Fleiner, T. and Irving, R.W. and Manlove, D.F. (2007) Efficient algorithms for generalized Stable Marriage and Roommates problems. *Theoretical Computer Science* 381(1-3):pp. 162-176.

Efficient algorithms for generalised stable marriage and roommates problems

Tamás Fleiner[1,*], Robert W. Irving[2,†] and David F. Manlove[2,†,‡]

[1] *Department of Computer Science and Information Theory, Budapest University of Technology and Economics, H-1111 Budapest, Hungary.*
Email: fleiner@cs.bme.hu.

[2] *Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK.*
Email: {rwi,davidm}@dcs.gla.ac.uk.

Abstract

We consider a generalisation of the Stable Roommates problem (SR), in which preference lists may be partially ordered and forbidden pairs may be present, denoted by SRPF. This includes, as a special case, a corresponding generalisation of the classical Stable Marriage problem (SM), denoted by SMPF. By extending previous work of Feder, we give a two-step reduction from SRPF to 2-SAT. This has many consequences, including fast algorithms for a range of problems associated with finding "optimal" stable matchings and listing all solutions, given variants of SR and SM. For example, given an SMPF instance I, we show that there exists an $O(m)$ "succinct" certificate for the unsolvability of I, an $O(m)$ algorithm for finding all the super-stable pairs in I, an $O(m+kn)$ algorithm for listing all the super-stable matchings in I, an $O(m^{1.5})$ algorithm for finding an egalitarian super-stable matching in I, and an $O(m)$ algorithm for finding a minimum regret super-stable matching in I, where n is the number of men, m is the total length of the preference lists, and k is the number of super-stable matchings in I. Analogous results apply in the case of SRPF.

Keywords: Stable Roommates problem; Stable marriage problem; Partial order; Forbidden pair; Super-stable matching

1 Introduction

The Stable Roommates problem (SR) is a classical combinatorial problem that has received much attention in the literature [8, 19, 13, 11, 28, 17]. Gale and Shapley [8] were the first to study SR and defined an instance I to comprise n agents, where n is even, each of whom ranks the others in strict order of preference. A *matching* in I is a set of $n/2$ disjoint (unordered) pairs of agents. A matching M is *stable* if there is no pair of agents $\{a_i, a_j\}$, each of whom prefers the other to his partner in M. Such a pair is said to *block* M, or to be a *blocking pair* with respect to M. A blocking pair represents a situation in which

*Supported by the János Bolyai Research Fellowship of the Hungarian Academy of Sciences, and OTKA grant F 037301 of the Hungarian Scientific Research Fund. Part of this work was carried out during a visit to the Department of Computing Science, University of Glasgow, supported by the Royal Society of Edinburgh International Exchange Programme.

†Supported by Engineering and Physical Sciences Research Council grant EP/E011993/1.

‡Supported by Royal Society of Edinburgh / Scottish Executive Personal Research Fellowship, and Engineering and Physical Sciences Research Council grant GR/R84597/01. Part of this work was carried out during a visit to the Egerváry Research Group on Combinatorial Optimization, Department of Operations Research, Eötvös Loránd University.

the two agents involved would rather disregard their partners in M and become matched to each other, undermining the integrity of M. Gale and Shapley [8] showed that an SR instance need not admit a stable matching. Irving [13] solved a problem posed by Knuth [19] when he described an $O(n^2)$ algorithm – linear in the input size – that finds a stable matching or reports that none exists, for a given instance of SR.

As the problem name suggests, an application of SR arises in the context of campus accommodation allocation, where we seek to assign students to share two-person rooms, based on their preferences over one another. Another application occurs in the context of forming pairings of players for chess tournaments [20]. Very recently, a more serious application of SR has been studied, involving pairwise kidney exchange between incompatible patient-donor pairs [27]. Here, preference lists can be constructed on the basis of compatibility profiles between potential donors and existing patients.

SR is a non-bipartite extension of the classical Stable Marriage problem (SM) [8]. In an SM instance, the set of agents is partitioned into two disjoint sets, the *men* and *women*, each of size $n/2$, and each person ranks all members of the opposite sex in strict order of preference. A matching is a set of $n/2$ disjoint (man,woman) pairs, whilst a blocking pair is a (man,woman) pair, each of whom prefers the other to his/her partner in M. Every instance of SM admits at least one stable matching, and such a matching may be found in $O(n^2)$ time using the Gale/Shapley algorithm [8]. SM and its many-one variant (the Hospitals / Residents problem or College Admissions problem [8, 11]) arise in many practical applications, such as the annual match of graduating medical students to hospital posts in a number of countries [26]. Furthermore SM is a special case of SR: given an SM instance I, we may construct in $O(n^2)$ time an SR instance J such that the stable matchings in I are in 1-1 correspondence with the stable matchings in J [11, Lemma 4.1.1].

Incomplete preference lists

SR may be generalised by allowing the preference lists of those involved to be incomplete. In this case, we say that agent p is *acceptable* to agent q if p appears on the preference list of q, and *unacceptable* otherwise. Also we need not insist that n is even. Such a generalisation of the original SR definition has been referred to in the literature as SRI (Stable Roommates with Incomplete lists) [17], though for convenience, in this paper from this point onwards, each use of the term SR or SM refers to the more general problem model that includes the possibility of incomplete lists.

In this context, a matching M must satisfy the property that $\{a_i, a_j\} \in M$ implies that a_i, a_j find each other acceptable. We also revise the definition of stability as follows: a pair of mutually acceptable agents $\{a_i, a_j\}$ is a blocking pair of M if each is either unmatched in M or prefers the other to his partner in M. (It follows that, from the point of view of finding stable matchings, we may assume without loss of generality that p is acceptable to q if and only if q is acceptable to p.) Note that this definition assumes that an agent would prefer to be matched to an acceptable partner rather than to remain unmatched.

A stable matching for an instance of SR need not be a complete matching. However, all stable matchings for a given instance have the same size and match exactly the same set of agents [11, Theorem 4.5.2]. It is straightforward (see [11, Section 4.5.2]) to extend Irving's algorithm [13] to give an $O(m)$ algorithm that finds a stable matching or reports that none exists, given an SR instance I (with possibly incomplete lists), where m is the total length of the preference lists in I.

Indifference in the preference lists

Another natural generalisation of SR arises when agents are permitted to express indifference in their preference lists. Partially ordered preference lists (referred to as *preference* *posets*) allow for the most general form of indifference. We let SRP denote the extension of SR in which preference lists are partially ordered. A special case of SRP arises when indifference takes the form of ties in the preferences lists (i.e. the relation "is indifferent between" is transitive) – we refer to this restriction as SRT (Stable Roommates with Ties). Three stability criteria have been defined in the context of the Stable Marriage problem with Ties (henceforth SMT) [15] and these definitions have been generalised to SRP [17]. Under the weakest of these three, a matching M in an SRP instance I is *weakly stable* if there is no pair of mutually acceptable agents $\{a_i, a_j\}$, each of whom is either unmatched in M or prefers the other to his partner in M. Ronn [25] showed that the problem of deciding whether an SRT instance I admits a weakly stable matching is NP-complete, even if I contains no incomplete lists, each list has at most one tie, and each tie is of length 2. Irving and Manlove [17] gave a shorter proof of this result, for the same problem restrictions. For the SMT case, it is known that weakly stable matchings must exist but can be of different sizes, and each of the problems of finding a maximum or minimum weakly stable matching is NP-hard, for similar restrictions on the positions and lengths of ties [23].

According to the strongest of the three stability criteria, a matching M in an SRP instance I is *super-stable* if there is no pair of mutually acceptable agents $\{a_i, a_j\}$, each of whom is either unmatched or prefers the other to his partner in M or is indifferent between them. The following proposition, whose proof is straightforward and is omitted, gives a necessary and sufficient condition for M to be super-stable in I.

Proposition 1.1. *Let M be a matching in an instance I of* SRP*. Then M is super-stable in I if and only if M is stable in every instance of* SR *obtainable by creating a linear extension of each preference poset in I.*

A super-stable matching need not exist, given an instance of SMT [15]. However Irving and Manlove [17] gave an $O(m)$ algorithm, Algorithm SRT-super, that finds a super-stable matching in an SRT instance I or reports that none exists. They also indicated how to generalise this algorithm to the case that I is an instance of SRP. Algorithm SRT-super is an extension of Phase 1 of Irving's algorithm for SR [13], and an algorithm that finds a super-stable matching or reports that none exists, given an instance of SMT [15]. It is also known that, for a given SRP instance I, the same set of agents are matched in all super-stable matchings in I [17].

A third form of stability that lies "in between" weak stability and super-stability is *strong stability.* A matching M in an SRP instance I is *strongly stable* if there is no pair of mutually acceptable agents $\{a_i, a_j\}$, such that a_i is either unmatched in M or prefers a_j to his partner in M, and a_j is either unmatched in M or prefers a_i to his partner in M or is indifferent between them. Again a strongly stable matching need not exist, given an instance of SMT [15]. Clearly a super-stable matching is strongly stable, and a strongly stable matching is weakly stable. Scott [29] gave an $O(m^2)$ algorithm that finds a strongly stable matching in I or reports that none exists, if I is an instance of SRT. This algorithm is an extension of Algorithm SRT-super [17], and an algorithm that finds a strongly stable matching or reports that none exists, given an instance of SMT [15]. However the problem of deciding whether an instance I of SRP admits a strongly stable matching is NP-complete [18] (this result holds even if I is an instance of SMP, i.e. SM with partially ordered preference lists).

The most natural form of indifference arises when preference posets can be expressed in terms of ties. However Fishburn [7] discusses practical situations in which more general preference structures, including arbitrary partial orders, may be appropriate. One particular scenario in which super-stability in the context of SRP or SMP is relevant is when there is uncertainty in the preference structures. Suppose that, in an SR instance, we wish to find a stable matching, but for some or all of the agents we have only partial information regarding preferences. In general, each preference "list" may be expressible only as a partial order, and the particular linear extension that represents an agent's true preferences is unknown. Therefore in view of Proposition 1.1, a super-stable matching is one that is stable no matter which linear extensions of the preference posets represent the true preferences.

Forbidden pairs

Recently Dias et al. [2] introduced the concept of *forbidden pairs* to an instance I of SM. A set of forbidden pairs is a set F of (man,woman) pairs, none of which is permitted to belong to any stable matching in I. That is, a matching M is *stable* if M admits no blocking pair, and $M \cap F = \emptyset$. Note that forbidden pairs can be blocking pairs, so they cannot simply be declared as unacceptable partners. We denote by SMF the generalisation of SM in which an instance may contain forbidden pairs. One motivation for considering SMF is that an administrator of a centralised matching scheme may, for whatever reason, wish to prevent a particular pairing from being returned in a constructed matching M. However the agents concerned may nevertheless wish to become matched to one another, so could potentially form a blocking pair and therefore undermine M. Dias et al. [2] gave an $O(m)$ algorithm that finds a stable matching or reports that none exists, given an instance of SMF.[1] Clearly the notion of forbidden pairs may be applied to the other stable matching problems SMP, SR and SRP considered here, giving SMPF, SRF and SRPF respectively (with analogous extensions when indifference takes the form of ties).

Contribution of this paper

Let I be an instance of SMPF. In this paper we establish the following results:

1. an $O(m)$ algorithm to find a "succinct certificate", using $O(n)$ space, for the unsolvability (i.e. the non-existence of a super-stable matching) of I;

2. an $O(m)$ algorithm to find a super-stable matching in I or report that none exists;

3. an $O(m)$ algorithm to find a *minimum regret* super-stable matching in I – this is a super-stable matching that minimises the maximum rank of an agent's partner;

4. an $O(m^{1.5})$ algorithm to find an *egalitarian* super-stable matching in I – this is a super-stable matching that minimises the sum of the ranks of the agents' partners;

5. an $O(m)$ algorithm to find all the *super-stable pairs* (i.e. all the pairs that belong to some super-stable matching) in I;

6. an algorithm to list all the super-stable matchings in I: the first super-stable matching can be output in $O(m)$ time, and each subsequent super-stable matching can be output in $O(n)$ time.

We also extend the algorithms under Items 1-3 and 5-6 to the case that I is an instance of SRPF. In this case, the complexity of the algorithms for each of Items 1-3 and 5 changes to $O(nm)$, as does the complexity of generating the first super-stable matching in the case of Item 6.

[1] Dias et al. [2] considered the version of SM where there are no incomplete lists, however it is straightforward to extend their results to the case of incomplete lists.

Our results are established via a two-step reduction from SRPF under super-stability to 2-SAT. The first reduction shows that an instance I of SRPF may be transformed in $O(nm)$ time to an instance J of SRF, with the property that the super-stable matchings in I are in 1-1 correspondence with the stable matchings in J. This reduction may be carried out in $O(m)$ time if I is an instance of SMPF. The second reduction extends earlier work of Feder [5, 6], who showed that an SR instance J may be transformed in $O(m)$ time to an instance K of 2-SAT, such that the stable matchings in J are in 1-1 correspondence with the satisfying truth assignments for K. Here we extend this reduction to the case that J is an instance of SRF. In this setting, K may be constructed from J in $O(nm)$ time; the complexity improves to $O(m)$ if J is an instance of SMF. The aforementioned reduction from an instance I of SRPF to an instance J of SRF holds in particular if I is an instance of SRT. We give a third reduction, in the opposite direction, showing that the problem of finding a super-stable matching, given an instance of SRT, is polynomial-time equivalent to the problem of finding a stable matching, given an instance of SRF.

Previous work

Conway (see Knuth [19]), Irving and Leather [16] and Feder [5, 6] established some important structural properties relating to stable matchings in an SM instance, leading to a number of efficient algorithms for the problems listed under Items 3-6 for the SM case, without altering the time complexities [9, 6]. Moreover both the algorithms for Items 5 and 6 have been extended to the SMF case [2], again without change to the time complexities.

Similarly Gusfield [10] and Irving [14] (see also [11, Chapter 4]), Feder [5, 6], Subramanian [31] and Tan [32] explored structural aspects of SR, and the exploitation of this structure has led to a number of efficient algorithms for problems concerned with finding stable matchings in an SR instance. These include an $O(m)$ algorithm for Item 1 (in this case the succinct certificate is referred to as a *stable partition*) [32], an $O(m)$ algorithm for Item 3 [11, Section 4.4.3], an $O(nm)$ algorithm for Item 5 [6] and an algorithm for Item 6 with the same time complexity as that indicated in Item 6 [6].

It is known that $\Omega(m)$ is a lower bound for each of the problems of finding a stable matching, and deciding whether a given (man,woman) pair is stable, given an SM instance [24]. Hence the algorithms for SM (and hence SMPF) listed under Items 2, 3 and 5 are optimal. The same lower bound, together with a discussion in [2], shows that the algorithm for SM in Item 6 is also optimal. In the context of SM, a stable partition is a stable matching, and hence the algorithm for SM in Item 1 is also optimal. Since SM is a special case of SR, it is immediate that the algorithms for SR under Items 1, 2, 3 and 6 are also optimal. Feder [6] states that $O(m)$ algorithms for the problems in Item 4 (even in the case of SM) and in Item 5 (even in the case of SR) are unlikely, due to an inherent dependency on transitive closure. Regarding Item 4 in the context of SR, Feder [5] shows that the problem of finding an egalitarian stable matching is NP-hard.

The distributive lattice structure for the set of stable matchings in an instance of SM, which is at the heart of the efficient algorithms listed above for SM, has also been shown to hold for SMT under super-stability [30, 22]. Scott [29] exploited this structure in order to give several algorithmic results for an SMT instance. These include an $O(m)$ algorithm for Item 3, an $O(m^2)$ algorithm for Item 4, an $O(m^2)$ algorithm for Item 5, and an algorithm for Item 6, whose complexity is as follows: after $O(m^2)$ pre-processing time, each super-stable matching can be output in $O(m)$ time.

Until now it has remained open to extend the algorithms listed under Items 1-6 to the SMP or SMPF cases, to extend the algorithms listed under Items 1, 3 and 4 to the SMF case, and to extend the algorithms in Items 1-3 and 5-6 to the SRF, SRP or SRPF cases. In particular, the algorithms under Items 3-6 for SMPF improve those of Scott [29] for SMT.

Organisation of the paper

The remainder of this paper is organised as follows. In Section 2 we give some preliminary definitions of notation and terminology used throughout this paper, including formal definitions of SR, SM and their variants. In Section 3, we present the reduction from an SRPF instance to an SRF instance. The reduction from SRF to SRT is presented in Section 4. Next, we present the reduction from SRF to 2-SAT in Section 5. A number of algorithmic consequences of the reductions in Sections 3 and 5 are given in Section 6. Finally, Section 7 contains some concluding remarks and open problems.

2 Preliminaries

An instance I of SR comprises a pair $\langle A, \prec \rangle$, where $A = \{a_1, a_2, \ldots, a_n\}$ is a set of *agents*, and $\prec$ is a set of linear orders over subsets of A, defined as follows. Each $a_i \in A$ has a non-empty *acceptable* set of agents $A_i \subseteq A \backslash \{a_i\}$. We assume that the acceptable sets are *consistent*, i.e. $a_j \in A_i$ if and only if $a_i \in A_j$. We also assume that $\prec = \{\prec_{a_i} : a_i \in A\}$, where $\prec_{a_i}$ is a strict linear order over the agents in A_i, referred to as a_i's *preference list.* If $a_j \prec_{a_i} a_k$ for two agents $a_j, a_k \in A$, we say that a_i *prefers* a_j to a_k. Given $a_i \in A$, if $A_i = A \backslash \{a_i\}$ then we refer to a_i's preference list as *complete*, otherwise it is *incomplete.*

Define the *underlying graph* of I to be an undirected graph $G = (A, E)$, where $E = \{\{a_i, a_j\} : a_i \in A \wedge a_j \in A_i\}$. Henceforth we let $m = |E|$ (so that m equals half the total length of the preference lists in I). A *matching* M in I is defined to be a matching in G; hence M comprises mutually acceptable pairs of agents. If $\{a_i, a_j\} \in M$, we define $M(a_i) = a_j$. A pair $\{a_i, a_j\} \in E \backslash M$ *blocks* M, or forms a *blocking pair* of M, if a_i is ummatched in M or prefers a_j to $M(a_i)$, and similarly a_j is unmatched in M or prefers a_i to $M(a_j)$. A matching is *stable* if it admits no blocking pair.

An instance I of SRP is defined analogously to the SR case, with the distinction being that $\prec_{a_i}$ is a strict partial order over A_i, for each $a_i \in A$, referred to as a_i's *preference poset.* For any $a_i \in A$, and for any distinct $a_j, a_k \in A_i$, if $a_j \not\prec_{a_i} a_k$ and $a_k \not\prec_{a_i} a_j$, then we say that a_i is *indifferent between* a_j *and* a_k, denoted by $a_j \approx_{a_i} a_k$. Define the relation $\preceq_{a_i} = \prec_{a_i} \cup \approx_{a_i}$. Then I is an instance of SRT if $\preceq_{a_i}$ is a preorder (i.e. a reflexive and transitive relation) for each $a_i \in A$. (That is, each preference poset can be represented in terms of a preference list with ties, since the "is indifferent between" relation is transitive.) Throughout this paper, for the purposes of algorithmic complexity arguments, we assume that, in order to represent an instance of SRP, each partial order $\prec_{a_i}$ is given in terms of the adjacency lists of a digraph that represents the transitive reduction of $\prec_{a_i}$ as given by a Hasse diagram.

The restriction SM of SR arises when G is bipartite. The definitions of the variants SMP and SMT of SM are analogous to SRP and SRT respectively.

An instance I of SRPF comprises a triple $\langle A, \prec, F \rangle$, where A and $\prec$ are as defined for the SRP case, and $F \subseteq E$. A pair of agents $\{a_i, a_j\} \in E \backslash M$ is said to *block* M, or to be a *blocking pair* of M, if either a_i is unmatched or $a_j \preceq_{a_i} M(a_i)$, and either a_j is unmatched or $a_i \preceq_{a_j} M(a_j)$. A matching M in I is *super-stable* if (i) $M \cap F = \emptyset$, and (ii) M admits no blocking pair. If M is a super-stable matching in I and $\{a_i, a_j\} \in M$, then $\{a_i, a_j\}$ is a *super-stable pair*, and a_j is a *super-stable partner* of a_i. The extensions of the definitions of SR, SRT, SM, SMP and SMT to include the possibility of forbidden pairs in an instance are analogous; these problems are denoted by SRF, SRTF, SMF, SMPF and SMTF respectively. In instances of SRF or SMF, we shorten the term *super-stable* to *stable*.

Given an SRPF instance $I = \langle A, \prec, F \rangle$, and agents $a_i \in A$ and $a_j \in A_i$, we define $rank_{\prec_{a_i}}(a_j) = 1 + |\{a_k \in A_i : a_k \prec_{a_i} a_j\}|$. For a matching M in I, define $A_M \subseteq A$ to be the set of agents who are matched in M. Let M be a super-stable matching in I and define $r_{\prec}(M) = \max_{a_i \in A_M} rank_{\prec_{a_i}}(M(a_i))$. Then M is a *minimum regret super-stable matching* if $r_{\prec}(M)$ is minimum over all super-stable matchings in I. We now suppose that weights are defined on the edges of the underlying graph of I. That is, suppose that, for each $a_i \in A$, $wt_{\prec_{a_i}} : A_i \longrightarrow \mathbb{R}$ is a given function. Define $w_{\prec}(M) = \sum_{a_i \in A_M} wt_{\prec_{a_i}}(M(a_i))$. Then M is an *optimal super-stable matching* if $w_{\prec}(M)$ is minimum over all super-stable matchings in I. In the case that $wt_{\prec_{a_i}}(a_j) = rank_{\prec_{a_i}}(a_j)$ for each $a_i \in A$ and $a_j \in A_i$, we refer to M as an *egalitarian super-stable matching*.

3 Reduction from SRPF under super-stability to SRF

Let $I = \langle A, \prec, F \rangle$ be an instance of SRPF. In this section we show how to form an instance $J = \langle A, \prec', F' \rangle$ of SRF such that the super-stable matchings in I are in 1-1 correspondence with the stable matchings in J. Firstly let $\prec''$ be formed by creating an arbitrary linear extension of each preference poset in $\prec$. This may be carried out in $O(m)$ time using a topological sorting algorithm. We create an SR instance $K = \langle A, \prec'' \rangle$ by ignoring the set F of forbidden pairs. If K is unsolvable then so is I by Proposition 1.1, so we may set $\prec' = \prec''$ and $F' = \emptyset$. Hence assume that K is solvable and calculate the stable pairs of K in $O(nm)$ time [6] ($O(m)$ time if I is an instance of SMPF [9]). Given any $a_i \in A$, let S_i denote the set of stable partners of a_i in K. Clearly by Proposition 1.1, any super-stable partner of a_i belongs to S_i. The following lemma restricts the potential super-stable partners of a_i in S_i.

Lemma 3.1. *Let the* SRPF *instance* $I = \langle A, \prec, F \rangle$ *and the* SR *instance* $K = \langle A, \prec'' \rangle$ *be as defined above, and let* $a_i \in A$. *Suppose that* $a_j, a_k \in S_i$, *where* $a_j \approx_{a_i} a_k$ *and* $a_j \prec''_{a_i} a_k$. *Then* a_j *cannot be a super-stable partner of* a_i *in* I. [2]

Proof. Suppose that M is a super-stable matching in I containing $\{a_i, a_j\}$. Then M is stable in K by Proposition 1.1. Also by assumption there exists a matching M' containing $\{a_i, a_k\}$ that is stable in K. Since $a_j = M(a_i) \prec''_{a_i} M'(a_i) = a_k$, it follows by [11, Lemma 4.3.9] that $a_i = M'(a_k) \prec''_{a_k} M(a_k)$. Hence in I, $a_i \preceq_{a_k} M(a_k)$. Also by assumption, $a_k \approx_{a_i} M(a_i)$. Hence $\{a_i, a_k\}$ blocks M in I, a contradiction. □

We use Lemma 3.1 as the basis of a modified topological sorting algorithm that creates a second linear extension of each preference poset in $\prec$, in order to form an instance $J = \langle A, \prec', F' \rangle$ of SRF. This algorithm is shown in Figure 1 as Algorithm Linear-ext. Intuitively, for any agents $a_i, a_j, a_k \in A$, if a_j is a super-stable partner of a_i in I and $a_k \approx_{a_i} a_j$, then the algorithm resolves the indifference in J such that $a_k \prec'_{a_i} a_j$.

Let $a_i \in A$. As mentioned in Section 2, we assume that $\prec_{a_i}$ is represented in I by the adjacency lists of a digraph D_i corresponding to the transitive reduction of $\prec_{a_i}$. Algorithm Linear-ext creates an ordered list L_i that represents the constructed linear extension $\prec'_{a_i}$ of $\prec_{a_i}$. Initially each agent $a_j \in A_i$ is set as *marked* or *unmarked* according as a_j belongs to S_i or not. A marked agent can subsequently become unmarked, but not vice versa. Also we maintain a counter $c(a_j)$ for each agent $a_j \in A_i$; initially $c(a_j)$ is set to be the indegree of a_j in D_i. A dequeue of agents Q is maintained, containing *source vertices* (i.e. agents $a_j \in A_i$ such that $c(a_j) = 0$). Initially each source vertex is added to Q at the front or rear according to the cases shown in the *add-to-Q* subroutine. During the execution of Algorithm Linear-ext, the following two invariants hold: (i) a_i is indifferent among all agents in Q, and (ii) either all agents in Q are unmarked, or a single agent a_k at the rear of Q is marked, where a_k is the worst agent in $Q \cap S_i$ according to a_i's preferences in K.

[2] In the case that I is an instance of SMTF, this result corresponds to Lemma 6.2.2 of [29].

```
Linear-ext(agent a_i) {
    Q := ⟨⟩;
    L_i := ⟨⟩;
    for (each agent a_j ∈ A_i in arbitrary order) {
        if (a_j ∈ S_i)
            set a_j to be marked;
        else
            set a_j to be unmarked;
        c(a_j) := indegree(a_j);
        if (c(a_j) = 0)
            add-to-Q(a_j);
    }
    while (Q ≠ ⟨⟩) {
        remove an agent a_j from the front of Q;
        append a_j to L_i;
        for (each agent a_k adjacent from a_j in D_i) {
            c(a_k) := c(a_k) − 1;
            if (c(a_k) = 0)
                add-to-Q(a_k);
        }
    }
}

add-to-Q(a_j) {
    if (a_j is marked)
        if (no marked agent is at the rear of Q)
            add a_j to the rear of Q;
        else if (some marked a_k is at the rear of Q)
            if (a_k ≺''_{a_i} a_j) {
                set a_k as unmarked;
                add a_j to the rear of Q; }
            else
                set a_j as unmarked;
                add a_j to the front of Q; }
    else
        add a_j to the front of Q;
}
```

Figure 1: Algorithm Linear-ext for creating a linear extension $\prec'_{a_i}$ of $\prec_{a_i}$.

With a suitable choice of data structures (including *ranking arrays* [11, Section 1.2.4], which may be constructed from the preference lists in K using $O(m)$ time, allowing one to decide in $O(1)$ time whether $a_k \prec''_{a_i} a_j$), it is straightforward to verify that Algorithm Linear-ext may be implemented to run in $O(m)$ overall time. At the termination of the algorithm's execution for the given agent a_i, let C_i denote the set of marked agents. Also, let $\prec' = \{\prec'_{a_i} : a_i \in A\}$.

The following lemma indicates the significance of marked agents relative to the linear orders constructed by Algorithm Linear-ext.

Lemma 3.2. *Let the* SRPF *instance $I = \langle A, \prec, F \rangle$ and the set of linear orders $\prec'$ be as defined above, and let $a_i \in A$. Then:*

(i) if $a_j \notin C_i$ then a_j cannot be a super-stable partner of a_i in I;

(ii) if $a_j \in C_i$ and $a_k \approx_{a_i} a_j$, then $a_k \prec'_{a_i} a_j$.

Proof. (i). Suppose $a_j \notin C_i$. If $a_j \notin S_i$ then a_j cannot be a super-stable partner of a_i in I by Proposition 1.1. Hence suppose that $a_j \in S_i$, so that a_j was marked when the first for loop considered a_j. Then a_j became unmarked subsequently during the algorithm's execution. This occurred during some call to the *add-to-Q* subroutine – just before this call returns, there is some marked agent a_k in Q such that $a_j \prec''_{a_i} a_k$. Now $a_j \approx_{a_i} a_k$, for otherwise a_j and a_k cannot both be source vertices. The result follows by Lemma 3.1.

(ii). As a_j is marked at the termination of the algorithm's execution, a_j must have been added to the rear of Q when it became a source vertex. Furthermore, a_j must remain at the rear of Q until it is removed from Q, for otherwise a_j would become unmarked. In particular, when a_j is removed from Q, it is the only source vertex. Since $a_k \approx_{a_i} a_j$, it follows that a_k must be (added to and) removed from Q before a_j is removed from Q. Hence by construction of $\prec'_{a_i}$, it follows that $a_k \prec'_{a_i} a_j$. □

Now define the SR instance $J' = \langle A, \prec' \rangle$. The next step is to calculate the stable pairs of J' in $O(nm)$ time [6] ($O(m)$ time if I is an instance of SMPF [9]). Given any agent $a_i \in A$, let T_i denote the set of stable partners of a_i in J', and let $F_i = T_i \backslash C_i$. Let $F' = \bigcup_{a_i \in A} \{\{a_i, a_j\} : a_j \in F_i\} \cup F$ be the set of forbidden pairs in J. Finally, define the SRF instance $J = \langle A, \prec', F' \rangle$. The following result indicates the relationship between super-stable matchings in I and stable matchings in J.

Theorem 3.3. *Given an* SRPF *instance $I = \langle A, \prec, F \rangle$, we may construct in $O(nm)$ time an* SRF *instance $J = \langle A, \prec', F' \rangle$ as defined above. Moreover, if M is a matching in I, then M is super-stable in I if and only if M is stable in J. If I is an instance of* SMPF *then J is an instance of* SMF*, and may be constructed in $O(m)$ time.*

Proof. Suppose that M is super-stable in I. Then $M \cap F = \emptyset$. Now let $\{a_i, a_j\} \in M$. Then $a_j \in C_i$ by Lemma 3.2(i). Hence $M \cap F' = \emptyset$. Moreover M is stable in J by Proposition 1.1.

Conversely suppose that M is stable in J. Then $M \cap F = \emptyset$. Now suppose that $\{a_i, a_j\}$ is a blocking pair of M in I. If a_i is matched in M and $a_j \prec_{a_i} M(a_i)$, then $a_j \prec'_{a_i} M(a_i)$, since $\prec'_{a_i}$ is a linear extension of $\prec_{a_i}$. Now suppose that $a_j \approx_{a_i} M(a_i)$. Since $M \cap F' = \emptyset$, it follows that $M(a_i) \in C_i$. Hence by Lemma 3.2(ii), it follows that $a_j \prec'_{a_i} M(a_i)$. By a similar argument it follows that a_j is unmatched in M or $a_i \prec'_{a_j} M(a_j)$. Hence $\{a_i, a_j\}$ blocks M in J, a contradiction. Thus M is super-stable in I.

Finally we remark that the last sentence in the statement of the theorem is an immediate consequence of the fact that the reduction preserves bipartiteness. □

Consequences of the reduction for SMPF

For the case that I is an instance of SMPF, the following theorem presents a number of algorithmic results that hold as a consequence of analogous results in the context of SMF [2] and Theorem 3.3.

Theorem 3.4. *Let I be an instance of* SMPF*. Then:*

(i) There is an $O(m)$ algorithm that finds a super-stable matching in I or reports that none exists. (If I is an instance of SMT*, see also [15, 21].)*

(ii) There is an $O(m)$ algorithm for finding all the super-stable pairs in I.

(iii) There is an algorithm for listing all the super-stable matchings in I: the first super-stable matching can be output in $O(m)$ time, and each subsequent super-stable matching can be output in $O(n)$ time.

4 Reduction from SRF to SRT under super-stability

In this section we consider a counterpart to the reduction presented in Section 3. Theorem 3.3 shows in particular that the problem of finding a super-stable matching if one exists, given an instance I of SRT, may be reduced in $O(nm)$ time to the problem of finding a stable matching if one exists, given an instance J of SRF. It turns out that, by modifying a reduction of Cechlárová and Fleiner [1, Theorem 2.1], we may formulate a reduction in the opposite direction, as the following result indicates.

Theorem 4.1. *Given an instance I of* SRF*, we may construct in $O(m)$ time an instance J of* SRT *such that a super-stable matching in J can be derived in $O(m)$ time from a stable matching in I, and vice versa.*

Proof. Let $I = \langle A, \prec, F\rangle$ be an instance of SRF. We form an instance $J = \langle A', \prec'\rangle$ of SRT as follows. Initially let $A' = A$ and $\prec' = \prec$. Suppose that

$$F = \{\{a_{i_k}, a_{j_k}\} : i_k < j_k \wedge 1 \le k \le r\}.$$

For each k ($1 \le k \le r$), add the new agents w_k, x_k, y_k, z_k to A', replace a_{j_k} by w_k in $\prec'_{a_{i_k}}$, and replace a_{i_k} by z_k in $\prec'_{a_{j_k}}$. The preference lists in $\prec'$ for the newly-introduced agents in A' are as follows:

$$\begin{array}{rlcrl} w_k: & z_k \ (x_k \ \ a_{i_k}) & \qquad & x_k: & w_k \ \ y_k \\ y_k: & x_k \ \ z_k & & z_k: & y_k \ (w_k \ \ a_{j_k}) \end{array}$$

In a given preference list, agents are listed from left to right in decreasing order of preference, and agents within brackets are tied.

Suppose that M is a stable matching in I. Then $M \cap F = \emptyset$. Initially let $M' = M$. Now let k ($1 \le k \le r$) be given. Either (i) a_{i_k} is matched in M and prefers $M(a_{i_k})$ to a_{j_k} in I, or (ii) a_{j_k} is matched in M and prefers $M(a_{j_k})$ to a_{i_k} in I, for otherwise $\{a_{i_k}, a_{j_k}\}$ blocks M in I. In Case (i), add the pairs $\{w_k, x_k\}, \{y_k, z_k\}$ to M', whilst in Case (ii), add the pairs $\{w_k, z_k\}, \{x_k, y_k\}$ to M'. It may be verified that the matching M' so constructed is a super-stable matching in J.

Conversely suppose that M' is a super-stable matching in J. We firstly note that, for each k ($1 \le k \le r$), $\{a_{i_k}, w_k\} \notin M'$, for otherwise $\{w_k, x_k\}$ blocks M' in J, and similarly $\{a_{j_k}, z_k\} \notin M'$, for otherwise $\{w_k, z_k\}$ blocks M' in J. Let $G = (A, E)$ be the underlying graph of I, and let $M = M' \cap E$. Clearly $M \cap F = \emptyset$. Let $\{a_i, a_j\} \in E \backslash M$. If $\{a_i, a_j\} \notin F$, then clearly this pair cannot block M in I, by the super-stability of M' in J. Now suppose that $\{a_i, a_j\} \in F$. Without loss of generality assume that $\{a_i, a_j\} = \{a_{i_k}, a_{j_k}\}$ for some k ($1 \le k \le r$). Suppose that a_{i_k} is unmatched in M or prefers a_{j_k} to $M(a_{i_k})$ in I. Then since $\{a_{i_k}, w_k\} \notin M'$, it follows that a_{i_k} is unmatched in M' or prefers w_k to $M'(a_{i_k})$ in J. Hence by the super-stability of M' in J, it follows that $\{w_k, z_k\} \in M'$, so that in turn, a_{j_k} is matched in M' and prefers $M'(a_{j_k})$ to z_k. As previously observed, $\{a_{j_k}, M'(a_{j_k})\} \in M$, so that $\{a_{i_k}, a_{j_k}\}$ does not block M in I. Hence M is stable in I. □

The following observation is an immediate consequence of the above reduction and Algorithm SRT-super [17].

Corollary 4.2. *There is an $O(m)$ algorithm that finds a stable matching or reports that none exists, given an instance of* SRF*.*

We remark that if I is an instance of SMF then J is an instance of SMT, since the above reduction preserves bipartiteness.

5 Reduction from SRF to 2-SAT

In this section, we show how to modify the constructions of Feder [5, 6] in order to obtain a reduction from SRF to 2-SAT. We first review the relationship between the set of all stable matchings for an instance I of SR and the so-called *rotations* for that instance.

Irving's algorithm for SR [13, 11] consists of two phases. The first phase is analogous to an extended form of the classical Gale-Shapley algorithm for SM [8]; it involves a sequence of "proposals" from one agent a_i to the first agent a_j on his list, each such proposal resulting in the deletion of all successors of a_i from a_j's list. (Here, and subsequently, the deletion of a_k from the list of a_j implies the deletion of a_j from the list of a_k.) On termination of this phase, the (reduced) preference lists form an example of what is called a *stable table* [11, p.169]; among the properties of such a table are that all first entries are distinct, and that a_j is first in a_i's list if and only if a_i is last in that of a_j.

A *rotation* exposed in a stable table T is a sequence $(a_{i_0}, a_{j_0}), \ldots, (a_{i_{r-1}}, a_{j_{r-1}})$ of pairs such that a_{j_k} is first and $a_{j_{k+1}}$ second in a_{i_k}'s list in T, for each k $(0 \leq k \leq r-1)$, where arithmetic with respect to rotations is taken modulo r. *Elimination* of the rotation involves deleting all successors of $a_{i_{k-1}}$ from the list of a_{j_k}, for each k $(0 \leq k \leq r-1)$. A key result is that, provided no list becomes empty, the elimination of an exposed rotation from a stable table gives another (smaller) stable table. Phase 2 of the algorithm consists of the successive elimination of rotations from the current stable table until either some list becomes empty as a result, in which case no stable matching exists, or all lists that were non-empty after phase 1 are reduced to a single entry, in which case these entries constitute a stable matching. In what follows of this section, we assume that I is *solvable*, i.e. I admits a stable matching. At the termination of phase 1, we may identify the *fixed pairs* of I – these are the stable pairs that belong to every stable matching in I. A pair $\{a_i, a_j\}$ is a fixed pair if and only if a_i's list contains only a_j at the termination of phase 1 [11, Lemma 4.4.1].

Suppose that $\rho = (a_{i_0}, a_{j_0}), \ldots, (a_{i_{r-1}}, a_{j_{r-1}})$ is a rotation that is exposed in some stable table. The *syntactic dual* of ρ is $\bar{\rho} = (a_{j_1}, a_{i_0}), \ldots, (a_{j_0}, a_{i_{r-1}})$. If there is some sequence of rotations that leads to a stable table in which $\bar{\rho}$ is exposed, then $\bar{\rho}$ is also a rotation; in this case ρ and $\bar{\rho}$ are called *non-singular* rotations, and are *duals* of each other, otherwise ρ is *singular*. (Hence the syntactic dual of a singular rotation is not actually a rotation at all.) A partial order $\lhd$ is defined on the set of rotations as follows: $\rho \lhd \sigma$ if and only if ρ must be eliminated to give a stable table in which σ is exposed. The rotations under $\lhd$ form the *rotation poset* for I. A subset S of this poset is *closed* if, whenever ρ is in S, so also is every rotation σ such that $\sigma \lhd \rho$. Also S is *complete* if S contains every singular rotation of I, together with exactly one of each dual pair of non-singular rotations. The following theorem encapsulates the relationship between the rotation poset and the set of all stable matchings in I.

Theorem 5.1 ([10, 11]). *Let I be a solvable* SR *instance. There is a 1-1 correspondence between the stable matchings in I and the complete closed subsets of the rotation poset of I.*

The so-called *extended rotation poset* R_I^* for I contains all the rotations together with the syntactic duals of the singular rotations, and restricting this structure by excluding these latter elements gives the actual rotation poset. We can find, in $O(m)$ time, a directed graph R_I that represents R_I^*, in the sense that the transitive closure of R_I is isomorphic to R_I^* (see [11, Section 4.4.1]). The digraph R_I is constructed by scanning each preference list in turn, adding a sequence of edges derived from the rotations represented in that list (see [11, Section 4.4.1]). As a consequence, the *explicit width* of R_I is at most n, meaning that we can find a set of at most n vertex-disjoint paths in R_I that cover all the vertices – one such path arises from each preference list.

The digraph R_I turns out to be equivalent to the implication digraph of an instance J of acyclic 2-SAT. In J, each variable and its negation correspond to a rotation and its syntactic dual. The clauses of J are of the form $(\rho \vee \bar{\sigma})$ for any pair of rotations such that (ρ, σ) is an edge in R_I (which implies that ρ precedes σ in R_I^*). Because a singular rotation precedes its syntactic dual in R_I^*, the singular rotations are precisely the trivial variables in J – i.e., those that are true in every satisfying truth assignment. Hence the true variables in any satisfying truth assignment for J correspond to a complete closed set of rotations in I. The converse is also true, so by Theorem 5.1 there is a 1-1 correspondence between satisfying truth assignments for J and stable matchings in I.

The implication digraph D of J has a vertex for each literal and a directed edge (σ, ρ) if $(\rho \vee \bar{\sigma})$ is a clause in J. So, in fact, D is structurally identical to R_I, except that the direction of every edge is reversed.

Feder [6] has established that we can construct in $O(nm)$ time a representation of the transitive closure D^* of D, which enables us to test in $O(1)$ time whether a given edge is in D^* or not. This allows the singular rotations to be identified, since a rotation ρ is singular if and only if $(\bar{\rho}, \rho) \in D^*$. In turn, this allows the stable pairs of I to be found, since these are precisely the (disjoint) union of the fixed pairs and the pairs that are in some non-singular rotation [11, Lemma 4.4.1]. Furthermore, Feder [6] shows that the stable matchings of I may be listed efficiently by considering the digraph D. The following result gives two consequences that arise from the discussion so far.

Theorem 5.2 ([6]). *Let I be an instance of* SR *and let J be the instance of 2-*SAT *as described above. Then:*

(i) The stable pairs for I can be found in $O(nm)$ time.

(ii) The satisfying truth assignments for J, and therefore the stable matchings for I, can be listed in $O(n)$ time per solution, after $O(m)$ pre-processing time.

Now let $K = \langle A, \prec, F \rangle$ be an instance of SRF, and let $I = \langle A, \prec \rangle$ be the instance of SR obtained from K by ignoring the forbidden pairs in F. We observe first that forbidden pairs in F that are not stable pairs of I have no effect and can be ignored. Also if any pair in F is a fixed pair of I, clearly K is unsolvable. Hence we assume that no pair in F is a fixed pair of I. As described above, let D be the implication digraph of the 2-SAT instance representing I. We show how to extend D so as to obtain a reduction from K to an instance J of 2-SAT. Corresponding to each forbidden pair that is a stable pair of I, we add one or two additional edges to D, which results in two of the variables in the 2-SAT instance becoming equivalent.

Suppose the forbidden pair is $\{a_i, a_j\}$. Note that, for every stable pair $\{a_i, a_j\}$ of I that is not a fixed pair of I, either (a_i, a_j) or (a_j, a_i) is in a rotation [11, Lemma 4.4.1]. In fact, the stable pair (a_i, a_j) is in a rotation if and only if a_i is not the best stable partner of a_j (or, equivalently, a_j is not the worst stable partner of a_i). Let us suppose that (a_i, a_j) belongs to a rotation ρ of I.

Case (i). Suppose that a_j is the first stable partner (according to $\prec_{a_i}$) of a_i in I. Note that, in this case, the pair (a_j, a_i) is not in a rotation of I. Thus to avoid the presence of the pair $\{a_i, a_j\}$ in any stable matching, we have to ensure that ρ (and any predecessor rotation) is eliminated, or, in the 2-SAT context, that variable ρ is true. We do this by adding the edge $(\bar{\rho}, \rho)$ to the implication digraph D.

Case (ii). Suppose that a_k is the stable partner of a_i in I that immediately precedes a_j (according to $\prec_{a_i}$), and that $(a_i, a_k) \in \sigma$ in I. Then to avoid the presence of the pair $\{a_i, a_j\}$ in any stable matching, we have to ensure that if σ is eliminated then so is ρ, or, in the 2-SAT context, that variable ρ is true if σ is true. We do this by adding the edge (σ, ρ) to the implication digraph, thereby making variables ρ and σ equivalent – σ is true if and only if ρ is true. Note also that, since a_j is not the best stable partner of a_i, the pair (a_j, a_i) must belong to a rotation, so we must also carry out the corresponding action for this pair.

Let D be the implication digraph that arises once this process has been carried out for each pair in F. By reversing the edge directions in D, we obtain a digraph R_K that may be viewed as the analogue of R_I for K. The transitive closure R_K^* of R_K may also be viewed as the analogue of R_I^* for K, but in general R_K^* contains cycles and therefore does not correspond to a poset. Nevertheless, we may define a complete set of rotations S in I to be *closed in* R_K^* if, whenever $\rho \in S$, so also is every rotation σ such that (σ, ρ) is an edge of R_K^*. The following result is an immediate consequence of the construction of D.

Theorem 5.3. *Given an instance $K = \langle A, \prec, F\rangle$ of* SRF*, suppose that the instance $I = \langle A, \prec\rangle$ of* SR *is solvable, and no pair in F is a fixed pair of I. Then there is a 1-1 correspondence between the stable matchings in K and the complete subsets of the rotation poset of I that are closed in R_K^*.* [3]

Proof. Let M be a stable matching in K. By Theorem 5.1, M corresponds to a complete closed subset $\mathcal{Z}$ of rotations in I. As $M \cap F = \emptyset$, it is straightforward to verify that, for any rotation ρ identified by Case (i) above, $\rho \in \mathcal{Z}$, and for any rotations σ and ρ identified by Case (ii) above, if $\sigma \in \mathcal{Z}$ then $\rho \in \mathcal{Z}$. Hence $\mathcal{Z}$ is closed in R_K^*.

Conversely suppose that $\mathcal{Z}$ is a complete subset of the rotation poset of I that is closed in R_K^*. Then $\mathcal{Z}$ is closed in I, so that $\mathcal{Z}$ corresponds to a stable matching M in I by Theorem 5.1. For any rotation ρ identified by Case (i) above, $\rho \in \mathcal{Z}$, for otherwise $\bar{\rho} \in \mathcal{Z}$ as $\mathcal{Z}$ is complete, and hence $\rho \in \mathcal{Z}$ as $\mathcal{Z}$ is closed in R_K^*, a contradiction. Similarly for any rotations σ and ρ identified by Case (ii) above, if $\sigma \in \mathcal{Z}$ then $\rho \in \mathcal{Z}$, as $\mathcal{Z}$ is closed in R_K^*. Hence $M \cap F = \emptyset$, so that M is stable in K. □

The implication digraph D can be made acyclic (in $O(m)$ time) using a strong components algorithm. That is, each strongly connected component C_i is replaced by a single vertex v_i, and each edge between two vertices in distinct strongly connected components is replaced by an edge between the two corresponding representative vertices. Let J be the 2-SAT instance represented by the resulting implication digraph D'. Clearly D' still has explicit width at most n, since adding additional edges to D as described in Cases (i) and (ii) above, and coalescing vertices during the strong components algorithm, cannot increase the explicit width. We thus obtain the following result, using Theorem 5.3, together with the the fact that the true variables in any satisfying truth assignment for J correspond to a complete set of rotations in I that are closed in R_K^*.

Theorem 5.4. *Given an instance $K = \langle A, \prec, F\rangle$ of* SRF*, suppose that the instance $I = \langle A, \prec\rangle$ of* SR *is solvable, and no pair in F is a fixed pair of I. Then we may construct in $O(nm)$ time an instance J of 2-*SAT*, such that the implication digraph of J has explicit width at most n, and the stable matchings for K are in 1-1 correspondence with the satisfying truth assignments for J. If K is an instance of* SMF *then J may be constructed in $O(m)$ time.*

[3] In the case that K is an instance of SMF, this result corresponds to Theorem 6 of [2].

6 Further algorithmic results for SRPF and SMPF

In this section we present a number of algorithmic results for SRPF and SMPF that follow as a consequence of the reductions given in Sections 3 and 5, and Theorems 3.3, 5.3 and 5.4 in particular. These results add to those already given by Theorem 3.4 for SMPF.

Theorem 6.1. *Let I be an instance of* SRPF. *Then:*

(i) There is a succinct certificate, using $O(m)$ space, for the unsolvability of I, which may be computed in $O(nm)$ time ($O(m)$ time if I is an instance of SMPF*).*

(ii) There is an $O(nm)$ algorithm that finds a super-stable matching in I or reports that none exists.

(iii) There is an $O(nm)$ algorithm for finding all the super-stable pairs in I.

(iv) There is an algorithm for generating all the super-stable matchings in I: the first stable matching can be output in $O(nm)$ time, and each subsequent stable matching can be output in $O(n)$ time.

Proof. (i). Let $I = \langle A, \prec, F\rangle$ be an instance of SRPF. Let $K = \langle A, \prec', F'\rangle$ be the instance of SRF obtained from I as in Section 3, and let $I' = \langle A, \prec'\rangle$ be the instance of SR obtained from K by ignoring the pairs in F. Firstly, we note that if I' is unsolvable, the characterisation of Tan [32] gives a succinct $O(n)$ certificate for the unsolvability of I in $O(nm)$ time, by Theorem 3.3 (clearly this case cannot occur if I is an instance of SMPF). Now suppose that I' is solvable. If a pair in F is a fixed pair of I', then a succinct $O(m)$ certificate of this is the set of preference lists after phase 1 of Irving's algorithm (which has $O(m)$ complexity) applied to I'. Hence suppose that no pair in F is a fixed pair of I'. Let J be the instance of 2-SAT obtained from K as in Section 5. The unsolvability of J is characterised by a cycle, using $O(m)$ space, in the implication digraph D of J, involving a variable ρ and its negation $\bar{\rho}$. Such a cycle would be discovered by the strong components algorithm when attempting to make D acyclic following the addition of the vertices and edges to D. The result then follows by Theorems 3.3 and 5.4.

(ii). This result follows by the discussion in Case (i), together with the $O(m)$ algorithm for finding a satisfying truth assignment or reporting that none exists, given an instance of 2-SAT [3], followed by Theorems 3.3 and 5.4.

(iii). This is an immediate consequence of Theorems 3.3 and 5.4, together with [6, Theorem 6.2].

(iv). This is an immediate consequence of Theorems 3.3 and 5.4, together with [6, Theorem 8.1]. □

In the remainder of this section we consider optimal, egalitarian and minimum regret super-stable matchings in an SRPF instance $I = \langle A, \prec, F\rangle$. Let $K = \langle A, \prec', F'\rangle$ be the instance of SRF obtained from I as in Section 3. Let J be the instance of 2-SAT obtained from K as in Section 5. As an aside, we firstly remark that, for any super-stable pair $\{a_i, a_j\}$ in I, it follows from the construction of K that

$$rank_{\prec'_{a_i}}(a_j) = rank_{\prec_{a_i}}(a_j) + |\{a_k \in A_i \backslash \{a_j\} : a_k \approx_{a_i} a_j\}|.$$

Optimal and egalitarian super-stable matchings

For each of the problems of computing an optimal and egalitarian super-stable matching in I, we define a weight function $wt_{\prec'_{a_i}}$ in K for each $a_i \in A$. That is, for each $a_i \in A$ and $a_j \in A_i$, in the optimal case, define $wt_{\prec'_{a_i}}(a_j) = wt_{\prec_{a_i}}(a_j)$, and in the egalitarian case, define $wt_{\prec'_{a_i}}(a_j) = rank_{\prec_{a_i}}(a_j)$. Then an optimal stable matching in K is an optimal super-stable matching in I. It also follows that $w_{\prec}(M) = w_{\prec'}(M)$. Hence, and by Theorems 3.3, 5.4 and [6, Theorem 7.2], the following result holds concerning optimal and egalitarian super-stable matchings in the context of SMPF.

Theorem 6.2. *Let I be an instance of* SMPF*, and suppose that, for each $a_i \in A$, $wt_{\prec_{a_i}} : A_i \longrightarrow \mathbb{R}$ is a given function. Then an optimal super-stable matching in I of weight W can be found in $O(m\sqrt{W})$ time if $W = O((m/\log^2 m)^2)$, and in $O(nm \log W)$ time for arbitrary W. The egalitarian case has $W \leq m$ and hence can be solved in $O(m^{1.5})$ time.* [4]

Minimum regret super-stable matchings

We next give an algorithm for finding a minimum regret super-stable matching in an instance I of SRPF. Our algorithm is of $O(nm)$ complexity; this improves to $O(m)$ if I is an instance of SMPF. In what follows we assume that K is solvable.

Let $I' = \langle A, \prec' \rangle$ be the instance of SR obtained from K by ignoring the pairs in F. Recall from Theorems 3.3 and 5.3 that there is a 1-1 correspondence between the super-stable matchings in I and the complete subsets of the rotation poset of I' that are closed in R^*_K. We will create such a subset $\mathcal{Z}$ that corresponds to a minimum regret super-stable matching M in I. We firstly note that $\mathcal{Z}$ must contain the singular rotations of I'. Recall that the rotations in I' may be found in $O(m)$ time [11], and identified as singular or dual in $O(nm)$ time [6]. If I is an instance of SMPF then this latter step is not required, since there are no singular rotations in I'.

To facilitate the choice of non-singular rotations, we firstly compute the set of super-stable pairs in I. By Theorems 6.1 and 3.4, this step may be carried out in $O(nm)$ time ($O(m)$ time if I is an instance if SMPF). In what follows we assume there is at least one super-stable pair in I that is not a fixed pair of I, for otherwise the computation of a minimum regret super-stable matching in I is trivial. Given a rotation $\rho = (a_{i_0}, a_{j_0}), \ldots, (a_{i_{r-1}}, a_{j_{r-1}})$ in I', define

$$rank_{\prec}(\rho) = \max \left(\{0\} \cup \left\{ rank_{\prec_{a_{j_k}}}(a_{i_k}) : \begin{array}{l} 0 \leq k \leq r-1 \wedge \\ \{a_{i_k}, a_{j_k}\} \text{ is a super-stable pair in } I \end{array} \right\} \right).$$

The idea of the algorithm is to pick a non-singular rotation ρ for which $rank_{\prec}(\rho) = R$, where R is as large as possible. We then try to eliminate ρ – if this is not possible then, as we will show, $r_{\prec}(M) \geq R$ for any super-stable matching M in I. Any rotation ρ that does not contain a super-stable pair does not lead to the same condition on $r_{\prec}(M) \geq R$, which is why $rank_{\prec}(\rho)$ is defined to be 0 for such a rotation. It is straightforward to verify that $rank_{\prec}(\rho)$, for each rotation ρ in I', may be found in $O(m)$ overall time using a scan of the agents' preference lists in I'. The following lemma is crucial to our approach for finding a minimum regret super-stable matching in I.

Lemma 6.3. *Let K and I' be as defined above. Let ρ be a non-singular rotation in I' such that $rank_{\prec}(\rho) = R \geq 1$. Let $\mathcal{Z}$ be a complete subset of rotations in I' that is closed in R^*_K such that $\rho \notin \mathcal{Z}$. Let M be the stable matching in K corresponding to $\mathcal{Z}$, by Theorem 5.3. Then $r_{\prec}(M) \geq R$.*

Proof. Let $\rho = (a_{i_0}, a_{j_0}), \ldots, (a_{i_{r-1}}, a_{j_{r-1}})$. As $R \geq 1$, it follows that $rank_{\prec_{a_{j_k}}}(a_{i_k}) = R$ for some k ($0 \leq k \leq r-1$) where $\{a_{i_k}, a_{j_k}\}$ is a super-stable pair in I. Since $\rho \notin \mathcal{Z}$,

[4] In order to construct an optimal or egalitarian stable matching, Feder [6] assumes that $wt_{\prec'_{a_i}}$ is a nondecreasing function for each $a_i \in A$. However this assumption is made purely from a practical point of view, and the algorithm does not, in fact, require it.

it follows from the completeness of $\mathcal{Z}$ that $\bar{\rho} \in \mathcal{Z}$. But $\bar{\rho}$ moves a_{j_k} down to a_{i_k} in I', so that either $M(a_{j_k}) = a_{i_k}$ or $a_{i_k} \prec'_{a_{j_k}} M(a_{j_k})$. Hence in I, either (a) $M(a_{j_k}) = a_{i_k}$, or (b) $a_{i_k} \prec_{a_{j_k}} M(a_{j_k})$, or (c) $a_{i_k} \approx_{a_{j_k}} M(a_{j_k})$ by construction of K (i.e. since $\prec'_{a_{j_k}}$ is a linear extension of $\prec_{a_{j_k}}$). In Cases (a) and (b), it follows that $rank_{\prec_{a_{j_k}}}(M(a_{j_k})) \geq R$ as required. In Case (c), we obtain a contradiction to the stability of M in K, since $\{a_{j_k}, M(a_{j_k})\} \in F'$ by construction of K, as $\{a_{i_k}, a_{j_k}\}$ is a super-stable pair in I. □

The above observation gives rise to the following definition. Let $\mathcal{Z}$ be a complete subset of rotations in I' that is closed in R_K^*. Define

$$r_{\prec}(\mathcal{Z}) = \max\{rank_{\prec}(\rho) : \rho \text{ is a rotation in } I' \wedge \rho \notin \mathcal{Z}\}.$$

Then $\mathcal{Z}$ is said to be of *minimum regret* if $r_{\prec}(\mathcal{Z})$ is minimum over all complete subsets of rotations in I' that are closed in R_K^*. The following lemma establishes a link between such a subset of rotations and a minimum regret super-stable matching in I.

Lemma 6.4. *Let K and I' be as defined above. Let $\mathcal{Z}$ be a minimum regret complete subset of rotations in I' that is closed in R_K^*. Then $\mathcal{Z}$ corresponds to a minimum regret super-stable matching M in I.*

Proof. By Theorems 5.3 and 3.3, $\mathcal{Z}$ corresponds to a super-stable matching M in I. Suppose there exists a super-stable matching M' in I such that $r_{\prec}(M') < r_{\prec}(M) = R$. By Theorems 3.3 and 5.3, let $\mathcal{Z}'$ be the complete subset of rotations in I' that is closed in R_K^* that corresponds to M'. As $r_{\prec}(M) = R \geq 1$, there exists some $\{a_i, a_j\} \in M$ such that $rank_{\prec_{a_j}}(a_i) = R$. Now $\{a_i, a_j\} \notin M'$, so $\{a_i, a_j\}$ is not a fixed pair of I'. Hence by [11, Lemma 4.4.1], either (i) $(a_i, a_j) \in \rho$ or (ii) $(a_j, a_i) \in \rho$, for some rotation ρ in I'. We consider these two cases separately.

Case (i). As $(a_i, a_j) \in \rho$ and $\{a_i, a_j\}$ is a super-stable pair in I, it follows that $rank_{\prec}(\rho) \geq R$. Now $\rho \notin \mathcal{Z}$, since $\{a_i, a_j\} \in M$. Hence $r_{\prec}(\mathcal{Z}) \geq R$. Thus by the choice of $\mathcal{Z}$, it follows that $r_{\prec}(\mathcal{Z}') \geq R$. Hence there is a rotation σ in I' such that $\sigma \notin \mathcal{Z}'$ and $rank_{\prec}(\sigma) \geq R$. It follows by Lemma 6.3 that $r_{\prec}(M') \geq R$, a contradiction.

Case (ii). Suppose for a contradiction that $\rho \in \mathcal{Z}'$. Then in K, it follows that $a_i \prec'_{a_j} M'(a_j)$. Hence in I, either (a) $a_i \prec_{a_j} M'(a_j)$, or (b) $a_i \approx_{a_j} M'(a_j)$ by construction of K. In Case (a) it follows that $rank_{\prec_{a_j}}(M'(a_j)) > R$, a contradiction. In Case (b) we also obtain a contradiction, since $\{a_j, M'(a_j)\} \in F'$ by construction of K, as $\{a_i, a_j\}$ is a super-stable pair in I. Hence $\rho \notin \mathcal{Z}'$. It follows that $\bar{\rho} \in \mathcal{Z}'$. Rotation $\bar{\rho}$ moves a_i down to a_j in I'. Hence as $\{a_i, a_j\} \notin M'$, it follows that $(a_i, a_j) \in \pi$ for some $\pi \in \mathcal{Z}'$. The remainder of this case is similar to the proof of Case (i). □

Define the *cost* of a literal in J to be the rank of the corresponding rotation. By Lemma 6.4 and Theorem 5.4, it follows that finding a minimum regret super-stable matching in I is equivalent to finding a satisfying truth assignment in J that minimises the maximum cost of a false literal. It is straightforward to find such an assignment in J in $O(m)$ time using the following modification of the algorithm of [3].

Let D be the implication digraph corresponding to J. Initially we create a partial truth assignment by setting to true the literals in D corresponding to the singular rotations of I'. We then create $2n$ buckets $B_0, B_1, \ldots, B_{2n-1}$, each of which is initially empty and unmarked. Each non-singular rotation ρ is placed in the bucket corresponding to its rank – we denote this bucket by $B(\rho)$. We also include a pointer from ρ to its position in $B(\rho)$. If there is a directed path from ρ to $\bar{\rho}$ in D, we mark $B(\rho)$. Intuitively, a bucket B_R is marked if it contains a rotation ρ that cannot be eliminated – in this case $r_{\prec}(M) \geq R$ for any stable matching M in I by Lemma 6.3.

Let R be the maximum integer such that bucket B_R is non-empty. If B_R is marked then we extend the current partial truth assignment arbitrarily to a satisfying truth assignment using the algorithm of [3] and terminate. This assignment corresponds to a super-stable matching M in I such that $r_{\prec}(M) = R$. Otherwise pick a rotation ρ from B_R. Set the literal in D corresponding to ρ to be true, remove ρ from B_R and mark $B(\bar{\rho})$. For any rotation σ reachable from ρ by a directed path in D, set the literal in D corresponding to σ to be true and mark $B(\bar{\sigma})$. (Note that there cannot be a path from σ to $\bar{\sigma}$ in D, for this would imply a path from ρ to $\bar{\rho}$, as there is a path from $\bar{\sigma}$ to $\bar{\rho}$ by construction of D and by [11, Lemma 4.3.7(ii)], a contradiction.) We then repeat the steps in this paragraph.

We remark that $R \geq 1$ upon termination of the above algorithm. For, by our earlier assumption, there exists a pair $\{a_i, a_j\} \in M$ that is not a fixed pair of I. Hence by [11, Lemma 4.4.1], either (i) $(a_i, a_j) \in \rho$ or (ii) $(a_j, a_i) \in \rho$, for some rotation ρ in I'. It follows that $\bar{\rho} \in \mathcal{Z}$, where $\mathcal{Z}$ is the complete subset of rotations in I' that is closed in R_K^* corresponding M, by Theorems 3.3 and 5.3. Hence when the algorithm chooses $\bar{\rho}$, bucket $B(\rho)$ will be marked. But $rank_{\prec}(\rho) \geq 1$, since $\{a_i, a_j\}$ is a super-stable pair in I. It follows that $R \geq 1$ as required.

With a suitable choice of data structures the algorithm described above can be implemented to run in $O(m)$ time, and therefore we may summarise the discussion of this section by the following theorem.

Theorem 6.5. *Let I be a solvable instance of* SRPF*. Then we may find a minimum regret super-stable matching in I in $O(nm)$ time. If I is an instance of* SMPF *then this complexity improves to $O(m)$ time.*

7 Concluding remarks

In this paper we have presented efficient algorithms for a range of problems concerned with finding various types of stable matchings, given instances of SRPF and SMPF. These include the problems of finding all super-stable pairs, listing all super-stable matchings, and finding egalitarian and minimum regret super-stable matchings, given an instance of SMT. As mentioned in Section 1, alternative forms of stability have been defined in the context of SMT, and one could also consider the aforementioned problems with respect to weak stability. However in this setting, the following results are known, given an SMT instance I:

1. The problem of finding a minimum regret weakly stable matching in I is NP-hard [23] and not approximable within $\Omega(n)$ [12].

2. The problem of finding an egalitarian weakly stable matching in I is NP-hard [23] and not approximable within $\Omega(n)$ [12].

3. The problem of deciding whether a given (man,woman) pair (m, w) is a weakly stable pair in I is NP-complete [23].

4. Given a weakly stable matching M_1 in I, there is an $O(m)$ algorithm that finds a weakly stable matching $M_2 \neq M_1$ or reports that M_1 is the unique weakly stable matching in I [29].

It remains open as to whether the algorithm in Item 4 can be extended to give an efficient algorithm for listing all weakly stable matchings in I without repetition. However, it is known that the problem of deciding whether an instance of SMTF admits a weakly stable matching is NP-complete [29].

For the remaining stability criterion, it is open as to whether the results of this paper can be extended to the case of SRTF or SMTF under strong stability. (Recall from Section 1 that the problem of deciding whether a given SMP instance admits a strongly stable matching is NP-complete.) We note that Feder [4, p.148] conjectures that, for an instance I of SMT, the problem of deciding whether there is a strongly stable matching other than the man-optimal and woman-optimal strongly stable matchings is NP-complete. However it is known that there may be more than one man-optimal strongly stable matching in I, though all such matchings are equivalent up to indifference [22] (a similar remark holds for woman-optimal strongly stable matchings). Based on this observation, one may define a suitable equivalence relation on the set of strongly stable matchings in I, and it is then possible to check in polynomial time if a given matching belongs to a given equivalence class [22]. Hence the problem that Feder's conjecture relates to can be expressed as follows: is there a strongly stable matching M in I, such that $M \notin [M_0]$ and $M \notin [M_z]$, where $[M_0]$ and $[M_z]$ are the equivalence classes corresponding to any man-optimal and any woman-optimal strongly matching in I, respectively?

We conclude with a further open problem, which may be specified as follows: given an instance $I = \langle A, \prec, F \rangle$ of SRF, where $I' = \langle A, \prec \rangle$ is solvable, find a matching M in I such that (i) M is stable in I', and (ii) $|M \cap F|$ is minimum. Is this problem solvable in polynomial time? We remark that, if I is an instance of SMF, the problem may be solved in $O(m^{1.5})$ time by considering the weights $wt_{\prec_{a_i}}(a_j) = 1$ for each $\{a_i, a_j\} \in E$ and $wt_{\prec_{a_i}}(a_j) = 0$ for each $\{a_i, a_j\} \in E \backslash F$, and then applying Theorem 6.2, where $G = (V, E)$ is the underlying graph of I. However it also remains to show that the problem can be solved in $O(m)$ time in the SMF case.

References

[1] K. Cechlárová and T. Fleiner. On a generalization of the stable roommates problem. *ACM Transactions on Algorithms*, 1(1):143–156, 2005.

[2] V.M.F. Dias, G.D. da Fonseca, C.M.H. de Figueiredo, and J.L. Szwarcfiter. The stable marriage problem with restricted pairs. *Theoretical Computer Science*, 306(1-3):391–405, 2003.

[3] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1975.

[4] T. Feder. *Stable Networks and Product Graphs*. PhD thesis, Stanford University, 1990. Published in *Memoirs of the American Mathematical Society*, vol. 116, no. 555, 1995.

[5] T. Feder. A new fixed point approach for stable networks and stable marriages. *Journal of Computer and System Sciences*, 45:233–284, 1992.

[6] T. Feder. Network flow and 2-satisfiability. *Algorithmica*, 11(3):291–319, 1994.

[7] P. Fishburn. Preference structures and their numerical representations. *Theoretical Computer Science*, 217:359–383, 1999.

[8] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.

[9] D. Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987.

[10] D. Gusfield. The structure of the stable roommate problem – efficient representation and enumeration of all stable assignments. *SIAM Journal on Computing*, 17(4):742–769, 1988.

[11] D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, 1989.

[12] M.M. Halldórsson, R.W. Irving, K. Iwama, D.F. Manlove, S. Miyazaki, Y. Morita, and S. Scott. Approximability results for stable marriage problems with ties. *Theoretical Computer Science*, 306(1-3):431–447, 2003.

[13] R.W. Irving. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms*, 6:577–595, 1985.

[14] R.W. Irving. On the stable room-mates problem. Technical Report CSC/86/R5, University of Glasgow, Department of Computing Science, 1986.

[15] R.W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48:261–272, 1994.

[16] R.W. Irving and P. Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667, 1986.

[17] R.W. Irving and D.F. Manlove. The Stable Roommates Problem with Ties. *Journal of Algorithms*, 43:85–105, 2002.

[18] R.W. Irving, D.F. Manlove, and S. Scott. Strong stability in the Hospitals/Residents problem. In *Proceedings of STACS 2003: the 20th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2607 of *Lecture Notes in Computer Science*, pages 439–450. Springer, 2003.

[19] D.E. Knuth. *Stable Marriage and its Relation to Other Combinatorial Problems*, volume 10 of *CRM Proceedings and Lecture Notes*. American Mathematical Society, 1997. English translation of *Mariages Stables*, Les Presses de L'Université de Montréal, 1976.

[20] E. Kujansuu, T. Lindberg, and E. Mäkinen. The stable roommates problem and chess tournament pairings. *Divulgaciones Matemáticas*, 7(1):19–28, 1999.

[21] D.F. Manlove. Stable marriage with ties and unacceptable partners. Technical Report TR-1999-29, University of Glasgow, Department of Computing Science, January 1999.

[22] D.F. Manlove. The structure of stable marriage with indifference. *Discrete Applied Mathematics*, 122(1-3):167–181, 2002.

[23] D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002.

[24] C. Ng and D.S. Hirschberg. Lower bounds for the stable marriage problem and its variants. *SIAM Journal on Computing*, 19:71–77, 1990.

[25] E. Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11:285–304, 1990.

[26] A.E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92(6):991–1016, 1984.

[27] A.E. Roth, T. Sönmez, and M. Utku Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125:151–188, 2005.

[28] A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: a study in game-theoretic modeling and analysis*, volume 18 of *Econometric Society Monographs*. Cambridge University Press, 1990.

[29] S. Scott. *A study of stable marriage problems with ties*. PhD thesis, University of Glasgow, Department of Computing Science, 2005.

[30] B. Spieker. The set of super-stable marriages forms a distributive lattice. *Discrete Applied Mathematics*, 58:79–84, 1995.

[31] A. Subramanian. A new approach to stable matching problems. *SIAM Journal on Computing*, 23(4):671–700, 1994.

[32] J.J.M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *Journal of Algorithms*, 12:154–178, 1991.