# Stable Matchings in Trees

Satoshi Tayu(✉) and Shuichi Ueno

Department of Information and Communications Engineering,
Tokyo Institute of Technology, S3-57, Tokyo 152-8550, Japan
tayu@eda.ict.e.titech.ac.jp

**Abstract.** The maximum stable matching problem (Max-SMP) and the minimum stable matching problem (Min-SMP) have been known to be NP-hard for subcubic bipartite graphs, while Max-SMP can be solved in polynomal time for a bipartite graph $G$ with a bipartition $(X, Y)$ such that $\deg_G(v) \leq 2$ for any $v \in X$. This paper shows that both Max-SMP and Min-SMP can be solved in linear time for trees. This is the first polynomially solvable case for Min-SMP, as far as the authors know. We also consider some extensions to the case when $G$ is a general/bipartite graph with edge weights.

## 1 Introduction

Let $G$ be a simple bipartite graph (bigraph) with vertex set $V(G)$ and edge set $E(G)$. For each vertex $v \in V(G)$, let $I_G(v)$ be the set of all edges incident with $v$, and $\deg_G(v) = |I_G(v)|$. For each $v \in V(G)$, $\preceq_v$ is a total preorder (a binary relation with transitivity, totality, and hence reflexivity) on $I(v)$, and $\preceq_G = \{\preceq_v | \ v \in V(G)\}$. A total preorder $\preceq_v$ is said to be *strict* if $e \preceq_v f$ and $e \neq f$ imply $f \not\preceq_v e$. We say that $\preceq_G$ is *strict* if $\preceq_v$ is strict for every $v \in V(G)$. It should be noted that a strict total preorder is just a linear order. A pair $(G, \preceq_G)$ is called a *preference system*. A preference system $(G, \preceq_G)$ is said to be *strict* if $\preceq_G$ is strict. We say that an edge $e$ *dominates* $f$ at vertex $v$ if $e \preceq_v f$. A matching $M$ of $G$ is said to be *stable* if each edge of $G$ is dominated by some edge in $M$. The stable matching problem (SMP) is to find a stable matching of a preference system $(G, \preceq_G)$. It is well-known that any preference system $(G, \preceq_G)$ has a stable matching, and SMP can be solved in linear time by using the Gale/Shapley algorithm [3]. It is also well-known that every stable matching for a strict preference system has the same size and spans the same set of vertices, while a general preference system can have stable matchings of different sizes [3]. This leads us to the following two problems. The maximum stable matching problem (Max-SMP) is to find a stable matching with the maximum cardinality, and the minimum stable matching problem (Min-SMP) is to find a stable matching with the minimum cardinality. Manlove, Irving, Iwama, Miyazaki, and Morita showed that Max-SMP and Min-SMP are both NP-hard [9].

Let $(X, Y)$ be a bipartition of a bigraph $G$. A bigraph $G$ is called a $(p, q)$-*graph* if $\deg_x(\leq)p$ for every $x \in X$, and $\deg_y(\leq)q$ for every $y \in Y$. Irving, Manlove, and O'Malley showed that Max-SMP is NP-hard even for $(3, 3)$-graphs, while


Y. Cao and J. Chen (Eds.): COCOON 2017, LNCS 10392, pp. 492–503, 2017.
DOI: 10.1007/978-3-319-62389-4_41

Max-SMP can be solved in polynomial time for $(2, \infty)$-graphs [7]. Some indepth consideration on the approximation for both problems can be found in [5].

The purpose of the paper is to show that Max-SMP and Min-SMP can be solved in linear time if $G$ is a tree. This is the first polynomially solvable case for Min-SMP, as far as the authors know. We also consider some extensions to the case when $G$ is a general/bipartite graph with edge weights.

The rest of the paper is organized as follows. Section 2 gives a foundation for our algorithms. Section 3.1 shows a linear time algorithm based on a dynamic programming to compute the size of a maximum stable matching in a tree. Section 3.2 shows a linear time algorithm, a modification of the algorithm in Sect. 3.1, to compute a maximum stable matching in a tree. Section 3.3 mentions an extension of the algorithm in Sect. 3.2 to compute a maximum-weight stable matching in linear time for trees with edge weights. Section 4.1 mentions that minimum stable matchings can be computed in linear time for trees (with edge weights) by modifying algorithms in Sect. 3. Section 4.2 mentions some extensions of our results to the case when $G$ is a general graph with edge weights.

## 2 Stable Matchings in Trees

We need preliminaries to describe our algorithms.

Let $T$ be a tree, and $(T, \preceq_T)$ be a preference system, which is called a tree preference system. A stable matching of $(T, \preceq_T)$ is called a *stable matching* of $T$, for simplicity. We use the following notations:

- we write $u \preceq_v w$ (or $w \succeq_v u$) if $(v, u) \preceq_v (v, w)$,
- we write $u \equiv_v w$ if $(v, u) \preceq_v (v, w)$ and $(v, w) \preceq_v (v, u)$,
- we write $u \prec_v w$ (or $w \succ_v u$) if $u \preceq_v w$ and $u \not\equiv_v w$.

It should be noted that if $\preceq_v$ is strict, then $u \equiv_v w$ if and only if $u = w$. It should be also noted that $\equiv_v$ is an equivalence relation on $I_G(v)$.

We consider $T$ as a rooted tree with the root $r$, which is a leaf (a vertex of degree one) of $T$. For each vertex $v \in V(T) - \{r\}$, $p(v)$ is the parent of $v$, and $D(v)$ is the set of descendants of $v$. For any $v \in V(T) - \{r\}$, we denote by $T(v)$ the subtree induced by $D(v) \cup \{p(v)\}$. A matching $M$ of $T(v)$ is said to be $v$-*stable* if every edge of $E(T(v)) - \{(v, p(v))\}$ is dominated by some edge in $M$. A vertex $v$ is said to be *matched* with $u$ in $M$ if $(u, v) \in M$.

We define five sets of $v$-stable matchings of $T(v)$ as follows.

- $\mathcal{M}_v^P$ is the set of $v$-stable matchings of $T(v)$ in which $v$ is matched with $p(v)$.
- $\mathcal{M}_v^H$ is the set of $v$-stable matchings of $T(v)$ in which $v$ is matched with a child $c$ such that $c \preceq_v p(v)$.
- $\mathcal{M}_v^L$ is the set of $v$-stable matchings of $T(v)$ in which $v$ is matched with a child $c$ such that $c \succ_v p(v)$.
- $\mathcal{M}_v^F$ is the set of $v$-stable matchings of $T(v)$ in which $v$ is matched with no other vertices of $T(v)$.
- $\mathcal{M}_v^{\overline{P}}$ is the set of $v$-stable matchings of $T(v)$ in which $v$ is not matched with $p(v)$.

If $v(\neq r)$ is a leaf, $T(v)$ is a tree with $E(T(v)) = \{(v,p(v))\}$, and we have $E(T(v)) - \{(v,p(v))\} = \emptyset$. Thus, we have the following.

**Lemma 1.** *If $v(\neq r)$ is a leaf, then $\mathcal{M}_v^P = \{\{(v,p(v))\}\}$, $\mathcal{M}_v^H = \mathcal{M}_v^L = \emptyset$, and $\mathcal{M}_v^{\overline{P}} = \mathcal{M}_v^F = \emptyset$.* □

It should be noted that for any $v \in V(T) - \{r\}$, $\mathcal{M}_v^{\overline{P}} = \mathcal{M}_v^H \cup \mathcal{M}_v^L \cup \mathcal{M}_v^F$, $\mathcal{M}_v^{\overline{P}} \cap \mathcal{M}_v^P = \emptyset$, and every $v$-stable matching of $T(v)$ is in $\mathcal{M}_v^P \cup \mathcal{M}_v^{\overline{P}}$.

Let $r'$ be the child of $r$. Since $r'$ is the only child of $r$, we obtain the following.

**Lemma 2.** *A set $M \subseteq E(T)$ is a stable matching of $T$ if and only if $M \in \mathcal{M}_{r'}^P \cup \mathcal{M}_{r'}^H$.*

*Proof.* Suppose $M$ is a stable matching of $T = T(r')$. Since $M$ is an $r'$-stable matching, $M \in \mathcal{M}_{r'}^S$ for some $S \in \{P, H, L, F\}$. If $(r', r) \in M$ then $M \in \mathcal{M}_{r'}^P$. If $(r', r) \notin M$ then $(r', r)$ must be dominated by an edge in $M$, and thus, there is a child $c$ of $r'$ such that $(r', c) \in M$ and $c \preceq_{r'} r$, which means that $M \in \mathcal{M}_{r'}^H$. Therefore, we conclude that $M \in \mathcal{M}_{r'}^P \cup \mathcal{M}_{r'}^H$.

Conversely, suppose $M \in \mathcal{M}_{r'}^P \cup \mathcal{M}_{r'}^H$. Since $M$ is an $r'$-stable matching of $T = T(r')$, every edge in $E(T(v)) - \{(r', r)\}$ is dominated by an edge in $M$. If $M \in \mathcal{M}_{r'}^P$ then $(r', r) \in M$. If $M \in \mathcal{M}_{r'}^H$, then there exists a child $c$ of $r'$ such that $(r', c) \in M$ and $c \preceq_{r'} r$, which means that $(r', r)$ is dominated by $(r', c) \in M$. Thus, we conclude that $M$ is a stable matching of $T$, and we have the lemma. □

For a vertex $v \in V(T)$, let $\mathcal{C}(v)$ be the set of children of $v$. For a set $M \subseteq E(T)$ and $v \in V(T) - \{r\}$, we define $M(v) = E(T(v)) \cap M$.

**Lemma 3.** *If $M$ is a $v$-stable matching of $T(v)$ then $M(c)$ is a $c$-stable matching of $T(c)$ for any $c \in \mathcal{C}(v)$.*

*Proof.* Since $M$ is a matching of $T(v)$, $M(c)$ is a matching of $T(c)$. Since $M$ is a $v$-stable matching of $T(v)$, every edge in $E(T(c)) - \{(c, v)\}$ is dominated by an edge in $M(c)$. Thus, $M(c)$ is a $c$-stable matching of $T(c)$. □

**Lemma 4.** *For any $v \in V(T) - \{r\}$ and set $M \subseteq E(T(v))$, $M \in \mathcal{M}_v^P$ if and only if the following conditions are satisfied:*

*(i)* $(v, p(v)) \in M$,
*(ii)* $M(c) \in \mathcal{M}_c^{\overline{P}}$ *for every* $c \in \mathcal{C}(v)$, *and*
*(iii)* $M(c) \in \mathcal{M}_c^H$ *for every* $c \in \mathcal{C}(v)$ *with* $c \prec_v p(v)$.

*Proof.* Suppose $M \in \mathcal{M}_v^P$. Then, (i) follows from the definition of $\mathcal{M}_v^P$. Since $M$ is a $v$-stable matching of $T(v)$, $M(c)$ is a $c$-stable matching of $T(c)$ for every $c \in \mathcal{C}(v)$ by Lemma 3. Since $v$ is matched with $p(v)$, $(v, c) \notin M$ for any $c \in \mathcal{C}(v)$, that is, $M(c) \in \mathcal{M}_c^{\overline{P}}$. Thus, we have (ii). For any $c \in \mathcal{C}(v)$ with $c \prec_v p(v)$, $(v, c)$ is not dominated by $(v, p(v))$. Thus, there exists $g \in \mathcal{C}(c)$ such that $(c, g)$ dominates $(c, v) = (v, c)$. Since $g \preceq_c v$, $M(c) \in \mathcal{M}_c^H$, and we have (iii).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i), (ii), and (iii). For any $c \in \mathcal{C}(v)$, $M(c)$ is a matching such that $v$ is matched with no other vertex by (ii). Thus, $M$ is a matching of $T(v)$. For any $c \in \mathcal{C}(v)$, each edge of $E(T(c)) - \{(v, c)\}$ is dominated by an edge in $M$, since $M(c)$ is a $c$-stable matching by (i), (ii), and (iii). For any $c \in \mathcal{C}(v)$, if $c \succeq_v p(v)$ then $(v, c)$ is dominated by $(v, p(v))$, which is in $M$ by (i). If $c \prec_v p(v)$ then $(v, c)$ is dominated by an edge in $M$ by (iii). Thus, $M$ is a $v$-stable matching of $T(v)$, and we conclude that $M \in \mathcal{M}_v^P$ by (i). □

**Lemma 5.** *For any $v \in V(T) - \{r\}$ and set $M \subseteq E(T(v))$, $M \in \mathcal{M}_v^H$ if and only if the following conditions are satisfied:*

*(i)* $(v, p(v)) \notin M$, *and*
*(ii) there exists $c' \in \mathcal{C}(v)$ such that the following conditions are satisied:*
- *(ii-1)* $c' \preceq_v p(v)$ ,
- *(ii-2)* $M(c') \in \mathcal{M}_{c'}^P$,
- *(ii-3)* $M(c) \in \mathcal{M}_c^{\overline{P}}$ *for every $c \in \mathcal{C}(v) - \{c'\}$, and*
- *(ii-4)* $M(c) \in \mathcal{M}_c^H$ *for every $c \in \mathcal{C}(v)$ with $c \prec_v c'$.*

*Proof.* Suppose $M \in \mathcal{M}_v^H$. Then, (i) and (ii-1) follow from the definition of $\mathcal{M}_v^H$. Since $M$ is a $v$-stable matching of $T(v)$, $M(c)$ is a $c$-stable matching of $T(c)$ for every $c \in \mathcal{C}(v)$ by Lemma 3. Since $M \in \mathcal{M}_v^H$, there exists $c' \in \mathcal{C}(v)$ such that $(v, c') \in M$ and $c' \preceq_v p(v)$, that is, $M(c') \in \mathcal{M}_{c'}^P$. Thus, we have (ii-2). Since $M$ is a matching and $(v, c') \in M$, $(v, c) \notin M(c)$ for every $c \in \mathcal{C}(v) - \{c'\}$. Thus, $M(c) \in \mathcal{M}_c^{\overline{P}}$ for every $c \in \mathcal{C}(v) - \{c'\}$, and we have (ii-3). For any $c \in \mathcal{C}(v)$ with $c \prec_v c'$, $(v, c)$ is not dominated by $(v, c')$. Therefore, there exists $g \in \mathcal{C}(c)$ such that $(c, g)$ dominates $(c, v)$. Since $g \preceq_c v$, $M(c) \in \mathcal{M}_c^H$, and we have (ii-4).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i) and (ii). For any $c \in \mathcal{C}(v) - \{c'\}$, $M(c)$ is a matching such that $v$ is matched with no other vertex by (ii-3). Also, $M(c')$ is a matching by (ii-2). Thus $M$ is a matching of $T(v)$. For any $c \in \mathcal{C}(v) - \{c'\}$, each edge in $E(T(c)) - \{(v, c)\}$ is dominated by an edge in $M$, since $M(c)$ is a $c$-stable matching by (ii-3). For any $c \in \mathcal{C}(v)$, if $c \succeq_v c'$ then $(v, c)$ is dominated by $(v, c')$, which is in $M$ by (ii-2). If $c \prec_v c'$ then $(v, c)$ is dominated by an edge in $M$ by (ii-4). Thus, $M$ is a $v$-stable matching of $T(v)$ by (i) and (ii), and we conclude that $M \in \mathcal{M}_v^H$ by (ii-1) and (ii-2). □

**Lemma 6.** *For any $v \in V(T) - \{r\}$ and set $M \subseteq E(T(v))$, $M \in \mathcal{M}_v^L$ if and only if the following conditions are satisfied:*

*(i)* $(v, p(v)) \notin M$, *and*
*(ii) there exists $c' \in \mathcal{C}(v)$ such that the following conditions are satisfied:*
- *(ii-1)* $c' \succ_v p(v)$,
- *(ii-2)* $M(c') \in \mathcal{M}_{c'}^L$,
- *(ii-3)* $M(c) \in \mathcal{M}_c^{\overline{P}}$ *for every $c \in \mathcal{C}(v) - \{c'\}$, and*
- *(ii-4)* $M(c) \in \mathcal{M}_c^H$ *for every $c \in \mathcal{C}(v)$ with $c \prec_v c'$.*

*Proof.* Suppose $M \in \mathcal{M}_v^L$. Then, (i) and (ii-1) follow from the definition of $\mathcal{M}_v^L$. Since $M$ is a $v$-stable matching of $T(v)$, $M(c)$ is a $c$-stable matching of $T(c)$ for every $c \in \mathcal{C}(v)$ by Lemma 3. Since $M \in \mathcal{M}_v^L$, there exists $c' \in \mathcal{C}(v)$ such that $(v, c') \in M$ and $c' \succ_v p(v)$, that is, $M(c') \in \mathcal{M}_{c'}^L$. Thus, we have (ii-2). Since $M$ is a matching and $(v, c') \in M$, $(v, c) \notin M(c)$ for every $c \in \mathcal{C}(v) - \{c'\}$. Thus, $M(c) \in \mathcal{M}_c^{\overline{P}}$ for every $c \in \mathcal{C}(v) - \{c'\}$, and we have (ii-3). For any $c \in \mathcal{C}(v)$ with $c \prec_v c'$, $(v, c)$ is not dominated by $(v, c')$. Therefore, there exists $g \in \mathcal{C}(c)$ such that $(c, g)$ dominates $(c, v)$. Since $g \preceq_c v$, $M(c) \in \mathcal{M}_c^H$, and we have (ii-4).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i) and (ii). For any $c \in \mathcal{C}(v) - \{c'\}$, $M(c)$ is a matching such that $v$ is matched with no other vertex by (ii-3). Also, $M(c')$ is a matching by (ii-2). Thus $M$ is a matching of $T(v)$. For any $c \in \mathcal{C}(v) - \{c'\}$, each edge in $E(T(c)) - \{(v, c)\}$ is dominated by an edge in $M$, since $M(c)$ is a $c$-stable matching by (ii-3). For any $c \in \mathcal{C}(v)$, if $c \succeq_v c'$ then $(v, c)$ is dominated by $(v, c')$, which is in $M$ by (ii-2). If $c \prec_v c'$ then $(v, c)$ is dominated by an edge in $M$ by (ii-4). Thus, $M$ is a $v$-stable matching of $T(v)$ by (i) and (ii), and we conclude that $M \in \mathcal{M}_v^L$ by (ii-1) and (ii-2). □

**Lemma 7.** *For any $v \in V(T) - \{r\}$ and set $M \subseteq E(T(v))$, $M \in \mathcal{M}_v^F$ if and only if the following conditions are satisfied:*

*(i) $(v, p(v)) \notin M$ , and*
*(ii) $M(c) \in \mathcal{M}_c^H$ for any $c \in \mathcal{C}(v)$.*

*Proof.* Suppose $M \in \mathcal{M}_v^F$. Then, (i) follows from the definition of $\mathcal{M}_v^F$. Since $M$ is a $v$-stable matching of $T(v)$, $M(c)$ is a $c$-stable matching of $T(c)$ for every $c \in \mathcal{C}(v)$ by Lemma 3. Since $M \in \mathcal{M}_v^F$, for any $c \in \mathcal{C}(v)$, there exists $g \in \mathcal{C}(c)$ such that $(c, g)$ dominates $(c, v)$. Since $g \preceq_c v$, $M(c) \in \mathcal{M}_c^H$ for any $c \in \mathcal{C}(v)$, and we have (ii).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i) and (ii). For any $c \in \mathcal{C}(v)$, $M(c)$ is a matching such that $v$ is matched with no other vertex by (ii). Thus, $M$ is a matching of $T(v)$. For any $c \in \mathcal{C}(v)$, each edge in $E(T(c)) - \{(c, v)\}$ is dominated by an edge in $M$, since $M(c)$ is a $c$-stable matching by (ii). For any $c \in \mathcal{C}(v)$, $(c, v)$ is dominated by an edge in $M$ by (ii). Thus, $M$ is a $v$-stable matching of $T(v)$, and we conclude that $M \in \mathcal{M}_v^F$ by (i). □

## 3 Linear Time Algorithms for Trees

### 3.1 Computing the Size of Maximum Stable Matchings

Now, we are ready to show a linear time algorithm to compute the size of a maximum stable matching for a tree preference system. Our algorithm applies a dynamic programming scheme based on the results in the previous section.

Let $(T, \preceq_T)$ be a tree preference system. We consider $T$ as a rooted tree with root $r$, which is a leaf of $T$. For any $v \in V(T) - \{r\}$ and $S \in \{P, H, L, F, \overline{P}\}$, we define that

$$\mu_v^S = \max_{M \in \mathcal{M}_v^S} |M|. \tag{1}$$

That is, $\mu_v^S$ is the maximum number of edges of a $v$-stable matching in $\mathcal{M}_v^S$. We define that $\mu_v^S = -\infty$ if $\mathcal{M}_v^S = \emptyset$.

From Lemma 1, we have the following.

**Lemma 8.** *If $v(\neq r)$ is a leaf of $T$,*

$$\mu_v^H = \mu_v^L = -\infty, \tag{2}$$
$$\mu_v^P = 1, \text{and} \tag{3}$$
$$\mu_v^F = 0. \qquad \square$$

Define that for any $v \in V(T) - \{r\}$ and $c \in \mathcal{C}(v)$,

$$\mu_{v,c} = \mu_c^P + \sum_{c\in\mathcal{C}(v),\ c'\prec_v c} \mu_{c'}^H + \sum_{c\in\mathcal{C}(v),\ c'\succeq_v c,\ c'\neq c} \mu_{c'}^{\overline{P}}. \tag{4}$$

From Lemmas 4 –7, we have the following.

**Lemma 9.** *For any $v \in V(T) - \{r\}$,*

$$\mu_v^P = \sum_{c\in\mathcal{C}(v),\ c\prec_v p(v)} \mu_c^H + \sum_{c\in\mathcal{C}(v),\ c\succeq_v p(v)} \mu_c^{\overline{P}} + 1, \tag{5}$$
$$\mu_v^H = \max\{\mu_{v,c} \mid c \in \mathcal{C}(v), c \preceq_v p(v)\}, \tag{6}$$
$$\mu_v^L = \max\{\mu_{v,c} \mid c \in \mathcal{C}(v), c \succ_v p(v)\}, \tag{7}$$
$$\mu_v^F = \sum_{c\in\mathcal{C}(v)} \mu_c^H, \text{ and} \tag{8}$$
$$\mu_v^{\overline{P}} = \max\{\mu_v^H, \mu_v^L, \mu_v^F\}. \tag{9}$$

$\square$

From Lemmas 8 and 9, we have the following.

**Lemma 10.** *Procedure* COMP_$\mu(v)$ *shown in Fig. 1 computes $\mu_v^S$ for all $v \in V(T) - \{r\}$ and $S \in \{P, H, L, F, \overline{P}\}$.* $\square$

Now, we are ready to show the following.

**Theorem 1.** *Algorithm* MAX-SIZE$(T, \preceq_T, r)$ *shown in Fig. 2 solves Max-SMP for a tree preference system $(T, \preceq_T)$ in linear time.*

*Proof.* The validity of the algorithm follows from Lemmas 2 and 10. We shall show that the time complexity of the algorithm is $O(n)$, where $n = |V(T)|$.

**Lemma 11.** *Given $\mu_c^S$ for every $c \in \mathcal{C}(v)$ and $S \in \{P, H, L, F, \overline{P}\}$, $\{\mu_{v,c} \mid c \in \mathcal{C}(v)\}$ can be computed in $O(|\mathcal{C}(v)|)$ time,*

```
Input v ∈ V(T) − {r}.
begin
  for all c ∈ C(v)
    Comp_μ(c).
  endfor
  for all c ∈ C(v)
    μ_{v,c} = μ^P_c + Σ_{c'∈C(v), c'≺_v c} μ^H_{c'} + Σ_{c'∈C(v), c'⪰_v c, c'≠c} μ^{\overline{P}}_{c'}.
  endfor
  μ^P_v = Σ_{c∈C(v), c≺_v p(v)} μ^H_c + Σ_{c∈C(v), c⪰_v p(v)} μ^{\overline{P}}_c + 1, where μ^P_v = 1 if v is a leaf.
 (See Lemma 8.)
  μ^F_v = Σ_{c∈C(v)} μ^H_c, where μ^F_v = 0 if v is a leaf.
  μ^H_v = max_{c∈C(v), c⪯_v p(v)} μ_{v,c}, where μ^H_v = −∞ if v is a leaf.
  μ^L_v = max_{c∈C(v), c≺_v p(v)} μ_{v,c}, where μ^L_v = −∞ if v is a leaf.
  if μ^H_v ≥ max{μ^L_v, μ^F_v} then
    μ^{\overline{P}}_v = μ^H_v.
  endif
end
```

**Fig. 1.** Procedure COMP_$\mu(v)$.

*Proof of Lemma 11.* If $v$ is a leaf, $\{\mu_{v,c} \mid c \in \mathcal{C}(v)\}$ can be computed in $O(1)$ time, by definition. Let $v \in V(T)$ be a vertex of degree at least 2, $\delta = |\mathcal{C}(v)|$, and $c_1, c_2, \ldots, c_\delta$ be a sequence of the children of $v$ such that $c_i \preceq_v c_{i+1}$ for any $i \in [\delta - 1]$, where $c_1, c_2, \ldots, c_\delta$ are sorted in the problem instance. Define that for any $v \in V(T) - \{r\}$ and $c \in \mathcal{C}(v)$,

$$[c]_v = \{c' \mid c' \equiv_v c\},$$

which is an equivalence class for an equivalence relation $\equiv_v$ on $I_G(v)$. Then, $\mathcal{C}(v)$ is partitioned into equivalence classes for $\equiv_v$. Let $x \geq 2$. If $c_x \equiv_v c_{x-1}$ then

$$\mu_{v,c_x} = \mu_{v,c_{x-1}} - (\mu^P_{c_{x-1}} + \mu^{\overline{P}}_{c_x}) + (\mu^{\overline{P}}_{c_{x-1}} + \mu^P_{c_x}).$$

Thus, by (4), $\mu_{v,c_x}$ can be computed in $O(1)$ time by using $\mu_{v,c_{x-1}}$. If $c_{x-1} \prec_v c_x$, let $y$ be the integer satisfying

$$[c_{x-1}]_v = \{c_y, c_{y+1}, \ldots, c_{x-1}\}.$$

Since $c_{x-1} \not\equiv_v c_x$, $c_{x-1}$ and $c_x$ are contained in different equivalnce classes for $\equiv_v$. Thus from (4), we have

$$\mu_{v,c_x} = \mu_{v,c_{x-1}} - \left(\sum_{i=y}^{x-2} \mu^{\overline{P}}_{c_i} + \mu^P_{c_{x-1}} + \mu^{\overline{P}}_{c_x}\right) + \left(\sum_{i=y}^{x-1} \mu^H_{c_i} + \mu^P_{c_x}\right). \quad (10)$$

Therefore, $\mu_{v,c_x}$ can be computed in $O(|[c_{x-1}]_v|)$ time by using $\mu_{v,c_{x-1}}$.

Thus, $\{\mu_{v,c} \mid c \in \mathcal{C}(v)\}$ can be computed in $O(|\mathcal{C}(v)|)$ time, since the computation shown in (10) is executed once for each $x$ with $c_x \succ_v c_{x-1}$.

Once $\mu_{v,c}$ are obtained for every $c \in \mathcal{C}(v)$, $\{\mu_v^S \mid S \in \{H, L, P, F, \overline{P}\}\}$ can be computed in $O(|\mathcal{C}(v)|) = O(\deg T v)$ time, by Eqs. (5)–(9). From Lemma 10, except for the recursive calls, COMP_$\mu(v)$ for each vertex $v$ can be done in $O(\deg T v)$ time. Moreover, in the execution of MAX-SIZE$(T, \preceq_T, r)$ shown in Fig. 2, COMP_$\mu(v)$ is called once for every $v \in V(T)$, and thus, MAX-SIZE$(T, \preceq_T, r)$ can be executed in $\sum_{v \in V(T)} \deg T v = O(|V(T)|)$ time. Since MAX-SIZE$(T, \preceq_T, r)$ computes $\max\{\mu_{r'}^H, \mu_{r'}^P\}$, we have the theorem, by Lemma 2. $\square$

```
Input tree preference system (T, ≼_T), a root r ∈ V(T) of T.
Output the size of a maximum stable matching.
begin
    M = ∅.
    let r' be the child of r.
    COMP_μ(r').
    return max{μ^H_{r'}, μ^P_{r'}}.
end
```

**Fig. 2.** Algorithm MAX-SIZE$(T, \preceq_T, r)$.

### 3.2 Computing Maximum Stable Matchings

Before describing an algorithm for computing a maximum stable matching, we modify COMP_$\mu(v)$ to store an edge $(u, v)$ in a matching with $\mu_v^S$ edges for any $S \in \{H, \overline{P}\}$ when $\mu_v^S$ is computed. We use two variables $\gamma(H, v)$ and $\gamma(\overline{P}, v)$ to store a child $c$ of $v$. $\gamma(H, v)$ stores a child $c$ with $(c, v) \in M$ for any $M \in \mathcal{M}_v^H$ corresponding to $\mu_v^H$. $\gamma(\overline{P}, v)$ stores a child $c$ with $(c, v) \in M$ for any $M \in \mathcal{M}_v^H$ corresponding to $\mu_v^{\overline{P}}$.

Figure 3 shows a recursive procedure COMP_$\gamma(v)$, which is obtained from COMP\_$\mu(v)$ shown in Fig. 1 by adding some instructions for $\gamma(S, v)$.

**Lemma 12.** *For any $S \in \{H, \overline{P}\}$ and $v \in V(T) - \{r\}$, $\gamma(S, v)$ stores the edge of $M$ incident with $v$ such that $M \in \mathcal{M}_v^S$ with $|M| = \mu_v^S$, if any.* $\square$

We show an algorithm for computing a maximum stable matching of a tree preference system $(T, \preceq_T)$ in Fig. 4, where Procedure ADD_EDGES$(v, S, M)$ is shown in Fig. 5.

Procedure ADD_EDGES$(v, S, M)$ traverses vertices of $T$ in DFS order, and we have the following.

**Lemma 13.** *For any $S \in \{P, H, \overline{P}\}$ and $M \subseteq E(T)$ with $M \cap E(T(v)) = \emptyset$, ADD_EDGES$(v, S, M)$ adds edges in $M'$ to $M$ for some $M' \in \mathcal{M}_v^S$ satisfying $|M'| = \mu_v^S$.* $\square$

**Input** $v \in V(T) - \{r\}$.
**begin**
  **for** all $c \in \mathcal{C}(v)$
    Comp_$\gamma(c)$.
  **endfor**
  **for** all $c \in \mathcal{C}(v)$
    $\mu_{v,c} = \sum_{c' \in \mathcal{C}(v),\ c' \prec_v c} \mu_{c'}^H + \mu_c^P + \sum_{c' \in \mathcal{C}(v),\ c' \succeq_v c, c' \neq c} \mu_{c'}^{\overline{P}}$.
  **endfor**
  $\mu_v^P = \sum_{c \in \mathcal{C}(v),\ c \prec_v p(v)} \mu_c^H + \sum_{c \in \mathcal{C}(v),\ c \succeq_v p(v)} \mu_c^{\overline{P}} + 1$, where $\mu_v^P = 1$ if $v$ is a leaf.
  (See Lemma 8.)
  $\mu_v^F = \sum_{c \in \mathcal{C}(v)} \mu_c^H$, where $\mu_v^F = 0$ if $v$ is a leaf.
  $\mu_v^H = \max_{c \in \mathcal{C}(v), c \preceq_v p(v)} \mu_{v,c}$, where $\mu_v^H = -\infty$ if $v$ is a leaf.
  $\mu_v^L = \max_{c \in \mathcal{C}(v), c \prec_v p(v)} \mu_{v,c}$, where $\mu_v^L = -\infty$ if $v$ is a leaf.
  **let** $\gamma(H, v)$ be a child $c$ of $v$
    such that $\mu_v^H = \mu_{v,c}$ and $c \preceq_v p(v)$.
  **if** $\mu_v^H \geq \max\{\mu_v^L, \mu_v^F\}$ **then**
    $\mu_v^{\overline{P}} = \mu_v^H$.
    $\gamma(\overline{P}, v) = \gamma(H, v)$.
  **elseif** $\mu_v^L \geq \mu_v^F$ **then**
    $\mu_v^{\overline{P}} = \mu_v^L$.
    **let** $\gamma(\overline{P}, v)$ be a child $c$ of $v$
      such that $\mu_v^L = \mu_{v,c}$ and $c \succ_v p(v)$.
  **else** /* $\mu_v^F > \max\{\mu_v^H, \mu_v^L\}$ */
    $\mu_v^{\overline{P}} = \mu_v^F$.
    $\gamma(\overline{P}, v) = \text{NULL}$.
  **endif**
**end**

**Fig. 3.** Procedure Comp_$\gamma(S, v)$.

From Lemmas 12 and 13, we have the following.

**Theorem 2.** *Algorithm* Max-SMP$(T, \preceq_T, r)$ *solves Max-SMP in linear time for a tree preference system* $(T, \preceq_T)$. □

### 3.3 Computing Maximum-Weight Stable Matchings

A *weighted preference system* $(G, \preceq_G, w)$ is a preference system $(G, \preceq_G)$ with a weight function $w : E(G) \to \mathbb{Z}^+$. For a matching $M$ of $G$, $w(M) = \sum_{e \in M} w(e)$ is a weight of $M$. The maximum-weight stable matching problem (Max-WSMP) is to find a stable matching with maximum weight for a weighted preference system. It is easy to see that we can solve Max-WSMP for weighted tree preference systems by the algorithm in Sect. 3.2 with a slight modification of $\mu_v^S$. For any

**Input** tree preference system $(T, \preceq_T)$, a root $r \in V(T)$ of $T$.
**Output** a maximum stable matching $M$ and $\mu = |M|$.
**begin**
  $M = \emptyset$.
  **let** $r'$ be the child of $r$.
  Comp_$\gamma(r')$.
  **if** $\mu^H_{r'} \geq \mu^P_{r'}$
    $\mu = \mu^H_{r'}$.
    Add_Edges$(r', H, M)$
  **else**
    $\mu = \mu^P_{r'}$.
    Add_Edges$(r', P, M)$
  **endif**
  **return** $M$ and $\mu$.
**end**

**Fig. 4.** Algorithm Max-SMP$(T, \preceq_T, r)$.

$S \in \{H, L, P, F, \overline{P}\}$, we define that

$$\mu^S_v = \max_{M \in \mathcal{M}^S_v} w(M) \tag{11}$$

instead of (1). We also replace (3) and (5) with

$$\mu^P_v = w(v, p(v)) \text{ and} \tag{12}$$

$$\mu^P_v = \sum_{c \in \mathcal{C}(v),\ c \prec_v p(v)} \mu^H_c + \sum_{c \in \mathcal{C}(v),\ c \succeq_v p(v)} \mu^{\overline{P}}_c + w(v, p(v)), \tag{13}$$

respectively, and let Max-WSMP$(T, \preceq_T, r, w)$ be the algorithm obtained by the modifications above. Thus, we have the following.

**Theorem 3.** *Algorithm* Max-WSMP$(T, \preceq_T, r, w)$ *solves Max-WSMP in linear time for a weighted tree preference system* $(T, \preceq_T, w)$. □

# 4 Concluding Remarks

## 4.1 Min-SMP and Min-WSMP

The minimum-weight stable matching problem (Min-WSMP) is to find a stable matching with minimum weight for a weighted preference system. We can compute minimum stable matchings in a similar way. We define $\mu^S_v = +\infty$ if $\mathcal{M}^S_v = \emptyset$. Let Min-SMP$(T, \preceq_T, r)$ be the algorithm obtained from Max-SMP$(T, \preceq_T, r)$ by replacing (2), (6), (7), and (9) with

$$\mu^H_v = \mu^L_v = +\infty$$
$$\mu^H_v = \min\{\mu_{v,c} \mid c \in \mathcal{C}(v), c \preceq_v p(v), c' \neq p(v)\},$$
$$\mu^L_v = \min\{c \in \mathcal{C}(v),\ \mu_{v,c} \mid c \succ_v p(v)\}, \text{ and}$$
$$\mu^{\overline{P}}_v = \min\{\mu^H_v, \mu^L_v, \mu^F_v\},$$

```
Input v ∈ V(T) − {r}, S ∈ {P, H, P̄}, M ⊆ E(T).
begin
  if S = P then
    c_0 = p(v).
  elseif S = H then
    c_0 = γ(v, H).
  else
    c_0 = γ(v, P̄).
  endif
  for all c ∈ C(v) with c ≺_v c_0,  /* c ≺_v c_0 if c_0 = NULL */
    Add_Edges(c, H, M).
  endfor
  for all c ∈ C(v) with c ⪰_v c_0 and c ≠ c_0
    Add_Edges(c, P̄, M).
  endfor
  if S = P then  /* c_0 = p(v) */
    add (v, p(v)) to M.
  elseif c_0 ≠ NULL then  /* S = H or P̄ */
    Add_Edges(c_0, P, M).
  endif
end
```

**Fig. 5.** Procedure ADD_EDGES$(v, S, M)$.

respectively. Then, we have the following.

**Theorem 4.** *Algorithm* MIN-SMP$(T, \preceq_T, r)$ *solves Min-WSMP in linear time for a tree preference system* $(T, \preceq_T)$. □

Moreover, we can also compute the size of a minimum weighted stable matching for a weighted preference system by replacing (3) and (5) with (12) and (13), respectively. Let MIN-WSMP$(T, \preceq_T, r, w)$ be the algorithm obtained from MIN-SMP$(T, \preceq_T, r)$ by the modifications. Then, we have the following.

**Theorem 5.** *Algorithm* MIN-WSMP$(T, \preceq_T, r, w)$ *solves Min-WSMP in linear time for a weighted tree preference system* $(T, \preceq_T, w)$. □

### 4.2 Extensions

The (weighted) preference system can be defined for general graphs without any modification. The general stable matching problem (GSMP) is to find a stable matching of a general preference system $(G, \preceq_G)$, where $G$ is a general graph. It has been known that there exists a general preference system which has no stable matching, and GSMP is NP-hard [6,10]. If $G$ is a bipartite graph, $(G, \preceq_G)$ is referred to as a bipartite preference system in this section.

The maximum-weight general stable matching problem (Max-WGSMP) is to find a stable matching with maximum weight for a weighted general preference system. The minimum-weight general stable matching problem (Min-WGSMP) is to find a stable matching with minimum weight for a weighted general preference system.

It has been known that both Max-WGSMP and Min-WGSMP are NP-hard even for weighted strict general preference systems [2]. It is also known that both problems are sovable in $O(m^2 \log m)$ time for a weighted strict bipartite preference system $(G, \preceq_G, w)$, where $m = |E(G)|$ [1,4,8]. An extension can be found in [1].

It is interesting to note that both Max-WGSMP and Min-WGSMP can be solved in plynomial time if the treewidth of $G$ is bounded by a constant. We can prove the following by extending our resutlts on trees, although the proof is complicated.

**Theorem 6.** *Both Max-WGSMP and Min-WGSMP can be solved in* $O(n\Delta^{k+1})$ *time if the treewidth of* $G$ *is bounded by* $k$*, where* $n = |V(G)|$*, and* $\Delta$ *is the maximum degree of a vertex of* $G$*.* □

It should be noted that if $k = 1$, both problems can be solved in linear time as shown in Theorems 3 and 5.

**Acknowledgements.** The research was partially supported by JSPS KAKENHI Grant Number 26330007.

## References

1. Chen, X., Ding, G., Hu, X., Zang, W.: The maximum-weight stable matching problem: duality and efficiency. SIAM J. Discret. Math. **26**, 1346–1360 (2012)
2. Feder, T.: A new fixed point approach for stable networks and stable marriages. J. Comput. Syst. Sci. **45**, 233–284 (1992)
3. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am. Math. Mon. **69**, 9–15 (1962)
4. Gusfield, D., Irving, R.W.: The Stable Marriage Problem - Structure and Algorithms. Foundations of computing series. MIT Press, Cambridge (1989)
5. Halldórsson, M., Irving, R., Iwama, K., Manlove, D., Miyazaki, S., Morita, Y., Scott, S.: Approximability results for stable marriage problems with ties. Theor. Comput. Sci. **306**, 431–447 (2003)
6. Irving, R.W., Manlove, D.: The stable roommates problem with ties. J. Algorithms **43**, 85–105 (2002)
7. Irving, R.W., Manlove, D., O'Malley, G.: Stable marriage with ties and bounded length preference lists. J. Discret. Algorithms **1**, 213–219 (2009)
8. Király, T., Pap, J.: Total dual integrality of rothblum's description of the stable-marriage polyhedron. Math. Oper. Res. **33**, 283–290 (2008)
9. Manlove, D., Irving, R., Iwama, K., Miyazaki, S., Morita, Y.: Hard variants of stable marriage. Theor. Comput. Sci. **276**, 261–279 (2002)
10. Ronn, E.: NP-complete stable matching problems. J. Algorithms **11**, 285–304 (1990)