Stable Matchings in Trees

Satoshi TAYU[1,a) Shuichi UENO[1,b)

Abstract: The maximum stable matching problem (Max-SMP) and the minimum stable matching problem (Min-SMP) have been known to be NP-hard for bipartite graphs, while Max-SMP can be solved in polynomial time for a bipartite graph G with $\deg_G(v) \leq 2$ for any $v \in X$, where (X, Y) is a bipartition of G. 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.

1. Introduction

Let G be a simple bipartite graph (bigraph). 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 $f \not\preceq_v e$, and $\preceq_G$ is said to be *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 $O(m)$ time by using the Gale/Shapley algorithm [1], where $m = |E(G)|$. 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, and we have the following two problems [2]. 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. It is known that Max-SMP and Min-SMP are both NP-hard [6].

Let (X, Y) be a bipartition of a bigraph G. A bigraph G is called a (p, q)*-graph* if $\deg_G(x) \leq p$ for every $x \in X$, and $\deg_G(y) \leq q$ for every $y \in Y$. It is shown in [5] that Max-SMP is NP-hard even for $(3, 3)$-graphs, while Max-SMP can be solved in polynomial time for $(2, \infty)$-graphs. Some indepth consideration on the approximation for both problems can be found in [3].

[1] Department of Communication and Computer Engineering, Tokyo Institute of Technology, 152-8550-S3-57, Japan
[a) tayu@eda.ce.titech.ac.jp
[b) ueno@eda.ce.titech.ac.jp

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.

2. Stable Matchings in Trees

Let T be a tree, and $(T, \preceq_T)$ be a preference system. We use the following notations:
- $u \preceq_v w$ and $w \succeq_v u$ if $(u, v) \preceq_v (v, w)$,
- $u \equiv_v w$ if $(u, v) \preceq_v (v, w)$ and $(v, w) \preceq_v (u, v)$, and
- $u \prec_v w$ and $w \succ_v u$ 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 $u = w$.

We consider T as a rooted tree with the root r, which is a leaf 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 as follows.
- $\mathcal{M}_v^P$ is the set of v-stable matchings in which v is matched with the parent, $p(v)$.
- $\mathcal{M}_v^H$ is the set of v-stable matchings in which v is matched with a child c with $c \preceq_v p(v)$.
- $\mathcal{M}_v^L$ is the set of v-stable matchings in which v is matched with a child c with $c \succ_v p(v)$.
- $\mathcal{M}_v^F$ is the set of v-stable matchings in which v is not matched with any other vertices.
- $\mathcal{M}_v^{\overline{P}}$ is the set of v-stable matchings in which v is not matched with the parent.

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

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

Lemma 1 $M \subseteq E(T)$ is a stable matching of T if and only if $M \in \mathcal{M}_{r'}^H \cup \mathcal{M}_{r'}^P$.

Proof. Let M be a stable matching of T. Since M is an r'-stable matching, $M \in \mathcal{M}^S_{r'}$ for some $S \in \{P, H, L, F\}$. If $(r, r') \in M$ then $M \in \mathcal{M}^P_v$, by definition. Otherwise, (r, r') must be dominated by an edge in M. Therefore, there is a child c of r' such that $(r', c) \in M$ and $c \preceq_v r$. Thus, we have $M \in \mathcal{M}^H_{r'}$.

Conversely, let $M \in \mathcal{M}^H_{r'} \cup \mathcal{M}^P_{r'}$. 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. Thus, it suffices to show that (r, r') is dominated by an edge in M. If $M \in \mathcal{M}^P_{r'}$, (r, r') is dominated by itself. Otherwise, $M \in \mathcal{M}^H_v$, and there exists a child c of r' such that $(c, r') \in M$ and $c \preceq_{r'} r$. Therefore, (r, r') is dominated by $(c, r') \in M$, and M is a stable matching of T.

Thus, we have the lemma. ■

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

Lemma 2 If M is a v-stable matching of $T(v)$ then $M(c)$ is a c-stable matching of $T(c)$ for any $c \in C(v)$.

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

Lemma 3 For any $M \subseteq E(T(v))$, M is in $\mathcal{M}^P_v$ if and only if the following conditions are satisfied:
(i) $(v, p(v)) \in M$,
(ii) $M(c) \in \mathcal{M}^{\overline{P}}_c$ for every $c \in C(v)$ with $c \succeq_v p(v)$, and
(iii) $M(c) \in \mathcal{M}^H_c$ for every $c \in C(v)$ with $c \prec_v p(v)$.

Proof. Let $M \in \mathcal{M}^P_v$. Then, we have (i) by definition. From Lemma 2, $M(c)$ is a c-stable matching of $T(c)$ for every $c \in C(v)$. For $c \succeq_v p(v)$, $M(c) \in \mathcal{M}^{\overline{P}}_c$ holds, since (i) holds, and $M(c)$ is a matching. Thus, we have (ii). For any $c \prec_v p(v)$, $(v, p(v)) \in M$ does not dominate (c, v), and v is matched with $p(v)$. Therefore, (c, v) must be dominated by an edge (g, c) for some vertex g with $g \preceq_c v$. Since v is the parent of c, g is a child of c. Therefore, $(c, g) \in M(c)$, and we have (iii).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i)–(iii). Then, M is a matching of $T(v)$, since there is only one common vertex v for subtrees $T(c)$, and edges in $M(c)$ are not incident with v. Moreover, all edges of $M(c) - \{(c, v)\}$ are dominated by an edge in $M(c)$ for any $c \in C(v)$, since $M(c)$ is a c-stable matching. Therefore, the rest of the proof is to show that every (c, v) is dominated by an edge in M, since $(v, p(v)) \in M$. For any $c \prec_v p(v)$, we have $M(c) \in \mathcal{M}^H_c$ by (iii), and thus, (c, v) is dominated by an edge $(c, g) \in M(c)$ for some $g \in C(c)$. For any $c \succeq_v p(v)$, (c, v) is dominated by $(v, p(v)) \in M$ by (ii). Thus, all edges in $M(v) - \{(v, p(v))\}$ are dominated by edges in M, i.e., M is v-stable. Therefore, M is in $\mathcal{M}^P_v$, since $(v, p(v)) \in M$. ■

Fig. 1 shows an example of a subtree $T(v)$ and a matching M in $\mathcal{M}^P_v$, where c_i are the children of v such that $c_1 \succ_v p(v)$, $c_2 \equiv_v c_3 \equiv_v p(v)$, and $c_i \prec_v p(v)$ for $i \geq 4$. Some edges of the matching is shown by bold lines. Since $M \in \mathcal{M}^P_v$ and $c_1 \succ_v p(v)$, some edge (c_1, g) with $g \preceq_{c_1} v$ must be contained in M.

For a tree T and a vertex $v \in V(T)$, let $T - v$ be a tree obtained from T by deleting v and edges incident with v.

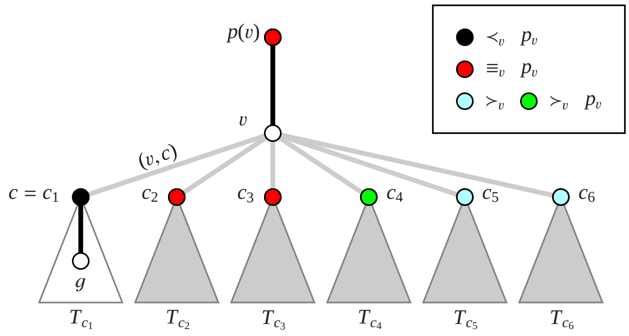

Fig. 1 a matching M of $T(v)$ in $\mathcal{M}^P_v$.

Lemma 4 For any $M \subseteq E(T(v))$, M is in $\mathcal{M}^H_v$ if and only if the following conditions are satisfied:
(i) $(v, p(v)) \notin M$,
(ii) there exists $c' \preceq_v p(v)$ such that $M(c') \in \mathcal{M}^P_{c'}$,
(iii) $M(c) \in \mathcal{M}^H_c$ for every $c \in C(v)$ with $c \prec_v c'$, and
(iv) $M(c) \in \mathcal{M}^{\overline{P}}_c$ for every $c \in C(v)$ with $c \succeq_v c'$ and $c \neq c'$.

Proof. Let $M \in \mathcal{M}^H_v$, and $c' \in C(v)$ be the child of v such that $(c', v) \in M$. Then, we have (i) by definition. From Lemma 2, $M(c)$ is a c-stable matching of $T(c)$ for every $c \in C(v)$. Since $M \in \mathcal{M}^H_v$, there exists $c' \in C(v)$ such that $c' \preceq_v p(v)$ and $(c', v) \in M$. Therefore, $M(c') \in \mathcal{M}^P_{c'}$ and we have (ii). For any $c \prec_v c'$, the edge (c, v) is not dominated by (c', v). Therefore, (c, v) must be dominated by an edge (g, c) incident with c. Since $g \in C(c)$, we have $M(c) \in \mathcal{M}^H_c$. Therefore, (iii) holds. For any $c \succeq_v c'$, (c, v) is dominated by (c', v). Therefore, $M(c) \in \mathcal{M}^{\overline{P}}_c$, and we have (iv).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i)–(iv). Then, M is a matching of $T(v)$, since there is only one common vertex v for subtrees $T(c)$, there is only one edge (c, v) of M incident with v, and $(v, p(v)) \notin M$. Moreover, all edges in $M(c) - \{(c, v)\}$ are dominated by an edge in $M(c)$ for any $c \in C(v)$, since $M(c)$ is a c-stable matching. In addition, there exists $c' \preceq_v p(v)$. Therefore, the rest of the proof is to show that every (c, v) is dominated by an edge in M, since $(v, p(v)) \notin M$ by (i). Since $(c', v) \in M$, (c, v) is dominated by (c', v) for $c \succeq_v c'$. For $c \prec_v c'$, (c, v) is dominated by (g, c) for some $g \in C(c)$, since $M(c) \in \mathcal{M}^H_c$ by (iii). Moreover, (c', v) is dominated by itself. Therefore, all edges (c, v) for $c \in C(v)$ are dominated by an edge in M, and $M(c)$ is v-stable. Thus, M is in $\mathcal{M}^H_v$. ■

Fig. 2 shows an example of a subtree $T(v)$ and a matching M in $\mathcal{M}^H_v$, where c_i are the children of v such that $(c_2, v) \in M$, $c_1 \prec_v p(v)$, $c_2 \equiv_v c_3 \equiv_v p(v)$, and $c_i \succ_v p(v)$ for $i \geq 4$. Some edges of the matching is shown by bold lines. Since $M \in \mathcal{M}^H_v$ and $c_1 \prec_v p(v)$, some edge (c_1, g_1) with $g_1 \preceq_{c_1} v$ must be contained in M.

Lemma 5 For any $M \subseteq E(T(v))$, M is in $\mathcal{M}^L_v$ if and only if the following conditions are satisfied:
(i) $(v, p(v)) \notin M$,
(ii) there exists $c' \succ_v p(v)$ such that $M(c') \in \mathcal{M}^P_{c'}$,
(iii) $M(c) \in \mathcal{M}^H_c$ for every $c \in C(v)$ with $c \prec_v c'$, and
(iv) $M(c) \in \mathcal{M}^{\overline{P}}_c$ for every $c \in C(v)$ with $c \succeq_v c'$ and $c \neq c'$.

Proof. Let $M \in \mathcal{M}^H_v$, and $c' \in C(v)$ be the child of v

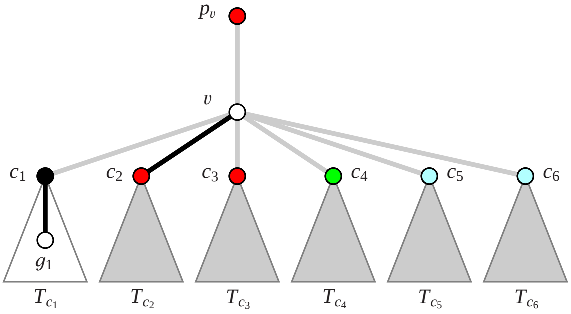

Fig. 2 $M \in \mathcal{M}_v^H$.

such that $(c', v) \in M$. Then, we have (i) by definition. From Lemma 2, $M(c)$ is a c-stable matching of $T(c)$ for every $c \in C(v)$. Since $M \in \mathcal{M}_v^L$, there exists $c' \in C(v)$ such that $c' \succ_v p(v)$ and $(c', v) \in M$. Therefore, $M(c') \in \mathcal{M}_{c'}^P$ and we have (ii). For any $c \prec_v c'$, the edge (c, v) is not dominated by (c', v). Therefore, (c, v) must be dominated by an edge (g, c) incident with c. Since $g \in C(c)$, we have $M(c) \in \mathcal{M}_c^H$. Therefore, (iii) holds. For any $c \succeq_v c'$, (c, v) is dominated by (c', v). Therefore, $M(c) \in \mathcal{M}_c^{\overline{P}}$, and we have (iv).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i)–(iv). Then, M is a matching of $T(v)$, since there is only one common vertex v for subtrees $T(c)$, there is only one edge (c, v) of M incident with v, and $(v, p(v)) \notin M$. Moreover, all edges in $M(c) - \{(c, v)\}$ are dominated by an edge in $M(c)$ for any $c \in C(v)$, since $M(c)$ is a c-stable matching. In addition, there exists $c' \succ_v p(v)$. Therefore, the rest of the proof is to show that every (c, v) is dominated by an edge in M, since $(v, p(v)) \notin M$ by (i). Since $(c', v) \in M$, (c, v) is dominated by (c', v) for $c \succeq_v c'$. For $c \prec_v c'$, (c, v) is dominated by (g, c) for some $g \in C(c)$, since $M(c) \in \mathcal{M}_c^H$ by (iii). Moreover, (c', v) is dominated by itself. Therefore, all edges (c, v) for $c \in C(v)$ are dominated by an edge in M, and $M(c)$ is v-stable. Thus, M is in $\mathcal{M}_v^H$. ■

Fig. 3 shows an example of a subtree $T(v)$ and a matching M in $\mathcal{M}_v^L$, where c_i are the children of v such that $(c_5, v) \in M$, $c_i \prec_v c_5$ for $i \leq 4$, $p(v) \prec_v c_5$, and $c_6 \equiv_v c_5$ Some edges of the matching is shown by bold lines. Since $M \in \mathcal{M}_v^L$ and $c_i \prec_v c_5$ for $i \leq 4$, some edge (c_i, g_i) with $g_i \prec_{c_i} v$ must be contained in M for $i \leq 4$.

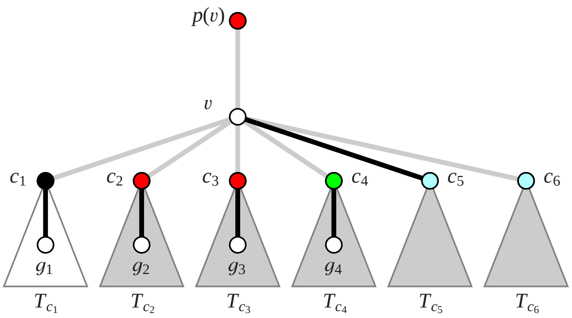

Fig. 3 $M \in \mathcal{M}_v^L$.

Lemma 6 For any $M \subseteq E(T(v))$, M is 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 $c \in C(v)$.

Proof. Let $M \in \mathcal{M}_v^H$. Then, we have (i) by definition. From Lemma 2, $M(c)$ is a c-stable matching of $T(c)$ for every $c \in C(v)$. Since $M \in \mathcal{M}_v^H$, $(c, v) \notin M$ for any $c \in C(v)$. Therefore, $M(c) \in \mathcal{M}_c^P$, and we have (ii).

Conversely, suppose a set $M \subseteq E(T(v))$ satisfies (i) and (ii). Then, M is a matching of $T(v)$, since there is only one common vertex v for subtrees $T(c)$, and there is no edge in $M(c)$ incident with v. In this case, all edges of $M(c) - \{(c, v)\}$ are dominated by an edge in $M(c)$ for any $c \in C(v)$, since $M(c)$ is a c-stable matching. Moreover, for any $c \in C(v)$, every edge (c, v) is dominated by (g, c) for some $g \in C(c)$, since $M(c) \in \mathcal{M}_c^H$. Thus, all edges in $M(v) - \{(v, p(v))\}$ are dominated by edges in M, i.e., M is v-stable. Therefore, M is in $\mathcal{M}_v^F$, since $(v, p(v)) \notin M$. ■

Fig. 4 shows an example of a subtree $T(v)$ and a matching M in $\mathcal{M}_v^F$, where c_i are the children of v. Some edges of the matching is shown by bold lines. Since $M \in \mathcal{M}_v^F$ and $c_i \prec_v c_4$ for $i \leq 3$, some edge (c_i, g_i) with $g_i \prec_{c_i} v$ must be contained in M for any i.

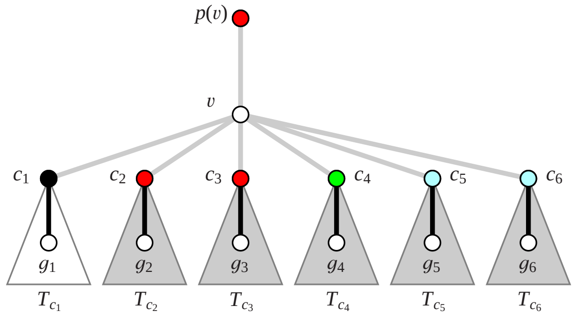

Fig. 4 $M \in \mathcal{M}_v^F$.

3. Linear Time Algorithm for Trees

3.1 Computing the Size of Maximum Stable Matchings

In this section, we show a linear time algorithm to compute the size of a maximum stable matching in a tree. Our algorithm applies a dynamic programing scheme based on the results in the previous section. The algorithm can be modified to find a maximum stable matching in a tree as shown in the next section. Min-SMP can be solved in linear time by a similar method.

For any $S \in \{P, H, L, F, \overline{P}\}$, let μ_v^S be the maximum number of edges of a v-stable matching in $\mathcal{M}_v^S$, where $\mu_v^S = -\infty$ if $\mathcal{M}_v^S = \emptyset$.

If v is a leaf, $T(v)$ is a tree consisting of only one edge $(v, p(v))$. Since v has no child, we have $\mathcal{M}_v^H = \mathcal{M}_v^L = \emptyset$ and thus, $\mu_v^H = \mu_v^L = -\infty$. Since $\{(v, p(v))\}$ is a unique v-stable matching containing $(v, p(v))$, $\mu_v^P = 1$. Moreover, $M = \emptyset$ is also v-stable matching with $M \in \mathcal{M}_v^F$, we have $\mu^F = 0$. Thus, if v is a leaf then

$$\mu_v^H = \mu_v^L = -\infty \tag{1}$$

$\mu_v^P = 1$, and $\mu_v^F = 0$.

Define that for $c \in C(v)$,

$$\mu_{v,c} = \sum_{c' \prec_v c, c' \neq p(v)} \mu_{c'}^H + \mu_c^P + \sum_{c' \succeq_v c, c' \neq c, c' \neq p(v)} \mu_{c'}^{\overline{P}}. \tag{2}$$

From Lemmas 3–6, we have

$$\mu_v^P = \sum_{c \prec_v p(v)} \mu_c^H + \sum_{c \succeq_v p(v), c' \neq p(v)} \mu_c^{\overline{P}} + 1, \tag{3}$$

$$\mu_v^H = \max\{\mu_{v,c} \mid c \preceq_v p(v), c' \neq p(v)\}, \quad (4)$$
$$\mu_v^L = \max\{\mu_{v,c} \mid c \succ_v p(v)\}, \quad (5)$$
$$\mu_v^F = \sum_{c \in C(v)} \mu_c^H, \text{ and} \quad (6)$$
$$\mu_v^{\overline{P}} = \max\{\mu_v^H, \mu_v^L, \mu_v^F\}. \quad (7)$$

Define that, for $c \in C(v)$,

$$[c]_v = \{c' \mid c' \equiv_v c, c' \neq p(v)\}.$$

In order to compute the size of a maximum stable matching, we use a recursive procedure Comp_$\mu(v)$ shown in Fig. 5. From Lemma 1, the maximum size of a stable matching in T is obtained by computing $\max\{\mu_{r'}^H, \mu_{r'}^P\}$, and $\mu_{r'}^H$ and $\mu_{r'}^P$ are obtained by executing Comp_$\mu(r')$.

We use variables $\gamma(H, v)$ and $\gamma(\overline{P}, v)$ to denote children c of v such that $(c, v) \in M$ for $M \in \mathcal{M}_v^H$ and $M \in \mathcal{M}_v^{\overline{P}}$, if any. These variables will be used when a maximum stable matching is computed.

Input $v \in V(T) - \{r\}$.
begin
  for all $c \in C(v)$
    Comp_$\mu(c)$.
  endfor
  for all $c \in C(v)$
    $\mu_{v,c} = \sum_{c' \prec_v c} \mu_{c'}^H + \mu_c^P + \sum_{c' \succeq_v c, c' \neq c} \mu_{c'}^{\overline{P}}$.
  endfor
  $\mu_v^P = \sum_{c \prec_v p(v)} \mu_c^H + \sum_{c \succeq_v p(v)} \mu_c^{\overline{P}} + 1$.
  $\mu_v^F = \sum_{c \in C(v)} \mu_c^H$, where $\mu_v^F = 0$ if v is a leaf.
  $\mu_v^H = \max_{c \preceq_v p(v)} \mu_{v,c}$, where $\mu_v^H = -\infty$ if v is a leaf.
  $\mu_v^L = \max_{c \prec_v p(v)} \mu_{v,c}$, 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 /* $\mu_v^F > \max\{\mu_v^H, \mu_v^L\}$ */
    $\mu_v^{\overline{P}} = \mu_v^F$.
    $\gamma(\overline{P}, v) =$ NULL.
  endif
end

Fig. 5 Procedure Comp_$\mu(v)$.

Lemma 7 Given μ_c^S for all $S \in \{P, H, L, F, \overline{P}\}$ and $c \in C(v)$, $\{\mu_{v,c} \mid c \in C(v)\}$ can be computed in $O(\delta)$ time.

Proof. Let $c_1, c_2, \ldots, c_\delta$ be a sequence of the children of v such that $c_i \preceq_v c_{i+1}$.

From (2), μ_{v,c_1} can be computed in $O(\delta)$ time.

Let $x \geq 2$. If $c_x \equiv_v c_{x-1}$ then

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

Thus, by (2), μ_{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 such that

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

From (2), we have

$$\mu_{v,c_x} = \mu_{v,c_{x-1}} - \left(\sum_{i=y}^{x-2} \mu_{c_i}^{\overline{P}} + \mu_{c_{x-1}}^P + \mu_{c_x}^{\overline{P}}\right) + \left(\sum_{i=y}^{x-1} \mu_{c_i}^H + \mu_{c_x}^P\right). \quad (8)$$

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

Thus, the computation of $\{\mu_{v,c} \mid c \in C(v)\}$ can be done in $O(\delta)$ time, since computation shown in (8) is executed once for each $[c_{x-1}]_v$. ■

Once $\mu_{v,c}$ are obtained for all $c \in C(v)$, the computation of $\{\mu_v^S \mid S \in \{H, L, P, F, \overline{P}\}\}$ can be done in $O(\delta) = O(\deg_T(v))$ time, by (3)–(7). From Lemma 7, except for the recursive calls, Comp_$\mu(v)$ for each vertex v can be done in $O(\deg_T(v))$ time. Since $\sum_{v \in V(T)} \deg_T(v) = O(|V(T)|)$, we have the following.

Theorem 1 The maximum size of a stable matching of trees can be computed in linear time. ■

3.2 Computing Maximum Stable Matchings

In order to compute a maximum stable matching of T, we use the procedure Add_Edges(v, S, M) shown in Fig. 6. In the execution of Add_Edges(v, S, M) except for the recursive calls in it, only edge $(v, p(v))$ can be added to M. In addition, Add_Edges(v, S, M) is executed by a depth first search strategy for v. Therefore, we have the following.

Lemma 8 In the execution of Add_Edges(r', S, M) start with $M = \emptyset$, $M \cap V(T(v)) = \emptyset$ before the execution of Add_Edges(v, S, M) for $v \in V(T) - \{r, r'\}$. ■

Input $v \in V(T) - \{r\}$, $S \in \{P, H, \overline{P}\}$, $M \subseteq E(T)$.
begin
  if $S = P$ **then**
    $c_0 = p(v)$.
  elseif $S = H$ **then**
    $c_0 = \gamma(v, H)$.
  else
    $c_0 = \gamma(v, \overline{P})$.
  endif
  for all $c \in C(v)$ with $c \prec_v c_0$
    /* where $c \prec_v$ NULL for $c \in C(v)$ */
    Add_Edges(c, H, M).
  endfor
  for all $c \in C(v)$ with $c \succeq_v c_0$ and $c \neq c_0$
    Add_Edges$(c, \overline{P}, M)$.
  endfor
  if $S = P$ **then**
    add $(v, p(v))$ **to** M.
  elseif $c_0 \neq$ NULL **then** /* $S = H$ or $\overline{P}$ */
    Add_Edges(c_0, P, M).
  endif
end

Fig. 6 Procedure Add_Edges(v, S, M).

Lemma 9 If $\mu_v^S \geq 0$ for $S \in \{P, H, \overline{P}\}$ and $M \cap E(T(v)) = \emptyset$, then Add_Edges$(v, S, M)$ adds edges in M' to M for some $M' \in \mathcal{M}_v^S$ with $|M'| = \mu_v^S$.

Proof. We prove the lemma by induction.

Assume that v is a leaf. Then, $\mu_v^P = 1$, $\mu_v^{\overline{P}} = 0$, and $\mu_v^H = -\infty$. Therefore, we only need to consider the cases of $S \in \{P, \overline{P}\}$. In this case, Add_Edges(v, S, M) adds $(v, p(v))$ to M if $S = P$, and adds no edge if $S = \overline{P}$. Thus, we have the lemma.

Let v be an internal vertex and assume that the lemma holds for all children $c \in C(v)$ and $S \in \{P, H, \overline{P}\}$. We distinguish four cases.

Case 1 $S = P$.

From Lemma 3, $\mu_c^H \geq 0$ for $c \in C(v)$ with $c \prec_v p(v)$, and $\mu_c^{\overline{P}} \geq 0$ for $c \in C(v)$ with $c \succeq_v p(v)$. In the execution of Add_Edges(v, S, M), Add_Edges(c, H, M) is executed for $c \in C(v)$ with $c \prec_v p(v)$, Add_Edges$(c, \overline{P}, M)$ is executed for $c \in C(v)$ with $c \succeq_v p(v)$, and $(v, p(v))$ is added to M. Thus by induction hypothesis and Lemma 8, edges in $M' = \bigcup_{c \prec_v p(v)} M_c^H \cup \bigcup_{c \succeq_v p(v)} M_c^{\overline{P}} \cup \{(v, p(v))\}$ are added to M such that $M_c^H \in \mathcal{M}_c^H$, $M_c^{\overline{P}} \in \mathcal{M}_c^{\overline{P}}$, $\mu_c^h = |M_c^H|$, and $\mu_c^{\overline{P}} = |M_c^{\overline{P}}|$. Therefore, from Lemma 3 and (3), we obtain $M' \in \mathcal{M}_v^H$ and $\mu_v^H = |M'|$.

Thus, we have the lemma.

Case 2 $S = H$.

In this case, $\gamma(v, H)$ is set to a child c' of v such that $\mu_v^H = \mu_{v,c'}$. From Lemma 4, we have $\mu_c^H \geq 0$ for $c \in C(v)$ with $c \prec_v c'$, $\mu_{c'}^P \geq 0$, and $\mu_c^{\overline{P}} \geq 0$ for $c \in C(v)$ with $c \succeq_v c'$. Then, Add_Edges(c, H, M) for $c \in C(v)$ with $c \prec_v c'$, Add_Edges$(c, \overline{P}, M)$ for $c \in C(v)$ with $c \succeq_v c'$, and Add_Edges(c', P, M) are executed in Add_Edges(v, S, M). Thus by induction hypothesis and Lemma 8, edges in $M' = \bigcup_{c \prec_v c'} M_c^H \cup \bigcup_{c \succeq_v c'} M_c^{\overline{P}} \cup M_c^P$ are added to M such that $M_c^H \in \mathcal{M}_c^H$, $M_c^{\overline{P}} \in \mathcal{M}_c^{\overline{P}}$, $M_{c'}^P \in \mathcal{M}_{c'}^P$, $\mu_c^h = |M_c^H|$, $\mu_c^{\overline{P}} = |M_c^{\overline{P}}|$, and $\mu_{c'}^P = |M_{c'}^P|$. Therefore, from Lemma 4 and (4), we obtain $M' \in \mathcal{M}_v^P$ and $\mu_v^P = |M'|$, respectively.

Thus, we have the lemma.

Case 3 $S = \overline{P}$ and $\gamma(v, \overline{P}) \neq$ NULL.

In this case, $\gamma(v, \overline{P})$ is set to a child c' of v such that $\mu_v^{\overline{P}} = \mu_{v,c'}$. Then by similar arguments to Case 2, we have the lemma.

Case 4 $S = \overline{P}$ and $\gamma(v, \overline{P}) =$ NULL.

Since $\gamma(v, \overline{P}) =$ NULL, $\mu_v^{\overline{P}} = \mu_v^F > \max\{\mu_v^H, \mu_v^L\}$. From Lemma 6, $\mu_c^H \geq 0$ for $c \in C(v)$. Then, Add_Edges(c, H, M) for $c \in C(v)$ are executed in Add_Edges$(v, \overline{P}, M)$. Thus by induction hypothesis and Lemma 8, edges in $M' = \bigcup_{c \in C(v)} M_c$ are added to M, such that $M_c \in \mathcal{M}_c^H$ and $|M_c| = \mu_c^H$. Therefore, from Lemma 6 and (6), we obtain $M' \in \mathcal{M}_v^F \subseteq \mathcal{M}_v^{\overline{P}}$ and $|M'| = \mu_v^F = \mu_v^{\overline{P}}$.

Thus, we have the lemma. ■

From Lemma 9, r'-stable matchings $M^P \in \mathcal{M}_{r'}^P$ with $|M^P| = \mu_{r'}^P$ and $M^H \in \mathcal{M}_{r'}^H$ with $|M^H| = \mu_{r'}^h$ can be computed in linear time. From Lemma 1, such M^P or M^H is a maximum stable matching. Thus, we have the following.

Theorem 2 A maximum stable matching of trees can be computed in linear time. ■

4. Concluding Remarks

4.1 Min-SMP

We can also compute minimum stable matchings by similar arguments as mentioned in Section3.1. In fact, by replacing (1), (4), (5), and (7) with

$$\mu_v^H = \mu_v^L = +\infty$$

$$\mu_v^H = \min\{\mu_{v,c} \mid c \preceq_v p(v), c' \neq p(v)\},$$
$$\mu_v^L = \max\{\mu_{v,c} \mid c \succ_v p(v)\}, \text{ and}$$
$$\mu_v^{\overline{P}} = \max\{\mu_v^H, \mu_v^L, \mu_v^F\},$$

respectively, we can obtain the size of a minimum stable matching by the algorithm shown in Fig. 5. Therefore, we can easily obtain an algorithm for computing a minimum stable matching by modifying algorithms shown in Figs. 5 and 6.

4.2 Extensions

The 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 preference system $(G, \leq_G)$, where G is a general graph. It has been known that there exists a preference system which has no stable matching, and GSMP is NP-hard [4], [7]. We can define associated optimization problems Max-GSMP and Min-GSMP as follows.

Max-GSMP

Instance : A general graph G, a total preorder $\preceq_v$ for every $v \in V(G)$, a weight function $w : E(G) \to \mathbb{Z}^+$.

Question : Find a stable matching M such that

$$\sum_{e \in M} w(e)$$

is maximum. ■

Min-GSMP

Instance : A general graph G, a total preorder $\preceq_v$ for every $v \in V(G)$, a weight function $w : E(G) \to \mathbb{Z}^+$.

Question : Find a stable matching M such that

$$\sum_{e \in M} w(e)$$

is minimum. ■

Since GSMP is a generalization of SMP, we have the following.

Theorem 3 Both Max-GSMP and Min-GSMP are NP-hard. ■

It is interesting to note that both Max-GSMP and Min-GSMP 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 might be complicated.

Theorem 4 Both Max-GSMP and Min-GSMP can be solved in $O(|V(G)|\Delta_G^{k+1})$ time if the treewidth of G is bounded by k, where Δ_G is the maximum degree of a vertex of G. ■

References

[1] D. Gale and L.S. Shapley, "College admissions and the stability of marriage," American Mathematical Monthly, vol.69, no.1, pp.9–15, 1962.
[2] D. Gale and M. Sotomayor, "Some remarks on the stable matching problem," Discrete Applied Mathematics, vol.11, pp.9–15, 1985.
[3] M. Halldórsson, R. Irving, K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, and S. Scott, "Approximability results for stable marriage problems with ties," Theoretical Computer Science, vol.306, no.1-3, pp.431–447, 2003.
[4] R.W. Irving and D. Manlove, "The stable roommates problem with ties," J. Algorithms, vol.43, no.1, pp.85–105, 2002.
[5] R.W. Irving, D. Manlove, and G. O'Malley, "Stable marriage with ties

and bounded length preference lists," J. Discrete Algorithms, vol.1, no.2, pp.213–219, 2009.

[6] D. Manlove, R. Irving, K. Iwama, S. Miyazaki, and Y. Morita, "Hard variants of stable marriage," Theoretical Computer Science, vol.276, no.1-2, pp.261–279, 2002.

[7] E. Ronn, "NP-complete stable matching problems," J. Algorithms, vol.11, no.2, pp.285–304, 1990.