# Characterization of Super-Stable Matchings

Changyong Hu(✉) and Vijay K. Garg

University of Texas at Austin, Austin, TX 78705, USA
colinhu9@utexas.edu, garg@ece.utexas.edu

**Abstract.** An instance of the super-stable matching problem with incomplete lists and ties is an undirected bipartite graph $G = (A \cup B, E)$, with an adjacency list being a linearly ordered list of ties. Ties are subsets of vertices equally good for a given vertex. An edge $(x, y) \in E \backslash M$ is a blocking edge for a matching $M$ if by getting matched to each other neither of the vertices $x$ and $y$ would become worse off. Thus, there is no disadvantage if the two vertices would like to match up. A matching $M$ is super-stable if there is no blocking edge with respect to $M$. It has previously been shown that super-stable matchings form a distributive lattice [14, 23] and the number of super-stable matchings can be exponential in the number of vertices. We give two compact representations of size $O(m)$ that can be used to construct all super-stable matchings, where $m$ denotes the number of edges in the graph. The construction of the second representation takes $O(mn)$ time, where $n$ denotes the number of vertices in the graph, and gives an explicit rotation poset similar to the rotation poset in the classical stable marriage problem. We also give a polyhedral characterization of the set of all super-stable matchings and prove that the super-stable matching polytope is integral, thus solving an open problem stated in the book by Gusfield and Irving [4].

**Keywords:** Super-stable matching · Distributive lattice · Matching polytope

## 1 Introduction

An instance of the super-stable matching problem with incomplete lists and ties is an undirected bipartite graph $G = (A \cup B, E)$, with an adjacency list being a linearly ordered list of ties. Ties are disjoint and may contain just one vertex. If vertices $b_1$ and $b_2$ are neighbors of vertex $a$ in the graph $G$, then either (1) $a$ strictly prefers $b_1$ to $b_2$, which we denote as $b_1 \succ_a b_2$; or (2) $a$ is indifferent between $b_1$ and $b_2$, which means $b_1$ and $b_2$ are in a tie in $a$'s adjacency list, and denote as $b_1 =_a b_2$; or (3) $a$ strictly prefers $b_2$ to $b_1$. We say $a$ weakly prefers $b_1$ to $b_2$ if either $a$ strictly prefers $b_1$ to $b_2$ or $a$ is indifferent between $b_1$ and $b_2$, which we denote as $b_1 \succeq_a b_2$. A matching $M$ is a set of disjoint edges in the graph $G$. Let $e = (u, v)$ be an edge contained in the matching $M$. Then, we say

© Springer Nature Switzerland AG 2021
A. Lubiw and M. Salavatipour (Eds.): WADS 2021, LNCS 12808, pp. 485–498, 2021.
https://doi.org/10.1007/978-3-030-83508-8_35

that vertices $u$ and $v$ are matched in $M$ and write $u = M(v)$ to denote that $u$ is matched to $v$ in $M$. An edge $(x, y) \in E \backslash M$ is a *blocking edge* for a matching $M$ if by getting matched to each other neither of the vertices $x$ and $y$ would become worse off, i.e. $x$ is either unmatched or $x$ weakly prefers $y$ to $M(x)$, and $y$ is either unmatched or $y$ weakly prefers $x$ to $M(y)$. We abuse the notation $y \succeq_x M(x)$ for the case that $x$ is unmatched in $M$. A matching is *super-stable* if there is no blocking edge with respect to it.

Super-stable matchings were first investigated by Irving [6], who gave three classes of stable matchings in the case of preference lists with ties, depending on the way of defining a *blocking edge* for a matching $M$. In the weakly stable matching problem an edge $e = (x, y)$ is blocking if by getting matched to each other, both $x$ and $y$ would become better off. In the strongly stable matching problem, an edge $e = (x, y)$ is blocking if one of $x$ and $y$ becomes better off and the other would not be worse off.

In this paper we study the problem of characterizing the set of all super-stable matchings. The problem was stated in the book by Gusfield and Irving [4] as one of the 12 open problems. The structure of the set of all stable matchings in the stable marriage problem without ties is well understood in Gusfield and Irving's book [4]. Recently, Kunysz et al. [11] gave compact representations for the set of all strongly stable matchings and showed that the construction can be done in $O(mn)$ time, where $n$ and $m$ denote the number of vertices and edges in the graph. Scott [22] investigated the structure of all super-stable matchings by defining an object that he called meta-rotation, which corresponds to one collection of rotations in some arbitrary tie-breaking instance of the original instance and the time complexity of the construction is $O(m^2)$.

We give two compact representations of the set of all super-stable matchings that can be constructed in, respectively, $O(nm^2)$ and $O(mn)$ time.

The first representation of the set of all super-stable matchings consists of $O(m)$ matchings, each of which is a man-optimal stable matching among all super-stable matchings that contains a given edge. We show that computing such matching for each edge can be reduced to computing a man-optimal super-stable matching in a reduced graph by deleting an appropriate subset of edges in graph $G$. The algorithm is described in Sect. 3.

Our second representation explicitly constructs rotations, which are differences between consecutive super-stable matchings in a maximal sequence of super-stable matchings starting with a man-optimal super-stable matching and ending with a woman-optimal super-stable matching. Unlike Scott's [22] meta-rotation, our rotation is the symmetric difference of two super-stable matchings, which could be a cycle or multiple cycles.

Our construction takes $O(mn)$ time, while Scott's [22] algorithm takes $O(m^2)$ time. We also show how to efficiently construct a partial order among rotations. This poset can be used to solve other problems connected to super-stable matchings such as the enumeration of all super-stable matchings and the maximum weight super-stable matching problem. Fleiner et al. [3] solve the weight super-stable matching by reducing it to the 2-SAT problem and the time complexity is $O(mn\log(W))$, where $W$ is the maximum weight among all edges in $G$. By using the rotation poset constructed in this paper, the weighted problem can also be solved in $O(mn\log(W))$ time.

In this paper we also give a polyhedral characterization for the set of all super-stable matchings and prove that the super-stable matching polytope is integral. This result implies that the maximum weight super-stable matching problem can be solved in polynomial time. Though the complexity of solving LP is usually higher than combinatorial methods, like in [3], this gives an alternative direction to solve the weighted super-stable matching problem. Previously, it has been shown that the stable matching polytope and the strongly stable matching polytope are integral [11,21,25], we complete all three cases by proving that the super-stable matching polytope is integral as well.

We also proved a property called self-duality for the super-stable matching polytope, which also holds for the classical stable matching polytope [24] and the strongly stable matching polytope [11]. See details in our full version.

### 1.1 Related Works

Irving [6] gave an $O(m)$ algorithm to find a super-stable matching if it exists. Spieker [23] showed that super-stable matchings form a distributive lattice. Further properties of super-stable matchings were proved by Manlove in [14]. Scott [22] introduced the concept called *meta-rotation poset* for super-stable matchings and showed the one-to-one correspondence between super-stable matchings and closed subsets of the poset.

Irving [6] and Manlove [14] gave an $O(m^2)$ algorithm to find a strongly stable matching if it exists. Kavitha et al. [9] gave an $O(nm)$ algorithm for the strongly stable matching problem. Manlove [14] showed that strongly stable matchings form a distributive lattice. Kunysz et al. [12] gave a characterization of all strongly stable matchings and later Kunysz [11] gave a polyhedral description for the set of all strongly stable matchings and proved that the strongly stable matching polytope is integral.

For weakly stable matchings, it is not true that all weakly stable matchings of a given instance always have the same size. Weakly stable matching can be easily found by running the deferred-acceptance algorithm while breaking ties in an arbitrary manner. The problem of computing a maximum-size weakly stable matching is NP-hard, which has been proved by Iwama et al. [7]. Thus finding good approximations of the problem becomes very interesting. For the version when ties are allowed on both sides, the currently best approximation factor is 3/2 [10,16,17]. For the case when ties only occur on one side, there are a sequence of works pushing the approximation factor lower. Iwama et al. [8] gave an 25/17 approximation algorithm. Huang and Kavitha [5] improved it to 22/15. Later Radnai [20] improved the approximation factor to 41/28, then Dean et al. [2] pushed the approximation factor to 19/13. Most recent result by Lam and Plaxton [13] gave the currently best approximation factor of $1 + 1/e$.

## 2 Preliminaries

In this section we give some definitions and theorems that are useful in the following sections.

**Theorem 1.** [6,14] *There is an $O(m)$ algorithm to determine a man-optimal super-stable matching of the given instance or report that no such matching exists.*

**Theorem 2.** [14] *In a given instance of the super-stable matching problem, the same set of vertices are matched in all super-stable matchings.*

**Lemma 1.** [14] *Let $M, N$ be two super-stable matchings in a given super-stable matching instance. Suppose that, for any agent $p$, $(p, q) \in M$ and $(p, q') \in N$, where $p$ is indifferent between $q$ and $q'$, then $q = q'$.*

We recall some standard notations and definitions from the theory of matchings under preferences. For a given edge $(m, w)$, any matching containing $(m, w)$ is called an $(m, w)$-matching. Let us denote the set of all super-stable matchings of $G$ by $\mathcal{M}_G$. Let $\mathcal{M}_G(m, w)$ be the set of all super-stable $(m, w)$-matchings in $G$.

For two super-stable matchings $M$ and $N$, we say that $M$ *dominates* $N$ and write $M \succeq N$ if each man $m$ weakly prefers $M(m)$ to $N(m)$. If $M$ dominates $N$ and there exists a man $m$ who prefers $M(m)$ to $N(m)$, then we say $M$ *strictly dominates* $N$, write $M \succ N$ and we call $N$ a *successor* of $M$. Note that by Lemma 1, $M \succeq N$ implies $M \succ N$, assuming $M$ is not equal to $N$.

## 3 Irreducible Super-Stable Matchings

In this section, we give our first representation via irreducible matchings. Birkhoff's representation theorem [1] for distributive lattices states that the elements of any finite distributive lattice can be represented as finite sets in such a way that the lattice operations correspond to unions and intersections of sets. The theorem gives a one-to-one correspondence between distributive lattices and partial orders. Our goal is to find the partial order that represents the set of all super-stable matchings.

Distributive lattice is closely related to rings of sets, which is a family of sets that is closed under set unions and set intersections. If the sets in a ring of sets are ordered by set inclusion, they form a distributive lattice. Theory regarding rings of sets and its application to representations of the set of stable matchings in the classical stable marriage problem is well studied by Irving and Gusfield [4]. Below we give a brief summary of this theory that serves as a preliminary for our algorithm.

Given a finite set $B$, the *base* set, a family $\mathcal{F} = \{F_0, F_1, \cdots, F_k\}$ of subsets of $B$ is called a ring of sets over $B$ if $\mathcal{F}$ is closed under set union and intersection. A ring of sets contains a unique minimal element and a unique maximal element.

For any element $a \in B$, we denote $\mathcal{F}(a)$ the set of all elements of $\mathcal{F}$ that contains $a$. It is obvious that $\mathcal{F}(a)$ is also a ring of sets over $B$. We define $F(a)$ to be the unique minimal element of $\mathcal{F}(a)$. An element $F \in \mathcal{F}$ that is $F(a)$ for some $a \in B$ is called *irreducible*. We denote $I(\mathcal{F})$ the set of all irreducible elements of $\mathcal{F}$. We view $(I(\mathcal{F}), \leq)$ as a partial order under the relation $\leq$ of set containment. We give the Birkhoff's representation theorem in the language of rings of sets below.

**Theorem 3.** [4] *i) There is a one-to-one correspondence between the closed subsets of $I(\mathcal{F})$ and the elements of $\mathcal{F}$.*
*ii) If $S$ and $S'$ are closed subsets of $I(\mathcal{F})$ that generate $F = \bigcup S$ and $F' = \bigcup S'$ respectively, then $F \subseteq F'$ if and only if $S \subseteq S'$.*

In the context of super-stable matchings, the base set $B$ corresponds to the set of all acceptable pairs $(m, w) \in E$. We define the $P$-set of a super-stable matching $M$ to be the set of all pairs $(m, w)$, where $w$ is either $M(m)$ or a woman whom $m$ weakly prefers to $M(m)$, which corresponds to an element in $\mathcal{F}$. It is obvious that the unique minimal (man-optimal) super-stable matching in $\mathcal{M}_G(m, w)$, if nonempty, is *irreducible*.

Here we describe an $O(|E|)$ algorithm for computing a man-optimal super-stable $(m, w)$-matching in $G$. Algorithm 1 essentially constructs a reduced graph $G' \subseteq G$ by removing some edges from $G$ (line 3 to line 13 in Algorithm 1). After that, the algorithm computes a man-optimal super-stable matching $M'$ in the reduced graph $G'$. By adding back the edge $(m, w)$, the new matching $M \cup (m, w)$ is super-stable in $G$.

**Lemma 2.** *Let $M$ be a super-stable $(m, w)$-matching. Then $M' = M \backslash \{(m, w)\}$ is a super-stable matching in the reduced graph $G'$.*

*Proof.* We need to prove $M' \subseteq G'$ or equivalently none of edges removed from $G$ is matched in $M'$. Suppose not, an edge $(m', w')$ was removed from $G$ and is matched in $M'$. Note that $m' \neq m$ and $w' \neq w$. Hence, it follows that there is an edge $(m, w')$ or $(m', w)$ which caused the removal of $(m', w')$. W.l.o.g, let's assume it is $(m, w')$ which caused the removal of $(m', w')$. Then we have $w \preceq_m w'$ and $m \succeq_{w'} m'$. Obviously, $(m, w')$ is a blocking pair, which leads to a contradiction of $M$ being super-stable.

To prove super-stability of $M'$ is easy. If there were an edge $e$ blocking $M'$, it would also block $M$.

**Lemma 3.** *Let $M'$ be some super-stable matching in the reduced graph $G'$ if exists. If $M' \cup (m, w)$ is a super-stable matching in $G$, then for each super-stable matching $N'$ in $G'$, $N' \cup (m, w)$ is a super-stable matching in $G$. If $G'$ does not have any super-stable matching, then there is no super-stable $(m, w)$-matching.*

*Proof.* Let $M = M' \cup (m, w)$. Since $M'$ is super-stable in $G'$. It follows that only the removed edges in $E \backslash E'$ can potentially block $M$. We have two cases. (i): any edge that is incident to $m$ or $w$ cannot block $M$. W.l.o.g, Suppose that for some $w'$ that is incident to $m$, and $(m, w')$ blocks $M$. Then we have $w' \succeq_m w$.

**Algorithm 1:** Computing man-optimal super-stable $(m, w)$-matching

**Input:** the graph $G = (A \cup B, E)$ and preference lists of $G$ and an edge $(m, w) \in E$.<br>
**Output:** man-optimal super-stable $(m, w)$-matching or deciding that no such matching exists.<br>
$G' \leftarrow G \backslash \{m, w\}$ // remove $m$ and $w$ and all edges that are incident to them<br>
**for** $m'$ s.t. $(m', w) \in E$ and $m \preceq_w m'$ **do**<br>
    **for** $w'$ s.t. $(m', w') \in E$ and $w \succeq_{m'} w'$ **do**<br>
        $G' \leftarrow G' \backslash (m', w')$<br>
    **end for**<br>
**end for**<br>
**for** $w'$ s.t. $(m, w') \in E$ and $w \preceq_m w'$ **do**<br>
    **for** $m'$ s.t. $(m', w') \in E$ and $m \succeq_{w'} m'$ **do**<br>
        $G' \leftarrow G' \backslash (m', w')$<br>
    **end for**<br>
**end for**<br>
compute man-optimal super-stable matching in $G'$.<br>
**if** exists man-optimal super-stable matching $M$ in $G'$ and $M \cup (m, w)$ is super-stable in $G$<br>
    **return** $M \cup (m, w)$<br>
**else**<br>
    **return** no super-stable $(m, w)$-matching exists.<br>
**end if**

By the construction of $G'$, any edge $(m', w')$ such that $m \succeq_{w'} m'$ was removed. Hence $w'$ must be unmatched in $M$. From Theorem 2, $w'$ is unmatched in any super-stable matching of $G$. Let us assume there exists some super-stable $(m, w)$-matching $N$. Then $N' = N \backslash (m, w)$ is super-stable in $G'$. Since $w'$ is unmatched in $N$, $(m, w')$ blocks $N$, contradiction. (ii): any edge $(m', w')$ such that $m' \neq m$ and $w' \neq w$ cannot block $M$. By the construction of the reduced graph $G'$, the removal of $(m', w')$ was caused by some edge $(m, w')$ or $(m', w)$. W.l.o.g, some edge $(m, w')$ caused the removal of $(m', w')$. Hence, if $w'$ is matched in $M$, then $M(w') \succeq_{w'} m'$. $(m', w')$ does not block $M$. In the case that $w'$ is unmatched in $M$, $w'$ is unmatched in any super-stable matching in $G$. Similar to Case 1, if there exists some super-stable $(m, w)$-matching $N$, then $(m, w')$ blocks N, contradiction. By the same argument, if $M$ is super-stable in $G$, for any other super-stable matching $N'$ in $G'$, $M'$ and $N'$ match the same set of vertices. No edges in $E \backslash E'$ can block $N' \cup (m, w)$.

**Theorem 4.** *Let $(m, w)$ be an edge in $G$. There is an $O(m)$ algorithm for computing a man-optimal super-stable $(m, w)$-matching or deciding that no super-stable $(m, w)$-matching exists.*

*Proof.* Lemma 3 makes sure if Algorithm 1 outputs a matching $M$, then $M$ is super-stable in $G$. Lemma 2 guarantees that if there exists any super-stable matching in $G$, then Algorithm 1 would never miss it.

**Theorem 5.** *$(I(\mathcal{M}_G), \leq)$ can be constructed in $O(nm^2)$ time.*

*Proof.* $I(\mathcal{M}_G)$ can be computed in $O(m^2)$ time by running Algorithm 1 for each edge $(m, w) \in E$. The set $I(\mathcal{M}_G)$ has at most $m$ elements. By checking each pair of $I(\mathcal{M}_G)$, we can construct the partial order. Each check takes $O(n)$ time. Thus, the total time is $O(nm^2)$.

## 4 A Maximal Sequence of Super-Stable Matchings

Representation via irreducible matchings is intuitive, but the time complexity is high. In this section, we give another representation via rotation poset and the time complexity to construct this rotation poset is only $O(mn)$.

Rotation poset derives from the concept of *minimal differences* of a ring of sets. A chain $C = \{C_1, \cdots, C_q\}$ in $\mathcal{F}$ is an ordered set of elements of $\mathcal{F}$ such that $C_i$ is an immediate predecessor of $C_{i+1}$ for each $i \in [q]$. The maximal chain is a chain that begins at the minimal element of $\mathcal{F}$, $F_0$ and ends at the maximal element of $\mathcal{F}$, $F_z$. Let $F_i$ and $F_{i+1}$ be two elements of $\mathcal{F}$ such that $F_i$ is an immediate predecessor of $F_{i+1}$. The difference $D = F_{i+1} \backslash F_i$ is called a *minimal difference* of $\mathcal{F}$. Note that for each two consecutive elements of a chain $C$, there is a minimal difference $D$, we say that $C$ contains $D$. The following two theorems give another version of Birkhoff's representation theorem in the language of minimal differences. The reader can find more details in Irving and Gusfield's book [4].

**Theorem 6.** [4] *If $F$ and $F'$ are two elements in $\mathcal{F}$ such that $F \subset F'$, then every chain from $F$ to $F'$ in $\mathcal{F}$ contains exactly the same set of minimal differences (in a different order).*

**Theorem 7.** [4] *Let $D(\mathcal{F})$ denote the set of all minimal differences in $\mathcal{F}$. For two minimal differences $D$ and $D'$, $D \prec D'$ if and only if $D$ appears before $D'$ on every maximal chain in $\mathcal{F}$. There is a one-to-one correspondence between the elements of $\mathcal{F}$ and the closed subsets of $D(\mathcal{F})$.*

In the context of super-stable matchings, we want to compute a maximal sequence of super-stable matchings in $\mathcal{M}(G)$, i.e. a sequence $M_0 \succ M_1 \succ \cdots \succ M_z$ where $M_0$ is the man-optimal super-stable matching and $M_z$ is the woman-optimal super-stable matching and for each $1 \leq i \leq z$, there is no super-stable matching $M'$ such that $M_{i-1} \succ M' \succ M_i$. We call a matching $M'$ a strict successor of a matching $M$ if $M'$ is a successor of $M$, i.e. $M \succ M'$ and there exists no super-stable matching $M''$ such that $M \succ M'' \succ M'$. We can solve this problem by computing a strict successor of any super-stable matching $M$.

Let $M$ be a super-stable matching in $G$ and $m$ a vertex in $A$. Suppose that there exists a super-stable matching $M'$ such that $m$ gets a worse partner in $M'$ than in $M$, i.e. $M(m) \succ_m M'(m)$. Let $w = M'(m)$, by Lemma 1, $w$ must be matched in $M$ and $m \succ_w M(w)$. Hence we are essentially searching for some vertex $w$ such that $M(m) \succ_m w$ and $m \succ_w M(w)$. In Algorithm 2, the set $E_c$

contains for each man $m$ highest ranked edges incident to him that satisfies the condition above. For each man $m$, the candidate edge $(m, w)$ is not unique, there might be other edge $(m, w')$ that forms a tie with $(m, w)$. While in the case of strict preference list, the candidate edge is unique.

A strongly connected component $S$ of a directed graph $G$ is a subgraph $S$ that is strongly connected, i.e. there is a path in $S$ in each direction between each pair of vertices of $S$, and is maximal with this property: no additional edges or vertices from $G$ can be included in the subgraph without breaking its property of being strongly connected. We say that $e = (m, w)$ is an outgoing edge of $S$ if $m \in S$ and $w \notin S$. Let $S(m)$ denote the strongly connected component that contains $m$.

In Algorithm 2 given below we maintain a directed graph $G_d = (V, E_d)$, whose every edge $(m, w) \in E_d \cap M$ is directed from $w$ to $m$ and every other edge $(m, w)$ is directed from $m$ to $w$. $G_d$ is a subgraph of $G$ that contains the edges the algorithm traverses so far. The basic idea of this algorithm is that for each man $m$ such that $M(m) \neq M_z(m)$, we traverse the preference list of $m$ until we find some candidate edges defined above. We add the edges traversed into $G_d$ and the candidate edges into $G_c$. For each strongly connected component $S$ of $G_d$ without outgoing edges, we try to find a perfect matching on $S$ in $G_c = (V, E_c)$. If we are successful, we find a strict successor of $M$. Otherwise, we modify $G_c$ and $G_d$ by allowing edges of lower ranks.

### 4.1 Correctness of Algorithm 2

Due to the space limit, we defer the proof of Lemma 4, Lemma 5 and Lemma 6 in our full version. Lemma 4 proves that any edge removed from $G_d$ (line 9 and line 30) never block any super-stable matching that the algorithm will output.

**Lemma 4.** *Let $M$ be a super-stable matching in $G$. For any successor $N$ of $M$ such that $N$ is also a super-stable matching in $G$ and each $(m, w) \in M$, any edge $(m, w')$ such that $w' \succeq_m w$ or $(m', w)$ such that $m \succ_w m'$ cannot block $N$.*

**Lemma 5.** *No edge deleted in line 17 can belong to any super-stable matching $N$ dominated by $M$.*

**Lemma 6.** *No edge deleted in line 23 can belong to any super-stable matching $N$ dominated by $M$.*

**Lemma 7.** *The output matching $M_i$ is super-stable and a strict successor of $M_{i-1}$.*

*Proof.* Note that the algorithm outputs $M_i$ when the edge set $E_c$ is a perfect matching in a strongly connected component $S$ with no outgoing edges and $M_i = (M_{i-1} \backslash S) \cup (E_c \cap S)$. Suppose, for a contradiction, that $M_i$ is blocked by some edge $(m, w) \in E_d$. There are four cases. (i): $m \notin S$ and $w \notin S$, it is obvious that $(m, w)$ cannot block $M_i$, since it would block $M_{i-1}$ as well. (ii): $m \in S$ and $w \notin S$, this is not possible, because this will imply $S$ has an outgoing edge

**Algorithm 2:** Computing a maximal sequence of super-stable matchings

let $M_0$ be the (unique) man-optimal super-stable matching of $G$.<br>
let $M_z$ be the (unique) woman-optimal super-stable matching of $G$.<br>
$M \leftarrow M_0$<br>
let $M'$ contain edge $(m, M(m))$ for each man $m$ such that $M(m) =_m M_z(m)$<br>
let $E_d$ contain all edges of $M$<br>
let $G_d$ be the directed graph $(V, E_d)$ such that each edge $(m, w) \in E_d \cap M$ is directed from $w$ to $m$ and every other edge $(m, w)$ is directed from $m$ to $w$<br>
$E' \leftarrow E \backslash E_d$<br>
let $E_c = M'$ and $G_c = (V, E_c)$<br>
for each $(m, w) \in M$ remove from $E'$ each edge $(m', w)$ such that $m' \prec_w m$ and each edge $(m, w')$ such that $w' \succeq_m w$<br>
**repeat**<br>
    **while** $(\exists m \in A)$ $deg_{G_c}(m) = 0$ **do**<br>
        add the set $E_m$ of top choices of $m$ from $E'$ to $E_d$<br>
        **if** $outdeg(S(m)) = 0$ **then**<br>
            add every edge $(m, w) \in E_m$ such that $m \succ_w M(w)$ and<br>
            $M(m) \succ_m w$ to $E_c$<br>
            for each edge $(m, w)$ of $E_c$ that becomes strictly dominated by<br>
            some added edge $(m', w)$ remove it from $G_c$<br>
            remove $E_m$ from $E'$<br>
        **end if**<br>
    **end while**<br>
    **for** each $m \in A$ such that $outdeg(S(m)) = 0$ **do**<br>
        delete all lowest ranked edge in $E_c \cup E'$ incident to any $w \in S$ such<br>
        that $w$ is multiple engaged<br>
    **end for**<br>
    **while** $(\exists S)$ $outdeg(S) = 0$ and $E_c$ is a perfect matching on $S$ **do**<br>
        $M \leftarrow (E_c \cap S) \cup (M \backslash S)$<br>
        $M_i \leftarrow M$<br>
        output $M_i$<br>
        $i \leftarrow i + 1$<br>
        update $G_c$ and $G_d$: $E_c \cap S$ contains only edges $(m, M(m))$ such that<br>
        $M(m) =_m M_z(m)$; an edge $(m, w) \in S$ stays in $G_d$ only if $w = M(m)$<br>
        and $rank_w(m) \leq rank_M(w)$<br>
    **end while**<br>
**until** $(\forall m \in A)$ $rank_M(m) = rank_{M_z}(m)$

in $E_d$. (iii): $m \notin S$ and $w \in S$, then $M_i(m)(= M_{i-1}(m)) \succ_m w$, hence $(m, w)$ would not block $M_i$. (iv): $m \in S$ and $w \in S$, if $(m, w)$ never belong to $E_c$, then $M_i(w) \succ_w M_{i-1}(w) =_w m$, $(m, w)$ can not block $M_i$; if $(m, w)$ once belongs to $E_c$ and got deleted later, then $w$ always get a strictly better partner than $m$. We prove that no edge from $E_d$ can block $M_i$. There might be some other edges $e \notin E_d$ that can potentially block $M_i$. These edges are deleted during the updating of $E_d$. Lemma 4 gives a proof that these set of edges cannot block any matching $N$ that is dominated by $M_{i-1}$. Hence $M_i$ is super-stable.

Next we prove that $M_i$ is a strict successor of $M_{i-1}$. Suppose not and let $m$ be any man in $S$ and $N$ a successor of $M_{i-1}$ such that $M_{i-1}(m) \succ N(m) \succ M_i(m)$. If $(m, N(m)) \in E_c$ and is not deleted during the algorithm, then $(m, M_i(m))$ would not be in $E_c$, which is not true. Since $N$ is a successor of $M$ and is super-stable, by Lemma 5 and Lemma 6, the edge $(m, N(m))$ can never once belong to $E_c$. Let $w = N(m)$, by our updating rule of $E_d$, we have $N(w) \succeq_w M(w)$. While if $N(w) \succ_w M(w)$, then the edge $(m, w)$ must once belong to $E_c$. Thus we have $N(w) =_w M(w)$, which violates Lemma 1.

**Lemma 8.** *If $M_i \neq M_z$, the algorithm always outputs a matching.*

*Proof.* The algorithm will end without outputting any matching if and only if in line 25 the while loop, it cannot find any strongly connected component with no outgoing edges. Note that every directed graph can be expressed as a directed acyclic graph of its strongly connected components. Hence, we can always find a strongly connected component without outgoing edges.

**Theorem 8.** *Algorithm 2 computes a maximal sequence of super-stable matchings.*

*Proof.* By Lemma 7 and Lemma 8, it is obvious that Algorithm 2 outputs a maximal sequence of super-stable matchings.

### 4.2 Running Time of Algorithm 2

**Theorem 9.** *The running time of Algorithm 2 is $O(mn)$.*

*Proof.* Each time we add new edges into $E_d$, we need to compute strongly connected components of $G_d$. Computing strongly connected component of any directed graph $G' = (V', E')$ can be done in $O(E)$ time. Each edge $e$ of $G$ is added to $G_d$ at most once, and $G_d$ is always a subgraph of $G$. Hence, a naive implementation takes $O(m^2)$ on computing strongly connected components of $G_d$. As mentioned in [12], Pearce [19] and Pearce and Kelly [18] sketch how to extend their algorithm and that of Marchetti-Spaccamela et al. [15] to compute strongly connected component dynamically. Their algorithm runs in $O(mn)$ if edges can only be added to the graph and not deleted. The edges in $G_d$ can be deleted during the algorithm, but they are deleted only when $E_c$ is perfect on a strongly connected component without outgoing edges. Thus, other strongly connected components are unchanged. Also as mentioned in [12], the edges remaining in the selected strongly connected component can be treated as they were added anew to the graph. Since the ranks of men increase as we output subsequent super-stable matchings, each edge can be added anew to $G_d$ constant number of times. Thus, the amortized cost of edge insertion remains unchanged. The reader can easily check the other part of the algorithm takes at most $O(m)$ time. Hence, the total time is $O(mn)$.

### 4.3 Rotation Poset

We have shown all rotations $D(\mathcal{M}_G)$ can be found in time $O(mn)$ by Algorithm 2. It remains to show how to efficiently construct the precedence ration $\prec$ on $D(\mathcal{M}_G)$. Our construction is essentially the same as the construction given in [4] for the classical stable marriage problem. The only difference here is that one rotation for super-stable matchings can be one or multiple cycles, while one rotation for stable matchings in the classical stable marriage problem is always a cycle. The reader can find more details in [4]. Due to the space limit, we defer this section in our full version.

We summarize Sect. 4 with the following theorem.

**Theorem 10.** *The partial order $(D(\mathcal{M}_G), \prec)$ can be constructed in $O(mn)$.*

*Proof.* The construction of $D(\mathcal{M}_G)$ takes $O(mn)$ time by running Algorithm 2. The precedence relation can be constructed in $O(m)$ time. Hence, the time complexity is $O(mn)$.

## 5 The Super-Stable Matching Polytope

In this section, we give a polyhedral characterization of the set of all super-stable matchings and prove that the super-stable matching polytope is integral. The main result is the following theorem.

**Theorem 11.** *Let $G = (V, E)$ be a stable matching problem with ties where the graph $G$ is bipartite, then the super-stable matching polytope $SUSM(G)$ is described by the following linear system:*

$$\sum_{u \in N(v)} x_{u,v} \leq 1, \qquad \forall v \in V, \tag{1a}$$

$$\sum_{i >_u v} x_{u,i} + \sum_{j >_v u} x_{j,v} + x_{u,v} \geq 1, \qquad \forall (u,v) \in E, \tag{1b}$$

$$x_{u,v} \geq 0, \qquad \forall (u,v) \in E \tag{1c}$$

*where $N(v)$ denotes the set of neighbors of $v$ in $G$, and $w >_u v$ means $u$ prefers $w$ to $v$.*

*Proof.* Let $x$ be a feasible solution. Define $E^+$ to be the set of edges $(u, v)$ with $x_{u,v} > 0$, and $V^+$ the set of vertices covered by $E^+$. For each $u \in V^+$, let $N^*(u)$ be the maximal elements in $\{i : x_{u,i} > 0\}$. Note that there might be multiple maximal elements that form a tie.

We first show the following lemma.

**Lemma 9.** *For each vertex $u$ and each vertex $v \in N^*(u)$, then $u$ is the unique minimal element in $\{j : x_{j,v} > 0\}$ and that $\sum_{j \in N(v)} x_{j,v} = 1$.*

*Proof.* Indeed, (1b) implies

$$1 \leq \sum_{j >_v u} x_{j,v} + x_{u,v} = \sum_{j \in N(v)} x_{j,v} - \sum_{j <_v u} x_{j,v} - \sum_{\substack{j =_v u; \\ j \neq u}} x_{j,v} \leq 1 - \sum_{j <_v u} x_{j,v} - \sum_{\substack{j =_v u; \\ j \neq u}} x_{j,v} \leq 1 \tag{2}$$

Hence we have equality throughout in (2). This implies that $x_{j,v} = 0$ for each $\{j : j <_v u\}$ and each $\{j : j =_v u; j \neq u\}$ and that $\sum_{j \in N(v)} x_{j,v} = 1$. Since $x_{j,v} = 0$ for each $\{j : j =_v u; j \neq u\}$, $v$ strictly prefers any other vertices in $\{j : x_{j,v} > 0\}$ over $u$, making $u$ the unique minimal element in $\{j : x_{j,v} > 0\}$.

We then prove that for any $v$ such that $v \in N^*(u)$ for some $u$, then $u$ is unique. Suppose not, there is a vertex $u' \neq u$ and $v \in N^*(u')$. By Lemma 9, $u$ is the unique minimal element in $\{j : x_{j,v} > 0\}$, and $u'$ is the unique minimal element in $\{j : x_{j,v} > 0\}$, contradiction.

Now let $U$ and $W$ be the color classes of $G$. For any $u \in U \cap V^+$, there is at least one unique vertex $w \in N^*(u)$, such that $\sum_{j \in N(w)} x_{j,w} = 1$. Let $F_W(x)$ be the set of these vertices. Formally, $F_W(x) = \{w : w \in N^*(u), u \in U \cap V^+\}$. Then we have $|F_W(x)| \geq |U \cap V^+|$. We also have that

$$|F_W(x)| = \sum_{w \in F_W(x)} \sum_{j \in N(w)} x_{j,w} = \sum_{j \in U \cap V^+} \sum_{w \in F_W(x)} x_{j,w} \leq \sum_{j \in U \cap V^+} 1 = |U \cap V^+| \tag{3}$$

implying that $|F_W(x)| = |U \cap V^+|$. Hence, we conclude that for each $u \in U \cap V^+$, $|N^*(u)| = 1$, which implies that $u$ has an unique maximal element in $\{i : x_{u,i} > 0\}$. Since $|N^*(u)| = 1$, we denote this unique vertex as $x^*(u)$. We then have the following corollary.

**Corollary 1.** *There is a bijection between $U \cap V^+$ and $F_W(x)$, and for each $u \in U \cap V^+$, $\sum_{i \in N(u)} x_{u,i} = 1$.*

Similarly, we may define $F_U(x) = \{u : u \in N^*(w), w \in W \cap V^+\}$ and we have

**Corollary 2.** *There is a bijection between $W \cap V^+$ and $F_U(x)$, and for each $w \in W \cap V^+$, $\sum_{j \in N(w)} x_{j,w} = 1$.*

Then we have $|U \cap V^+| = |F_W(x)| \leq |W \cap V^+|$ and $|W \cap V^+| = |F_U(x)| \leq |U \cap V^+|$, implying $|U \cap V^+| = |W \cap V^+| = |F_W(x)| = |F_U(x)|$. Then any $u \in U \cap V^+$ is also in $F_U(x)$, hence, $u$ has an unique minimal element, denoted by $x_*(u)$.

The bijection between $U \cap V^+$ and $F_W(x)$ forms a perfect matching $M$ in $(V^+, E^+)$, i.e. the set of edges $\{(u, x^*(u)) : u \in U \cap V^+\}$. Similarly, the bijection between $W \cap V^+$ and $F_U(x)$ forms another perfect matching $N$, i.e. the set of edges $\{(x^*(w), w) : w \in W \cap V^+\}$.

Consider the vector $x' = x + \varepsilon\chi^M - \varepsilon\chi^N$, with $\varepsilon$ close enough to 0 (positive or negative). we will show that $x'$ is also feasible solution of (1a)–(1c). It is easy to see that $x'$ satisfies (1a) and (1c). For each vertex $u \in U \cap V^+$, there is an unique maximal element $x^*(u)$ and $(u, x^*(u)) \in M$ and an unique minimal element $x_*(u)$ and $(u, x_*(u)) \in N$, implying $\sum_{i \in N(u)} x'_{u,i} = \sum_{i \in N(u)} x_{u,i} \leq 1$. To see that $x'$ satisfies (1b), let $(u, v)$ be an edge in $E^+$ attaining equality in (1b). The case that $(u, v) \in M$ or $(u, v) \in N$ is trivial. So assume that $(u, v) \notin M$ and $(u, v) \notin N$. The edge $(u, x^*(u)) \in M$ and $x^*(u) >_u v$. There is no other edge in $\{(u, i) : i \in N(u)\}$ belongs to $M$. We prove that there is no edge $(j, v)$ in $M$ and $j >_v u$ since if $(j, v) \in M$, $j$ is the minimal element of $v$. Similarly, we can prove that there is exact one edge $(j, v) \in N$ and $j >_v u$. Concluding, $\sum_{i >_u v} x'_{u,i} + \sum_{j >_v u} x'_{j,v} + x'_{u,v} = \sum_{i >_u v} x_{u,i} + \sum_{j >_v u} x_{j,v} + x_{u,v} = 1$. Let $x$ be an extreme point. The feasibility of $x'$ implies that $\chi^M = \chi^N$, that is, $M = N$. So $E^+ = M$ since the maximal element is the same as the minimal element for each vertex, hence, $x = \chi^M$.

### 5.1 Partial Order Preference Lists

Partial order preference lists are generalisation of preference lists with ties in such a way that the preference list of each man or woman is an arbitrary partial order. It turns out that the linear system (1a)–(1c) can also describe the set of all super-stable matchings with partial order preference list. See more details in our full version.

### 5.2 The Strongly Stable Matching Polytope

Kunysz [11] gives a linear system that characterizes the set of all strongly stable matchings and proves this linear system is integral using the duality theory of linear programming. Here, we give an alternate and simpler proof that does not rely on the duality theory and uses only Hall's theorem. See the proof in our full version.

## References

1. Birkhoff, G., et al.: Rings of sets. Duke Math. J. **3**(3), 443–454 (1937)
2. Dean, B., Jalasutram, R.: Factor revealing lps and stable matching with ties and incomplete lists. In: Proceedings of the 3rd International Workshop on Matching Under Preferences, pp. 42–53 (2015)
3. Fleiner, T., Irving, R.W., Manlove, D.F.: Efficient algorithms for generalized stable marriage and roommates problems. Theoret. Comput. Sci. **381**(1–3), 162–176 (2007)
4. Gusfield, D., Irving, R.W.: The Stable marriage problem - structure and algorithms. Foundations of computing series. MIT Press (1989)
5. Huang, C.-C., Kavitha, T.: An improved approximation algorithm for the stable marriage problem with one-sided ties. In: Lee, J., Vygen, J. (eds.) IPCO 2014. LNCS, vol. 8494, pp. 297–308. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07557-0_25

6. Irving, R.W.: Stable marriage and indifference. Discret. Appl. Math. **48**(3), 261–272 (1994)
7. Iwama, K., Miyazaki, S., Morita, Y., Manlove, D.: Stable marriage with incomplete lists and ties. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 443–452. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48523-6_41
8. Iwama, K., Miyazaki, S., Yanagisawa, H.: A 25/17-approximation algorithm for the stable marriage problem with one-sided ties. Algorithmica **68**(3), 758–775 (2014). https://doi.org/10.1007/s00453-012-9699-2
9. Kavitha, T., Mehlhorn, K., Michail, D., Paluch, K.E.: Strongly stable matchings in time o (nm) and extension to the hospitals-residents problem. ACM Trans. Algorithms (TALG) **3**(2), 15-es (2007)
10. Király, Z.: Linear time local approximation algorithm for maximum stable marriage. Algorithms **6**(3), 471–484 (2013)
11. Kunysz, A.: An algorithm for the maximum weight strongly stable matching problem. In: 29th International Symposium on Algorithms and Computation (ISAAC 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018)
12. Kunysz, A., Paluch, K., Ghosal, P.: Characterisation of strongly stable matchings. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 107–119. SIAM (2016)
13. Lam, C.K., Plaxton, C.G.: A (1+ 1/e)-approximation algorithm for maximum stable matching with one-sided ties and incomplete lists. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 2823–2840. SIAM (2019)
14. Manlove, D.F.: The structure of stable marriage with indifference. Discret. Appl. Math. **122**(1–3), 167–181 (2002)
15. Marchetti-Spaccamela, A., Nanni, U., Rohnert, H.: Maintaining a topological order under edge insertions. Inf. Process. Lett. **59**(1), 53–58 (1996)
16. McDermid, E.: A 3/2-approximation algorithm for general stable marriage. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 689–700. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02927-1_57
17. Paluch, K.: Faster and simpler approximation of stable matchings. Algorithms **7**(2), 189–202 (2014)
18. Pearce, D.J., Kelly, P.H.: Online algorithms for topological order and strongly connected components. Technical report, Citeseer (2003)
19. Pearce, D.J.: Some directed graph algorithms and their application to pointer analysis. Ph.D. thesis, University of London (2005)
20. Radnai, A.: Approximation algorithms for the stable matching problem. Eötvös Lorand University (2014)
21. Rothblum, U.G.: Characterization of stable matchings as extreme points of a polytope. Math. Program. **54**(1–3), 57–67 (1992)
22. Scott, S.: A study of stable marriage problems with ties. Ph.D. thesis. University of Glasgow (2005)
23. Spieker, B.: The set of super-stable marriages forms a distributive lattice. Discret. Appl. Math. **58**(1), 79–84 (1995)
24. Teo, C.P., Sethuraman, J.: The geometry of fractional stable matchings and its applications. Math. Oper. Res. **23**(4), 874–891 (1998)
25. Vate, J.H.V.: Linear programming brings marital bliss. Oper. Res. Lett. **8**(3), 147–153 (1989)