk whereas W is kept shrunk and labelled as saturated.)

Case 1 ($\epsilon = \epsilon_1$): Termination. Deshrink every sets in Ω' and return F.

Case 2 ($\epsilon = \epsilon_2$): Deshrinking. Apply Case 2.1 or 2.2.

Case 2.1 ($\epsilon = z(\mathcal{L}_U)/2$ for some $U \in \Upsilon_-$): Deshrinking. Delete U with $\epsilon = z(\mathcal{L}_U)/2$ from Υ, and then go to Step 3.

Case 2.2 ($\epsilon = z(\mathcal{L}_U)/2$ for some $U \in \Delta'_+ \setminus \Delta_+$): Deshrinking. Denote the maximal set in Υ containing U by W, and the maximal set in Δ containing U by X. Add $\tilde{U} = W \cup X$ to Υ, define a partition $\mathcal{L}_{\tilde{U}}$ of $\tilde{U}$ by $(\mathcal{L}_U \setminus \{U\}) \cup \{X\}$, and set $z(\mathcal{L}_{\tilde{U}}) = 0$. Then, delete U from Δ and go to Step 3.

Case 3 ($\epsilon = \epsilon_3$). Apply Case 3.1 or 3.2.

Case 3.1 ($\epsilon = z(\mathcal{L}_U)$ for some $U \in \Delta_+ \cap \Delta'_+$): Deshrinking and saturation. Denote the set in Υ_+ containing U by W, and the pseudo-vertex containing W by $\hat{W}$. Delete U from Δ, reset $\hat{F} := (\hat{M} \triangle P_{\hat{W}}) \cup \hat{B}$, and label $\hat{W}$ as "saturated." (Note that G_W° has a matching forest covering all vertices in G_W°.) Delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figure 11 for an illustration.

Case 3.2 ($\epsilon = z(\mathcal{L}_U)$ for some $U \in \Delta_- \setminus \Delta'_-$). Let $\hat{U}$ be the vertex in $\hat{V}$ containing U, and let $f_1, f_2 \in H$ be the two edges incident to $\hat{U}$. If f_1 and f_2 are incident to distinct vertices in G_U, then apply Case 3.2.1. Otherwise, apply Case 3.2.2.

Case 3.2.1: Deshrinking and augmentation. Reset $\hat{M} := \hat{M} \triangle P_{\hat{U}}$. Delete U from Δ and each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figure 12 for an illustration.

Case 3.2.2: Deshrinking. Delete U from Δ and then go to Step 3. See Figure 13 for an illustration.

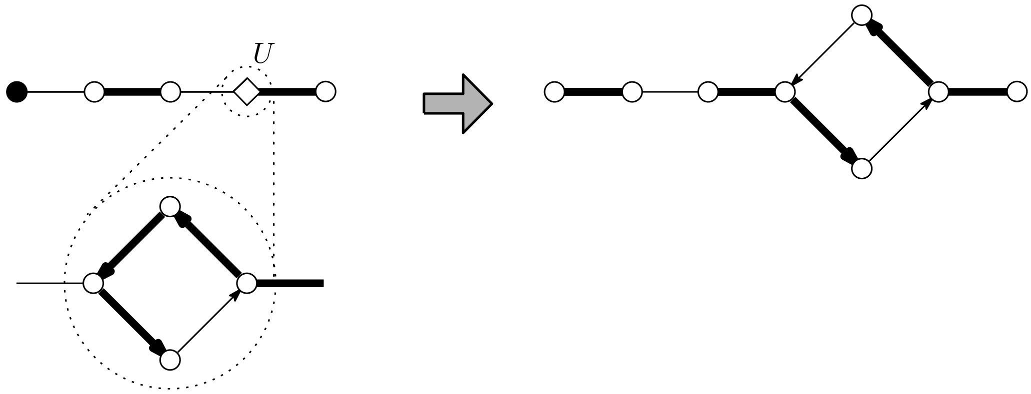

Figure 12: Deshrinking of U and augmentation in Case 3.2.1. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The graph inside the dotted circle indicates G_U.)

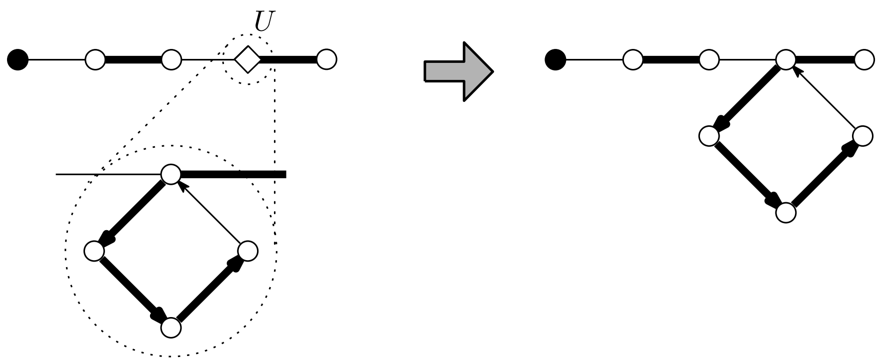

Figure 13: Deshrinking of U in Case 3.2.2. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The graph inside the dotted circle indicates G_U.)

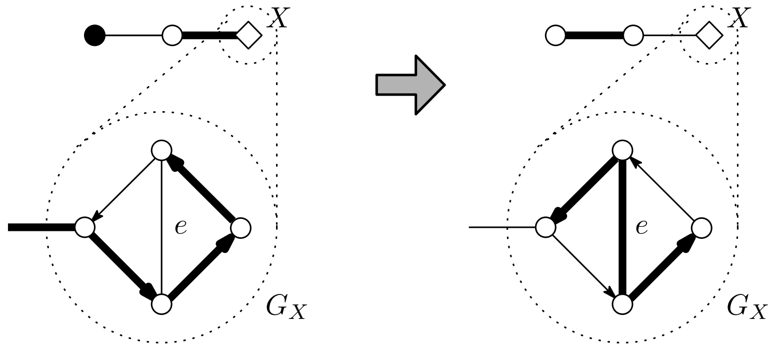

Figure 14: Saturation in Case 6. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The graph inside the dotted circle indicates G_X, and the undirected edge $e \in E$ satisfies that $w'(e) = \epsilon$. After the present Dual_Update, X is kept shrunk and labelled as "saturated.")

Case 4 ($\boldsymbol{\epsilon = \epsilon_4}$). Go to Step 5. (We can execute Step 5.1 or 5.2.)

Case 5 ($\boldsymbol{\epsilon = \epsilon_5}$). Go to Step 5. (We can execute Step 5.3 or 5.4.)

Case 6 ($\boldsymbol{\epsilon = \epsilon_6}$): Saturation. Let $X \subseteq V$ be an element in $\Delta_+ \cup \Delta'_+$ such that contains $e \in E$ with $\epsilon = w'(e)$, and let $\hat{X}$ denote the pseudo-vertex in $\hat{G}$ containing X. Reset $\hat{M} := \hat{M} \triangle P_{\hat{X}}$ and label $\hat{X}$ as "saturated." Delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then go to Step 2. See Figures 14 and 15 for an illustration.

Case 7 ($\boldsymbol{\epsilon = \epsilon_7}$). Apply Case 7.1 or 7.2.

Case 7.1 ($\boldsymbol{\epsilon = w'(a)}$ for some $\boldsymbol{a \in \hat{A}}$). Go to Step 4.

Case 7.2 ($\boldsymbol{\epsilon = w'(a)}$ for some $\boldsymbol{a \in A_U}$ with $\boldsymbol{U \in \Upsilon_+}$): Saturation. Reset $\hat{M} := \hat{M} \triangle P_U$ and label U as "saturated." Delete each $T \in \Omega'$ with $z(\mathcal{L}_T) = 0$ from Ω', and then then go to Step 2. See Figure 16 for an illustration.

3.3 Validity and complexity

In this subsection, we verify Algorithm SIMPLE. Our verification is threefold: check if the feasibility is maintained; check if (8) and (10) are maintained; and prove that (9) is achieved in polynomial time.

3.3.1 Feasibility

It is obvious that the initial primal and dual solutions defined in Step 1 are feasible. Feasibility of the primal solution all through the algorithm is also clear. We check the dual feasibility conditions (4)–(7) after executing Dual_Update.

Condition (6) directly follows from $\epsilon \le \epsilon_1$. Condition (7) also follows from $\epsilon \le \epsilon_2$ and $\epsilon \le \epsilon_3$. Consider Condition (4). If $e \in E$ does not belong to $\gamma(\mathcal{L}_U)$ for any $U \in \Omega'$, then $w'(e)$ decreases in the following two cases.

- $\boldsymbol{\partial e \subseteq V_+}$**.** In this case, $w'(e)$ decreases by 2ϵ. We have that $w'(e) \ge 0$ is maintained since $\epsilon \le \epsilon_4$.

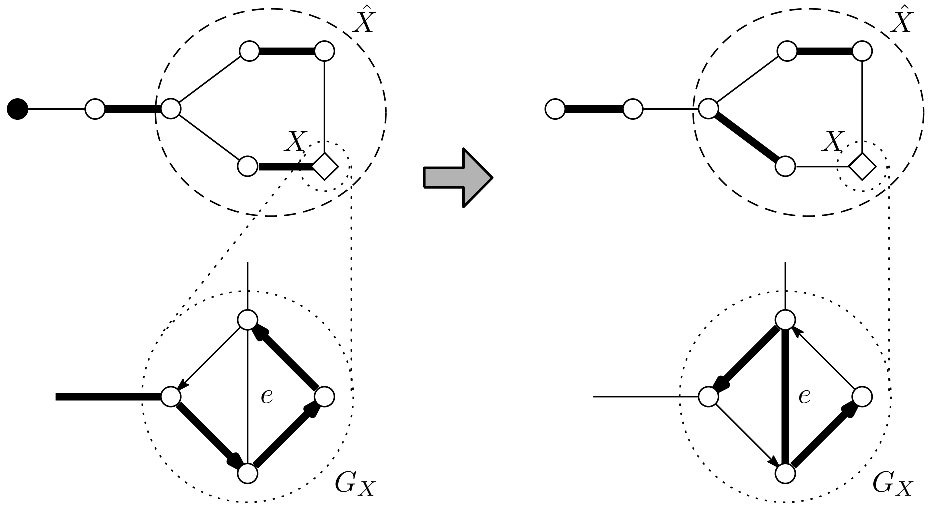

Figure 15: Saturation in Case 6. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The graph inside the dashed ellipse a subgraph shrunk into the pseudo-vertex $\hat{X}$, and that inside the dotted circle indicates G_X. The edge $e \in E$ satisfies that $w'(e) = \epsilon$. Both $G_{\hat{X}}$ and G_X are shrunk in $\hat{G}^\circ$. After the present Dual_Update, $\hat{X}$ and X are kept shrunk and $\hat{X}$ is labelled as "saturated.")

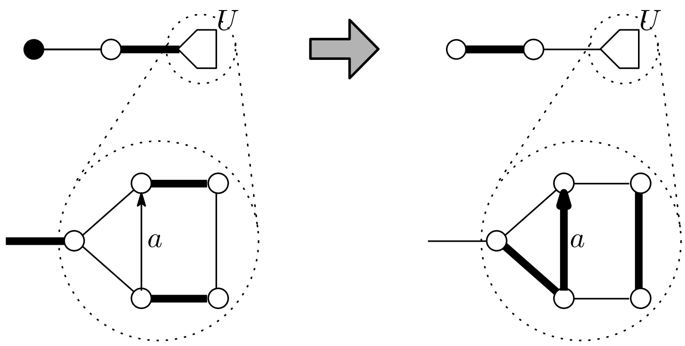

Figure 16: Saturation in Case 7.2. (Thick edges are in $\hat{F}$ and the black vertex is a source vertex. The graph inside the dotted circle indicates G_U. After the present Dual_Update, U is kept shrunk and labelled as "saturated.")

- $\boldsymbol{|\partial e \cap V_+| = 1}$ **and** $\boldsymbol{|\partial e \cap V \setminus (V_+ \cup V_-)| = 1}$**.** In this case, $w'(e)$ decreases by ϵ and $w'(e) \geq 0$ directly follows from $\epsilon \leq \epsilon_5$.

If $e \in E$ belongs to $\gamma(\mathcal{L}_U)$ for some $U \in \Omega'$, we have the following five cases.

- $\boldsymbol{e \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Upsilon_+}$**.** Since y decreases by ϵ at both endpoints of e and $z(\mathcal{L}_U)$ increases by 2ϵ, the only possibility for $w'(e)$ to decrease is that e also belongs to $\gamma(\mathcal{L}_X)$ for some $X \in \Delta'_+$ contained in U. In such a case, $w'(e)$ decreases by ϵ and $w'(e) \geq 0$ is maintained since $\epsilon \leq \epsilon_6$.

- $\boldsymbol{e \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Upsilon_-}$**.** In this case, y increases by ϵ at both endpoints of e and $z(\mathcal{L}_U)$ decreases by 2ϵ. Hence, $w'(e)$ does not change if $e \notin \gamma(\mathcal{L}_X)$ for any $X \in \Delta'_-$ contained in U, and increases by ϵ if $e \in \gamma(\mathcal{L}_X)$ for some $X \in \Delta'_-$ contained in U.

- $\boldsymbol{e \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Delta_+ \setminus \Delta'_+}$**.** In this case, $w'(e)$ decreases by ϵ since y decreases by ϵ at both endpoints of e and $z(\mathcal{L}_U)$ increases by ϵ. Here, $w'(e) \geq 0$ follows from $\epsilon \leq \epsilon_6$.

- $\boldsymbol{e \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Delta_- \setminus \Delta'_-}$**.** In this case, $w'(e)$ increases by ϵ since y increases at both endpoints of e and $z(\mathcal{L}_U)$ decreases by ϵ.

- $\boldsymbol{e \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U}$**, not in** $\boldsymbol{H}$**.** In this case, $w'(e)$ does not change.

Finally, consider Condition (5). If $a \in A$ does not belong to $\gamma(\mathcal{L}_U)$ for $U \in \Omega'$, $w'(a) \geq 0$ follows from $\epsilon \leq \epsilon_7$. If $a \in A$ belongs to $\gamma(\mathcal{L}_U)$ of some $U \in \Omega'$, We have the following five cases.

- $\boldsymbol{a \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Upsilon_+}$**.** In this case, a also belongs to $\gamma(\mathcal{L}_X)$ for some $X \in \Delta'_+$ contained in U. The dual variables $y(\partial^- a)$, $z(\mathcal{L}_U)$ and $z(\mathcal{L}_X)$ change by $-\epsilon$, 2ϵ and $-\epsilon$, respectively, and hence $w'(a)$ does not change.

- $\boldsymbol{a \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Upsilon_-}$**.** In this case, a also belongs to $\gamma(\mathcal{L}_X)$ for some $X \in \Delta'_-$ contained in U. The dual variables $y(\partial^- a)$, $z(\mathcal{L}_U)$ and $z(\mathcal{L}_X)$ change by ϵ, -2ϵ and ϵ, respectively, and hence $w'(a)$ does not change.

- $\boldsymbol{a \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Delta_+ \setminus \Delta'_+}$**.** In this case, $w'(a)$ does not change since $y(\partial^- a)$ decreases by ϵ and $z(\mathcal{L}_U)$ increases by ϵ.

- $\boldsymbol{a \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U \in \Delta_- \setminus \Delta'_-}$**.** In this case, $w'(a)$ does not change since $y(\partial^- a)$ increases by ϵ and $z(\mathcal{L}_U)$ decreases by ϵ.

- $\boldsymbol{a \in \gamma(\mathcal{L}_U)}$ **for some** $\boldsymbol{U}$**, not in** $\boldsymbol{H}$**.** In this case, $w'(a)$ does not change.

3.3.2 Complementary slackness conditions

Consider condition (8). Since the edges in F are picked from G°, it suffices to check that (8) is maintained in the Dual_Update. For each edge $e \in \hat{M}$ not shrunk, $w'(e)$ does not change. Also, it can be easily verified that $w'(e)$ does not change for each edge e which has belonged to a shrunk source component or a shrunk odd cycle.

For condition (10). it is not difficult to see that we can deshrink each set $U \in \Omega'$ with $z(\mathcal{L}_U) > 0$ so that $x(\gamma(\mathcal{L})) = \lfloor |\cup \mathcal{L}| - |\mathcal{L}|/2 \rfloor$.

3.3.3 Complexity

First, we show that $\epsilon > 0$ in a Dual_Update, the proof of which implies time complexity of Algorithm SIMPLE.

Proposition 10. *In the procedure* Dual_Update, *it holds that* $\epsilon > 0$.

Proof. **($\epsilon_1 > 0$).** If $\epsilon_1 = 0$, then we have that $S = \emptyset$ and Algorithm SIMPLE should have terminated.

($\epsilon_2, \epsilon_3 > 0$). A vertex set shrunk after the latest augmentation or saturation is contained in an even vertex in $\hat{G}$. Hence, for each $U \in \mathit{\Delta}_- \cup \mathit{\Upsilon}_-$, we have that U was shrunk before the latest augmentation or saturation, at which $z(\mathcal{L}_U) > 0$. Also, for $U \in \mathit{\Delta}'_+$, we have that $z(\mathcal{L}_U) > 0$ by the bifurcation rule between Steps 5.2.1 and 5.2.2. If $z(\mathcal{L}_U)$ hits 0 for such U, then U is deshrunk.

($\epsilon_4 > 0$). If there exists $e \in \hat{E}$ such that $w'(e) = 0$ and $\partial e \subseteq \hat{V}_+$, then we do not execute Dual_Update but Step 5.1 or 5.2.

($\epsilon_5 > 0$). If $w'(e) = 0$ for $e \in \hat{E}$ such that one of ∂e belongs to $\hat{V}_+$ and the other in $\hat{V} \setminus (\hat{V}_+ \cup \hat{V}_-)$, then we do not execute Dual_Update but Step 5.3, 5.4 or 5.5.

($\epsilon_6 > 0$). Suppose $w'(e) = 0$ for $e \in E$ such that $\partial e \subseteq X$ for some $X \in \mathit{\Delta}_+ \cup \mathit{\Delta}'_+$. Denote the minimal set in $\mathit{\Delta}$ containing ∂e by X_e. When X_e is added to $\mathit{\Delta}$, we have that $w'(e) > 0$ by the bifurcation rule between Steps 4.2.1 and 4.2.2. If $w'(e)$ hit zero for such e, then we should have executed saturation of the pseudo-vertex containing e.

($\epsilon_7 > 0$). If $w'(a) = 0$ for $a \in A$ attaining ϵ_7, then we execute Step 4 or saturation of the pseudo-vertex containing a.

□

We now discuss the time complexity of Algorithm SIMPLE. The bottleneck part is Dual_Update. It follows from the proof for Proposition 10 that Dual_Update is executed $\mathrm{O}(n)$ times between consecutive augmentations or saturations. Since augmentations and saturations collectively happen at most n times and Dual_Update takes $\mathrm{O}(m)$ time for determining ϵ, the total complexity is $\mathrm{O}(n^2 m)$.

Theorem 11. *Algorithm SIMPLE finds a maximum-weight matching forest in* $\mathrm{O}(n^2 m)$ *time.*

3.4 Remarks

We close this section by noting a property of the matching forests maintained in Algorithm SIMPLE: a matching forest appearing at any stage of Algorithm SIMPLE has the maximum weight among all matching forests with the same root-size.

Theorem 12. *Let F be a matching forest which we have at any stage of Algorithm SIMPLE. (Deshrink each element in $\mathit{\Omega}'$, if $\mathit{\Omega} \neq \emptyset$.) Then, it holds that $w(F) \geq w(F')$ for any matching forest F' with $|R(F')| = |R(F)|$.*

Proof. Let F be a matching forest obtained at an arbitrary stage of Algorithm SIMPLE and let (y, z) be the dual solution at that stage. For any matching forest F', it holds that

$$w(F') \leq y(V \setminus R(F')) + \sum_{\mathcal{L} \in \Lambda} z(\mathcal{L})|\gamma(\mathcal{L}) \cap F| \leq y(V \setminus R(F')) + \sum_{\mathcal{L} \in \Lambda} z(\mathcal{L})\lfloor|\cup \mathcal{L}| - |\mathcal{L}/2|\rfloor.$$

Especially for the matching forest F, conditions (8) and (10) imply that

$$w(F) = y(V \setminus R(F)) + \sum_{\mathcal{L} \in \Lambda} z(\mathcal{L})\lfloor|\cup \mathcal{L}| - |\mathcal{L}/2|\rfloor.$$

Here, for $v \in R(F)$, we have that $y(v) \leq y(v')$ for every $v' \in V$ since the values of $y \in \mathbf{R}^V$ are identical in Step 1 and $y(v)$ is decreased at each Dual_Update for $v \in R(F)$. Thus, $y(V \setminus R(F)) \geq y(V \setminus R(F'))$ if $|R(F)| = |R(F')|$. Therefore, $w(F) \geq w(F')$ follows. □

4 A faster algorithm

In this section, we present an $\mathrm{O}(n^3)$ algorithm for the weighted matching forest problem by incorporating Gabow's technique for weighted matching [13] into Giles' weighted matching forest algorithm [15]. The difference from the algorithm in § 3 is that we do not maintain the equality subgraph G° explicitly. Instead, we keep the following.

- For each pair Y, Z of disjoint sets in Ω, we keep an edge $e_{YZ} \in E$ connecting Y and Z and minimizing w'. We keep e_{YZ} as lists: for each $Y \in \Omega$, we have a list containing the e_{YZ}. Moreover, for each $Y \in \Omega$, we keep an edge e_Y with $e_Y = e_{YZ}$ for some $Z \in \Omega$ contained in an even (pseudo-)vertex in H and with $w'(e_{YZ})$ minimal. Similarly, for each pair Y, Z of disjoint sets in Ω, we keep an arc $a_{YZ} \in A$ from Y to Z minimizing w'. We keep a_{YZ} as lists: for each $Z \in \Omega$, we have a list containing the a_{YZ}. Moreover, for each $Z \in \Omega$, we keep an arc a_Z with $a_Z = a_{YZ}$ for some $Y \in \Omega$ and with $w'(a_{YZ})$ minimal.
- For each $X \in \Delta$, we keep an edge $f_X \in E_X$ minimizing w'. Also, we associate a graph G'_X, which is initially the directed cycle shrunk when X is added to Δ.
- For each $U \in \Upsilon$, we keep an arc $b_U \in A_U$ minimizing w'. We also associate graph G'_U, which is initially the odd undirected cycle shrunk when U is added to Υ.

The algorithm is described below.

Algorithm FAST

Initialization. Set $F := \emptyset$, $y(v) := \max\{\max\{w(e)/2 \mid e \in E\}, \max\{w(a) \mid a \in A\}\}$ for every $v \in V$, $\Delta := \emptyset$ and $\Upsilon := \emptyset$. (Hence $\Omega = \{\{v\} \mid v \in V\}$, $\hat{F} = \emptyset$ and z is void.) Moreover, set $H = \emptyset$. The e_{YZ}, e_Y, a_{YZ} and a_Z and are set easily.

Iteration. Reset (y, z) as described in Procedure Dual_Update in Algorithm SIMPLE. After that, at least one of the following cases applies.

Case 1 ($w'(a_U) = 0$ for some maximal set $U \in \Omega$ in even(H)). Denote $B' := \hat{B} \cup \{a_U\}$.

Case 1.1 (B' is a branching): Augmentation. Let $M' := M \triangle P_U$. Reset $\hat{M} := M'$, $\hat{B} := B'$ and $H := M'$, and update the e_Y.

Case 1.2 (B' contains a directed cycle). Let D be the directed cycle in B' and $X \subseteq V$ be the union of the vertices in D. Let $W_1, \ldots, W_l$ be the maximal proper subsets of X belonging to Υ. If, for some $i \in \{1, \ldots, l\}$, $\hat{A}D$ contains a pair of arcs $f^+ \in \delta^+ W_i$ and $f^- \in \delta^- W_i$ such that $\partial^+ f^+$ and $\partial^- f^-$ belong to distinct vertices in G'_{W_i}, apply Case 1.2.1. Otherwise, apply Case 1.2.2.

Case 1.2.1: Augmentation. Reset $\hat{F} := M' \cup B'$, where $M' := \hat{M} \triangle P_U$ and $B' := \hat{B} \cup \{a_U\}$, and $H := M'$, and update the e_Y.

Case 1.2.2: Shrinking. For each $i = 1, \ldots, l$ let $v_{W_i} \in V_{W_i}$ denote the unique vertex in G_{W_i} to which arcs in $\hat{A}D$ are incident. Let $X' = (X \setminus (W_1 \cup \cdots \cup W_l)) \cup \{v_{W_i}\} \cup \cdots \cup \{v_{W_l}\}$ and add X' to Δ. Let $\mathcal{L}_{X'} = \{X'\}$, set $z(\mathcal{L}_{X'}) := 0$ and determine f_X. Then, update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 2 ($w'(e_U) = 0$ for some maximum set $U \in \Omega$ in even(H)). Denote the other endpoint of e_U by W, which also belongs to even(H). Apply Case 2.1 or 2.2.

Case 2.1 (P_U and P_W are disjoint): Augmentation. Reset $\hat{M} := M' = \hat{M} \triangle (P_U \cup \{e\} \cup P_W)$, $H := M'$, and update the e_Y.

Case 2.2 (P_U and P_W intersect). Let C be the cycle in $H \cup \{e_U\}$ and $U \subseteq V$ be the union of the vertices in C. Denote the maximal proper subsets of U belonging to Δ and Υ by $Y_1, \ldots, Y_k \in \Delta$ and $W_1 \ldots, W_l \in \Upsilon$. Let C_U be an odd cycle in G_U obtained by adding even number of edges from each G'_{W_j} to C. For $i = 1, \ldots, k$, let $f_i^1, f_i^2 \in C_U$ denote the two edges incident to Y_i, and let $v_i^1, v_i^2 \in Y_i$ denote the vertices to which f_i^1 and f_i^2 are incident, respectively. If, for some Y_i, the two vertices v_i^1 and v_i^2 are distinct and $z(\mathcal{L}_{Y'_i}) = 0$ for the minimal subset $Y'_i \in \Delta$ of Y_i such that $\{v_i^1, v_i^2\} \subseteq Y'_i$, then apply Case 2.2.1. Otherwise, go to apply Case 2.2.2.

Case 2.2.1: Deshrinking and augmentation. Delete Y'_i from Δ and reset

$$\hat{M} := \begin{cases} (\hat{M} \triangle P_{Y_i}) \triangle C & (f_i^1, f_i^2 \in E \setminus M), \\ \hat{M} \triangle P^*_{Y_i} & \text{(otherwise)}. \end{cases}$$

Here, $P^*_{Y_i}$ denotes the path in $H \cup \{e\}$ from $\hat{S}$ to Y_i consisting of odd number of edges. Reset $H := M'$, and update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 2.2.2: Shrinking. Without loss of generality, assume that v_i^1 and v_i^2 are identical for $i = 1, \ldots, j$, and distinct for $i = j+1, \ldots, k$. For $i = j+1, \ldots, k$, let $Y'_i \in \Delta$ be the minimal subset of Y_i such that $\{v_i^1, v_i^2\} \subseteq Y'_i$. Let $U' = (U \setminus (Y_1 \cup \cdots \cup Y_k)) \cup \{v_{Y_1}, \ldots, v_{Y_j}\} \cup (Y'_{j+1} \cup \cdots Y'_k)$. We now add U' to Υ, and define a partition $\mathcal{L}_{U'}$ of U' by the collection of $\{v_{Y_1}\}, \ldots, \{v_{Y_j}\}, Y'_{j+1}, \ldots, Y'_k$ and singletons of the other vertices in U'. Set $z(\mathcal{L}_{U'}) = 0$, determine $b_{U'}$ and update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 3 ($w'(e_U) = 0$ for some $U \in$ free(H)). Apply Case 3.1 or 3.2.

Case 3.1 ($U \in \partial^- a$ for some $a \in \hat{B}$ or U saturated): Augmentation. Denote the endpoint of e_U other than U by W. Reset $\hat{M} := M' = (\hat{M} \triangle P_W) \cup \{e_U\}$, $\hat{B} := \hat{B} \setminus \{a\}$, $H := M'$, and update the e_Y. If U is a saturated pseudo-vertex, unlabel U.

Case 3.2 ($\boldsymbol{U \in \hat{V} \setminus \partial^- \hat{B}}$): Forest extension. Add e_U to H and update the e_Y.

Case 4 ($\boldsymbol{z(\mathcal{L}_U) = 0}$ for some $\boldsymbol{U \in \Upsilon_-}$): Deshrinking. Delete U from Υ. Let $u \in \hat{V}$ be the vertex covered by $H \setminus \hat{M}$ and $v \in \hat{V}$ be the one covered by $\hat{M}$. Let P be the even-length u-v path in G'_U and N be the matching in G'_U covering all vertices in the odd cycle other than v. Reset $H := H \cup P \cup N$ and $\hat{M} := \hat{M} \cup N$, and update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 5 ($\boldsymbol{z(\mathcal{L}_U) = 0}$ for some $\boldsymbol{U \in \Delta'_+ \cup (\Delta_- \setminus \Delta'_-)}$): Deshrinking. Apply Case 5.1, 5.2 or 5.3.

Case 5.1 ($\boldsymbol{U \in \Delta_- \setminus \Delta'_-}$). Let $u \in V_U$ be the vertex covered by $H \setminus \hat{M}$ and v be the one covered by $\hat{M}$. If u and v are distinct, then apply Case 5.1.1. Otherwise, apply Case 5.1.2.

Case 5.1.1: Deshrinking and augmentation. Delete U from Δ, reset $\hat{M} := M' = \hat{M} \triangle P_U$, $H := M'$, and update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 5.1.2: Deshrinking. Delete U from Δ, and update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 5.2 ($\boldsymbol{U \in \Delta'_+ \setminus \Delta_+}$): Deshrinking. Denote the set in Υ_+ containing U by W, and the maximal set in Δ containing U by X. Delete U from Δ, add $\tilde{U} = W \cup X$ to Υ, define a partition $\mathcal{L}_{\tilde{U}}$ of $\tilde{U}$ by $(\mathcal{L}_U \setminus \{U\}) \cup \{X\}$, and set $z(\mathcal{L}_{\tilde{U}}) = 0$. Then, update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 5.3 ($\boldsymbol{U \in \Delta_+ \cap \Delta'_+}$): Deshrinking and saturation. Denote the set in Υ_+ containing U by W, and the pseudo-vertex containing W by $\hat{W}$. Delete U from Δ, reset $\hat{M} := M' = (\hat{M} \triangle P_{\hat{W}})$, and label $\hat{W}$ as "saturated." Reset $H := M'$ and update the e_{YZ}, e_Y, a_{YZ} and a_Z.

Case 6 ($\boldsymbol{w'(f_X) = 0}$ for some $\boldsymbol{X \in \Delta_+ \cup \Delta'_+}$): Saturation. Let $\hat{X}$ denote the pseudo-vertex in $\hat{G}$ containing X. Add f_X to G'_X. Reset $\hat{M} := M' = \hat{M} \triangle P_{\hat{X}}$ and label $\hat{X}$ as "saturated." Reset $H := M'$ and update the e_Y.

Case 7 ($\boldsymbol{w'(b_U) = 0}$ for some $\boldsymbol{U \in \Upsilon_+}$): Saturation. Let $\hat{U}$ denote the pseudo-vertex containing U. Add b_U to G'_U. Reset $\hat{M} := M' = M \triangle P_{\hat{U}}$ and label $\hat{U}$ as "saturated." Reset $H := M'$ and update the e_Y.

The validity of Algorithm FAST can be verified just as we did for Algorithm SIMPLE. Let us mention here the complexity. In Algorithm FAST, Procedure Dual_Update can be executed in $\mathrm{O}(n)$ time, by scanning the e_Y, a_Z, f_X and b_U, together with the $z(\mathcal{L}_U)$, instead of scanning all edges in $E \cup A$. Updating the lists after shrinking, deshrinking or forest extension takes $\mathrm{O}(n)$ time and updating after augmentation and saturation takes $\mathrm{O}(n^2)$ time. Thus, the total time complexity $\mathrm{O}(n^3)$.

Theorem 13. *Algorithm FAST finds a maximum-weight matching forest in* $\mathrm{O}(n^3)$ *time.*

We remark that we do not incorporate the branching delta-matroid $\hat{F}^\circ_B$, which is used in Algorithm SIMPLE. This is because we need to decompose the equality subgraph $(\hat{V}, \hat{A}^\circ)$ into strong components in order to determine whether $R(\hat{B}) \setminus \{v\} \in \hat{\mathcal{F}}^\circ_B$, which takes $\mathrm{O}(m)$ time.

Acknowledgements

The author is thankful to Satoru Iwata for valuable comments on this research and the simple proof for the matching forest delta-matroid. The author also thanks Yusuke Kobayashi for discussions on simultaneous delta-matroids.

References

[1] A. Bouchet: Greedy algorithm and symmetric matroids, *Mathematical Programming*, 38 (1987), 147–159.

[2] A. Bouchet: Matchings and Δ-matroids, *Discrete Applied Mathematics*, 24 (1989), 55–62.

[3] R. Chandrasekaran and S.N. Kabadi: Pseudomatroids, *Discrete Mathematics*, 71 (1988), 205–217.

[4] W.H. Cunningham and A.B. Marsh, III: A primal algorithm for optimum matching, *Mathematical Programming Study*, 8 (1978), 50–72.

[5] A.W.M. Dress and T. Havel: Some combinatorial properties of discriminants in metric vector spaces, *Advances in Mathematics*, 62 (1986), 285–312.

[6] A.W.M. Dress and W. Wenzel: A greedy-algorithm characterization of valuated Δ-matroids, *Applied Mathematics Letters*, 4 (1991), 55–58.

[7] J. Edmonds: Maximum matching and a polyhedron with 0,1-vertices, *Journal of Research National Bureau of Standards*, *Section B*, 69 (1965), 125–130.

[8] J. Edmonds: Paths, trees, and flowers, *Canadian Journal of Mathematics*, 17 (1965), 449–467.

[9] J. Edmonds: Optimum branchings, *Journal of Research National Bureau of Standards*, *Section B*, 71 (1967), 233–240.

[10] J. Edmonds and R. Giles: A min-max relation for submodular functions on graphs, *Annals of Discrete Mathematics*, 1 (1977), 185–204.

[11] A. Frank: Covering branchings, *Acta Scientiarum Mathematicarum [Szeged]*, 41 (1979), 77–81.

[12] A. Frank and E. Tardos: Generalized polymatroids and submodular flows, *Mathematical Programming*, 42 (1988), 489–563.

[13] H.N. Gabow: *Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs*, Ph.D. thesis, Stanford University, 1973.

[14] R. Giles: Optimum matching forests I: Special weights, *Mathematical Programming*, 22 (1982), 1–11.

[15] R. Giles: Optimum matching forests II: General weights, *Mathematical Programming*, 22 (1982), 12–38.

[16] R. Giles: Optimum matching forests III: Facets of matching forest polyhedra, *Mathematical Programming*, 22 (1982), 39–51.

[17] J. Keijsper: A Vizing-type theorem for matching forests, *Discrete Mathematics*, 260 (2003), 211–216.

[18] L. Lovász: Matroid matching and some applications, *Journal of Combinatorial Theory, Series B*, 28 (1980), 208–236.

[19] K. Murota: Characterizing a valuated delta-matroid as a family of delta-matroids, *Journal of the Operations Research Society of Japan*, 40 (1997), 565–578.

[20] K. Murota: *Discrete Convex Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[21] A. Schrijver: Total dual integrality of matching forest constraints, *Combinatorica*, 20 (2000), 575–588.

[22] A. Schrijver: *Combinatorial Optimization—Polyhedra and Efficiency*, Springer-Verlag, Heidelberg, 2003.

[23] V.G. Vizing: Ob otsenke khromaticheskogo klassa p-grapha (in Russian), *Diskretnyĭ*, 3 (1964), 25–30.

[24] V.G. Vizing: Khromaticheskiĭ klass mul'tigrafa (in Russian), *Kibernetika*, 1965:3 (1965), 29–39.

[25] W. Wenzel: Δ-matroids with the strong exchange conditions, *Applied Mathematics Letters*, 6 (1993), 67–70.

[26] W. Wenzel: Pfaffian forms and Δ-matroids, *Discrete Mathematics*, 115 (1993), 253–266.