Pairwise preferences in the stable marriage problem

ÁGNES CSEH, Centre for Economic and Regional Studies, Institute of Economics, Hungary and Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
ATTILA JUHOS, Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Hungary

We study the classical, two-sided stable marriage problem under pairwise preferences. In the most general setting, agents are allowed to express their preferences as comparisons of any two of their edges and they also have the right to declare a draw or even withdraw from such a comparison. This freedom is then gradually restricted as we specify six stages of orderedness in the preferences, ending with the classical case of strictly ordered lists. We study all cases occurring when combining the three known notions of stability—weak, strong and super-stability—under the assumption that each side of the bipartite market obtains one of the six degrees of orderedness. By designing three polynomial algorithms and two NP-completeness proofs we determine the complexity of all cases not yet known, and thus give an exact boundary in terms of preference structure between tractable and intractable cases.

CCS Concepts: • **Mathematics of computing → Combinatorial algorithms**;

Additional Key Words and Phrases: stable marriage, intransitivity, acyclic preferences, poset, weakly stable matching, strongly stable matching, super stable matching

ACM Reference Format:
Ágnes Cseh and Attila Juhos. 2020. Pairwise preferences in the stable marriage problem. *ACM Transactions on Economics and Computation* 1, 1 (July 2020), 29 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

In the 2016 USA Presidential Elections, polls unequivocally reported Democratic presidential nominee Bernie Sanders to be more popular than Republican candidate Donald Trump [39, 40]. However, Sanders was beaten by Hillary Clinton in their own party's primary election cycle, thus the 2016 Democratic National Convention endorsed Clinton to be the Democrat's candidate. In the Presidential Elections, Trump defeated Clinton. This recent example demonstrates well how inconsistent pairwise preferences can be.

Preferences play an essential role in the stable marriage problem and its extensions. In the classical setting [14], each man and woman expresses their preferences on the members of the opposite gender by providing a strictly ordered list. A set of marriages is stable if no pair of agents blocks it. A man and a woman form a blocking pair if they mutually prefer one another to their respective spouses.

Authors' addresses: Ágnes Cseh, Centre for Economic and Regional Studies, Institute of Economics, Tóth Kálmán utca 4. Budapest, 1097, Hungary Hasso Plattner Institute, University of Potsdam, Potsdam, Germany, cseh.agnes@krtk.mta.hu; Attila Juhos, Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Magyar tudósok körútja 2. Budapest, 1117, Hungary, juhosattila@cs.bme.hu.

XXXX-XXXX/2020/7-ART $15.00
https://doi.org/0000001.0000001

Requiring strict preference orders in the stable marriage problem is a strong assumption, which rarely suits real world scenarios [6]. The study of less restrictive preference structures has been flourishing [3, 11, 19, 23, 26, 32] for decades. As soon as one allows for ties in preference lists, the definition of a blocking edge needs to be revisited. In the literature, three intuitive definitions are used, each of which defines weakly, strongly and super stable matchings. According to weak stability, a matching is blocked by an edge uw if agents u and w both strictly prefer one another to their partners in the matching. A strongly blocking edge is preferred strictly by one end vertex, whereas it is not strictly worse than the matching edge at the other end vertex. A blocking edge is at least as good as the matching edge for both end vertices in the super stable case. Super stable matchings are strongly stable and strongly stable matchings are weakly stable by definition.

Weak stability is an intuitive notion that is most aligned with the classical blocking edge definition in the model defined by Gale and Shapley [14]. However, reaching strong stability is the goal to achieve in many applications, such as college admission programs. In most countries, students need to submit a strict ordering in the application procedure, but colleges are not able to rank all applicants strictly, hence large ties occur in their lists. According to the equal treatment policy used in Chile and Hungary for example, it may not occur that a student is rejected from a college preferred by him, even though other students with the same score are admitted [7, 35]. Other countries, such as Ireland [9], break ties with lottery, which gives way to a weakly stable solution. Super stable matchings are admittedly less relevant in applications, however, they represent worst-case scenarios if uncertain information is given about the agents' preferences. If two edges are incomparable to each other due to incomplete information derived from the agent, then it is exactly the notion of a super stable matching that guarantees stability, no matter what the agent's true preferences are.

The goal of our present work is to investigate the three cases of stability in the presence of preference structures that are more general than ties.

1.1 Related work

It is an empirical fact that cyclic and intransitive preferences often emerge in the broad topic of voting and representation, if the set of voters differs for some pairwise comparisons [2], such as in our earlier example with the polls on the Clinton–Sanders–Trump battle. Preference aggregation is another field that often yields intransitive group preferences, as the famous Condorcet-paradox [10] also demonstrates.

It might be less known that nontrivial preference structures naturally emerge in the preferences of individuals as well. The study of cyclic and intransitive preferences of a person has been inspiring scientists from a wide range of fields for decades. Blavatsky [8] demonstrated that in choice situations under risk, the overwhelming majority of individuals expresses intransitive choice and violation of standard consistency requirements. Humphrey [17] found that cyclic preferences persist even when the choice triple is repeated for the second time. Using MRI scanners, neuroscientists identified brain regions encoding 'local desirability', which led to clear, systematic and predictable intransitive choices of the participants of the experiment [25]. Cyclic and intransitive preferences occur naturally in multi-attribute comparisons [12, 34]. May [34] studied the choice on a prospective partner and found that a significant portion of the participants expressed the same cyclic preference relations if candidates lacking exactly one of the three properties intelligence, looks, and wealth were offered at pairwise comparisons. In this paper, we investigate the stable marriage problem equipped with the ubiquitous and well-studied preference structures of pairwise preferences that might be intransitive or cyclic.

Regarding the stable marriage problem, all three notions of stability have been thoroughly investigated if preferences are given in the form of a partially ordered set, a list with ties, or a strict list [14, 19, 23, 26, 32, 33]. Weakly stable matchings always exist and can be found in polynomial time [32], and a super stable matching or a proof for its non-existence can also be produced in polynomial time [19, 33]. The most sophisticated ideas are needed in the case of strong stability, which turned out to be solvable in polynomial time if both sides have tied preferences [19]. Irving [19] remarked that "Algorithms that we have described can easily be extended to the more general problem in which each person's preferences are expressed as a partial order. This merely involves interpreting the 'head' of each person's (current) poset as the set of source nodes, and the 'tail' as the set of sink nodes, in the corresponding directed acyclic graph." Together with his coauthors, he refuted this statement for strongly stable matchings and showed that exchanging ties for posets actually makes the strongly stable marriage problem NP-complete [23]. We show in this paper that the intermediate case, namely when one side has ties, while the other side has posets, is solvable in polynomial time.

Beyond posets, studies on the stable marriage problem with general preferences occur sporadically. These we include in Table 1 to give a structured overview on them. Intransitive, but acyclic preference lists were permitted by Abraham [1], who connects the stable roommates problem with the maximum size weakly stable marriage problem with intransitive, acyclic preference lists in order to derive a structural perspective. Aziz et al. [3] discussed the stable marriage problem under uncertain pairwise preferences. They also considered the case of certain, but cyclic preferences and showed that deciding whether a weakly stable matching exists is NP-complete if both sides can have cycles in their preferences. Strongly and super stable matchings were discussed by Farczadi et al. [11]. Throughout their paper they assumed that one side has strict preferences, and proved that finding a strongly or a super stable matching (or proving that none exists) can be done in polynomial time if the other side has cyclic lists, where cycles of length at least 3 are permitted to occur, but the problems become NP-complete as soon as cycles of length 2 are also allowed.

1.2 Our contribution

This paper aims to provide a coherent framework for the complexity of the stable marriage problem under various preference structures. We consider the three known notions of stability: weak, strong and super. In our analysis we distinguish six stages of entropy in the preference lists; strict lists, lists with ties, posets, acyclic pairwise preferences, asymmetric pairwise preferences and arbitrary pairwise preferences. All of these have been defined in earlier papers, along with some results on them. Here we collect and organize these known results in all three notions of stability, considering six cases of orderedness for each side of the bipartite graph. Table 1 summarizes these results. Rows and columns distinguish between preference relations considered on the two sides of the graph. The cell itself shows the complexity class of determining whether the specified problem admits a stable matching.

Each of the three tables contained unfilled cells, i.e. cases with unknown complexity so far. These are denoted by colored cells in Table 1. We fill all gaps, providing two NP-completeness proofs and three polynomial time algorithms. Interestingly, the three tables have the border between polynomial time and NP-complete cases at very different places. As a byproduct of our new proofs, we are able to answer all analogous complexity questions in the non-bipartite stable roommates problem as well (see Table 2 in Section 6).

Structure of the paper. We define the problem variants formally in Section 2. Weak, strong and super stable matchings are then discussed in Sections 3, 4 and 5, respectively. In Section 6, we focus on non-bipartite instances, and then conclude with an open problem in Section 7.

WEAK	strict	ties	poset	acyclic	asymmetric or arbitrary
strict	$\mathcal{O}(m)$ [14]	$\mathcal{O}(m)$ [19]	$\mathcal{O}(m)$ [32]	$\mathcal{O}(m)$	NPC
ties		$\mathcal{O}(m)$ [19]	$\mathcal{O}(m)$ [32]	$\mathcal{O}(m)$	NPC
poset			$\mathcal{O}(m)$ [32]	$\mathcal{O}(m)$	NPC
acyclic				$\mathcal{O}(m)$	NPC
asymmetric or arbitrary					NPC [3]

STRONG	strict	ties	poset	acyclic	asymmetric	arbitrary
strict	$\mathcal{O}(m)$ [14]	$\mathcal{O}(m)$ [19, 26]	$\mathcal{O}(m)$ [11]	$\mathcal{O}(m)$ [11]	$\mathcal{O}(m)$ [11]	NPC [11]
ties		$\mathcal{O}(nm)$ [19, 26]	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	NPC [11]
poset			NPC [23]	NPC [23]	NPC [23]	NPC [23]
acyclic				NPC [23]	NPC [23]	NPC [23]
asymmetric					NPC [23]	NPC [23]
arbitrary						NPC [23]

SUPER	strict	ties	poset	acyclic	asymm.	arbitrary
strict	$\mathcal{O}(m)$ [14]	$\mathcal{O}(m)$ [19]	$\mathcal{O}(m)$ [19, 33]	$\mathcal{O}(m)$ [11]	$\mathcal{O}(m)$ [11]	NPC [11]
ties		$\mathcal{O}(m)$ [19]	$\mathcal{O}(m)$ [19, 33]	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	NPC [11]
poset			$\mathcal{O}(m)$ [19, 33]	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	NPC [11]
acyclic				NPC	NPC	NPC [11]
asymmetric					NPC	NPC [11]
arbitrary						NPC [11]

Table 1. The complexity tables for weak, strong and super-stability. For the sake of conciseness, NP-completeness is shortened to NPC. The number of agents in the input is denoted by n, while m stands for the number of acceptable pairs. Each blank cell under the diagonal of the table represents the same case—and thus has the same complexity—as the cell mirrored to the diagonal. All of our positive results also deliver a stable matching or a proof for its nonexistence.

2 PRELIMINARIES

In the stable marriage problem we are given a not necessarily complete bipartite graph $G = (U \cup W, E)$, where vertices in U represent men, vertices in W represent women, and edges mark the acceptable relationships between them. Each person $v \in U \cup W$ specifies a set $\mathcal{R}_v$ of pairwise comparisons on the vertices adjacent to them. These comparisons as ordered pairs define four possible relations between two vertices a and b in the neighborhood of v.

(1) a is preferred to b, while b is not preferred to a by v: $a \prec_v b$;
(2) a is not preferred to b, while b is preferred to a by v: $a \succ_v b$;
(3) a is not preferred to b, neither is b preferred to a by v: $a||_v b$;
(4) a is preferred to b, so is b preferred to a by v: $a \sim_v b$.

In words, the first two relations express that an agent v *prefers* one agent *strictly* to the other. The third option is interpreted as *incomparability*, or a not yet known relation between the two agents. The last relation indicates that v is certain that the two options are *equally good*. For example, if v is a sports sponsor considering to offer a contract to exactly one of players a and b, then v's preferences are described by these four relations in the following scenarios: a has beaten b, b has beaten a, a and b have not yet played against each other, and finally, a and b have played a draw. A more precise mathematical interpretation of the four relations is based on the set $\mathcal{R}_v$ of pairwise comparisons: the first one is equivalent to $(a, b) \in \mathcal{R}_v$, but $(b, a) \notin \mathcal{R}_v$, the second one is exactly the other way round: $(a, b) \notin \mathcal{R}_v$, but $(b, a) \in \mathcal{R}_v$, the third one indicates that $(a, b) \notin \mathcal{R}_v$, and $(b, a) \notin \mathcal{R}_v$, and finally, the last one is $(a, b) \in \mathcal{R}_v$, and $(b, a) \in \mathcal{R}_v$.

We say that edge va *is strictly preferred to* edge vb if $a \prec_v b$. If $a \prec_v b$ or $a||_v b$, then b is *not preferred to a*. This happens if and only if $(b, a) \notin \mathcal{R}_v$. For our previous example with players a and b, this relation delivers the information that either a has beaten b or they have not yet played. With this amount of somewhat uncertain information, the sports sponsor has no reason to choose b, but choosing a also involves risk, because it might still be the case that the two players have not played against each other yet, and b could beat a. For two out of the three notions of stability, we will define blocking based on this risk. Another choice would be to replace $a||_v b$ by $a \sim_v b$ in the definition above, which then would be equivalent to $(a, b) \in \mathcal{R}_v$. While it would lead to an equally correct model, we chose incomparability over being equally good consciously. Some early papers [19, 20] do not distinguish between two agents being incomparable and equally good, while some others in the more recent literature [3, 11] motivate strong and super-stability with uncertain information. Our definition fits the more recent framework.

The partner of vertex v in matching M is denoted by $M(v)$. The neighborhood of v in graph G is denoted by $\mathcal{N}_G(v)$ and it consists of all vertices that are adjacent to v in G. Analogously, $\mathcal{N}_G(X)$ denotes the neighborhood of vertex set $X \subseteq V$ and it consists of all vertices that are adjacent to at least one vertex $x \in X$ in G. To ease notation, we introduce the empty set as a possible partner to each vertex, symbolizing the vertex remaining unmatched in a matching M ($M(v) = \emptyset$). As usual, being matched to any acceptable vertex is preferred to not being matched at all: $a \prec_v \emptyset$ for every $a \in \mathcal{N}(v)$. Edges to unacceptable partners do not exist, thus these are not in a pairwise relation to each other or to edges incident to v.

We differentiate six degrees of preference orderedness in our study.

(1) The strictest, classical two-sided model [14] requires each vertex to rank all of its neighbors in a *strict* order of preference. For each vertex, this translates to a transitive, antisymmetric and complete set of pairwise relations $(a, b) \in \mathcal{R}_v$ on all adjacent vertices of v.
(2) This model has been relaxed very early to lists admitting *ties* [19]. The pairwise preferences of vertex v form a preference list with ties if the neighbors of v can be clustered into some sets $N_1, N_2, \ldots, N_k$ so that vertices in the same set are incomparable, while for any two vertices in different sets, the vertex in the set with the lower index is strictly preferred to the other one.
(3) Following the traditions [13, 20, 23, 32], the third degree of orderedness we define is when preferences are expressed as *partially ordered sets (posets)*. Any set of antisymmetric and transitive pairwise relations $(a, b) \in \mathcal{R}_v$ by definition forms a poset.
(4) By dropping transitivity of $(a, b) \in \mathcal{R}_v$, but still keeping the structure cycle-free, we arrive to *acyclic* preferences [1]. This category allows for example $a||_v c$, if $a \prec_v b \prec_v c$, but it excludes $a \sim_v c$ and $a \succ_v c$.
(5) *Asymmetric* preferences [11] may contain cycles of length at least 3. This is equivalent to dropping acyclicity from the previous cluster, but still prohibiting the indifference relation $a \sim_v b$, which is essentially a 2-cycle in the form $(a, b) \in \mathcal{R}_v$, $(b, a) \in \mathcal{R}_v$.
(6) Finally, an *arbitrary* set of pairwise preferences can also be allowed [3, 11].

A matching is *stable* if it admits no blocking edge. For strict preferences, a blocking edge was defined in the seminal paper of Gale and Shapley [14]: an edge $uv \notin M$ blocks matching M if both u and v prefer each other to their partner in M or they are unmatched. Already when extending this notion to preference lists with ties, one needs to specify how to deal with incomparability. Irving [19] defined three notions of stability. We extend them to pairwise preferences in the coming three sections. We omit the adjectives weakly, strongly, and super wherever there is no ambiguity about the type of stability in question.

We define the NP-complete [4] satisfiability problem (2,2)-E3-SAT here, because it will be used in the proofs of Theorems 3.4 and 5.5 later. Its input is a Boolean formula B in conjunctive normal form, in which each clause comprises exactly 3 literals and each variable appears exactly twice in positive and exactly twice in negated form. The decision question is whether there exists a truth assignment that satisfies B.

3 WEAK STABILITY

In weak stability, an edge outside of M blocks M if it is *strictly preferred* to the matching edge by *both* of its end vertices. From this definition follows that $w \sim_u w'$ and $w||_u w'$ are exchangeable in weak stability, because blocking occurs only if the non-matching edge is strictly preferred to the matching edges at both end vertices. Therefore, an instance with arbitrary pairwise preferences can be assumed to be asymmetric.

Definition 3.1 (blocking edge for weak stability). Edge uw blocks M, if

(1) $uw \notin M$;
(2) $w \prec_u M(u)$;
(3) $u \prec_w M(w)$.

For weak stability, preference structures up to posets have been investigated, see Table 1. A stable solution is guaranteed to exist in these cases [19, 32]. Here we extend this result to acyclic lists, and complement it with a hardness proof for all cases where asymmetric lists appear, even if they do so on one side only.

Theorem 3.2. *Any instance of the stable marriage problem with acyclic pairwise preferences for all vertices admits a weakly stable matching, and there is a polynomial time algorithm to determine such a matching.*

Proof. We utilize a widely used argument [19] to show this. A *linear extension* of an acyclic set $\mathcal{R}$ of pairwise relations on a finite set $V = (v_1, v_2, \ldots, v_n)$ of n elements is a total ordering $\pi = \pi_1, \pi_2, \ldots, \pi_n$ of V such that $\pi_i < \pi_j$ whenever $v_i \prec v_j$ appears in $\mathcal{R}$. For acyclic relations $\mathcal{R}_v$, a linear extension $\mathcal{R}'_v$ of $\mathcal{R}_v$ exists. The extended instance with linear preferences is guaranteed to admit a stable matching [14]. Compared to $\mathcal{R}_v$, relations in $\mathcal{R}'_v$ impose more constraints on stability, therefore, they can only restrict the original set of weakly stable solutions. If both sides have acyclic lists, a stable matching is thus guaranteed to exist and a single run of the Gale-Shapley algorithm on the extended instance delivers one.

The time complexity of the Gale-Shapley algorithm in the instance with extended preferences is $O(m)$. Constructing a linear extension to a set of acyclic relations at each vertex is part of a pre-processing phase, as in [32]. The complexity of this phase heavily depends on the format of $\mathcal{R}'_v$ in the input, and $O\left(n^3\right)$ serves as an upper bound for the whole graph [24]. □

Stable matchings are not guaranteed to exist as soon as a cycle appears in the preferences, as Example 3.3 demonstrates. Theorem 3.4 shows that the decision problem is in fact hard from that point on.

Example 3.3. No stable matching can be found in the following instance with strict lists on one side and asymmetric lists on the other side. There are three men u_1, u_2, u_3 adjacent to one woman w. The woman's pairwise preferences are cyclic: $u_1 \prec u_2, u_2 \prec u_3, u_3 \prec u_1$. Any stable matching M must consist of a single edge. Since the men's preferences are identical, we can assume that $M = \{u_1 w\}$ without loss of generality. Then $u_3 w$ blocks M.

Theorem 3.4. *If one side has strict lists, while the other side has asymmetric pairwise preferences, then determining whether a weakly stable matching exists is* NP*-complete, even if each agent finds at most four other agents acceptable.*

Proof. The NP-complete problem we reduce to our problem is (2,2)-e3-sat [4], defined in Section 2. When constructing graph G to a given Boolean formula B, we keep track of the three literals in each clause and the two positive and two negated appearances of each variable. Each appearance is represented by an interconnecting edge, running between the corresponding variable and clause gadget. The graphs underlying our gadgets resemble gadgets in earlier hardness proofs [5], but the preferences are designed specifically for our problem. Figure 1 illustrates our construction, in particular, the preference relations in it.

$$\begin{array}{rl} t: & \text{strict list: } x \prec \bar{x} \\ f: & \text{strict list: } \bar{x} \prec x \\ \hline x: & f \prec t, t \prec u, u \prec f \\ \bar{x}: & t \prec f, t \prec u, u \prec f \end{array}$$

$$\begin{array}{rl} u_1: & \text{strict list: } w_3 \prec w_2 \prec x/\bar{x} \prec w_1 \\ u_2: & \text{strict list: } w_3 \prec w_2 \prec x/\bar{x} \prec w_1 \\ u_3: & \text{strict list: } w_3 \prec w_2 \prec x/\bar{x} \prec w_1 \\ \hline w_1: & \emptyset \\ w_2: & \emptyset \\ w_3: & \emptyset \end{array}$$

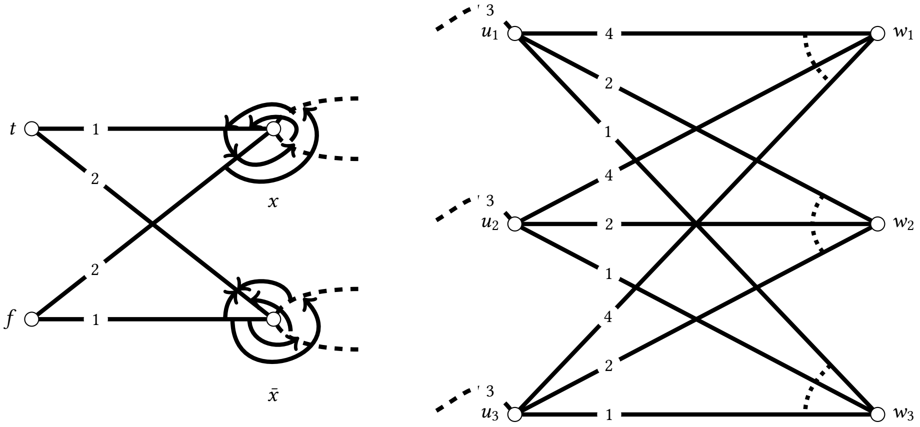

Fig. 1. A variable gadget to the left and a clause gadget to the right. Strict lists are to be found at t, f, and u-vertices, while the rest of the vertices have asymmetric relations. The elements u and $x/\bar{x}$ in the lists stand for the corresponding u_i and x or $\bar{x}$ vertices, respectively. Interconnecting edges are dashed. The exact clause is expressed by the connections established by these interconnecting edges. The arrows point to the strictly preferred edge, while dotted lines denote incomparability.

To each variable x in B, we create 4 vertices: $t, \bar{x}, f$, and x. To each clause in B, we create 6 vertices: u_1, u_2, u_3, w_1, w_2, and w_3.

In each *variable gadget*, x symbolizes the two positive occurrences of the variable, while $\bar{x}$ stands for the two negated occurrences. The edge set of each variable gadget comprises a 4-cycle $t, \bar{x}, f, x$ and four interconnecting edges, two of which are incident to x, and the remaining two are adjacent to $\bar{x}$. These four interconnecting edges are responsible for the communication between clause and variable gadgets and they connect the vertices x and $\bar{x}$ to u-vertices in clause gadgets.

The *clause gadget* consists of a complete bipartite graph on six vertices u_1, u_2, u_3, w_1, w_2, and w_3, where each vertex on the u-side is equipped with exactly one interconnecting edge. This side represents the three literals in the clause. Each interconnecting edge runs from the u-vertex to vertex x or $\bar{x}$ in the variable gadget of the literal. If the literal is positive, then the other end vertex of this edge is x in the corresponding variable gadget, otherwise it is $\bar{x}$.

The two sides of the bipartite graph will be formed by vertices t, f, u_1, u_2, u_3 from all gadgets on one side, and $x, \bar{x}, w_1, w_2, w_3$ from all gadgets on the other side. Edges inside variable gadgets connect vertices t, f with vertices $x, \bar{x}$, edges inside clause gadgets connect vertices u_1, u_2, u_3 with vertices w_1, w_2, w_3, and finally, interconnecting edges connect vertices u_1, u_2, u_3 with vertices $x, \bar{x}$. This guarantees the bipartite property of the constructed graph.

Claim 1. *If there is a weakly stable matching M in G, then there is a truth assignment that satisfies B.*

Proof. We first show that M does not contain any interconnecting edge by proving that all stable matchings include either $\{tx, f\bar{x}\}$ or $\{fx, t\bar{x}\}$, for each variable gadget. Two cases may occur.

- If $tx \in M$, then $f\bar{x} \in M$, otherwise fx blocks M.
- If $tx \notin M$, then $fx \in M$, otherwise tx blocks M. Furthermore, $t\bar{x} \in M$, otherwise $t\bar{x}$ blocks M.

We know that all weakly stable matchings are inclusionwise maximal by definition. From this fact and our arguments above follows that M restricted to an arbitrary clause gadget must be a perfect matching. The preferences in the clause gadgets are set so that out of the three interconnecting edges running to a clause gadget, exactly one is strictly preferred to M at the clause gadget, namely the edge incident to vertex u_i matched to w_1. We know that M is stable, therefore, this interconnecting edge xu_i or $\bar{x}u_i$ may not be strictly preferred to the matching edge at its other end vertex x or $\bar{x}$. This is only possible if the variable represented by the vertex u_i is set to

- true if the literal was positive in the clause to which u_i belongs, and the end vertex above was x;
- false if the literal was in negated form in the clause to which u_i belongs, and the end vertex above was $\bar{x}$.

Thus, we have found a satisfied literal in each clause. □

Claim 2. *If there is a truth assignment that satisfies B, then there is a stable matching M in G.*

Proof. In each variable gadget that belongs to a true variable, $\{tx, f\bar{x}\}$ is chosen, whereas all gadgets corresponding to a false variable contribute the edges $\{fx, t\bar{x}\}$. In each clause, there is at least one true literal. We match the vertex representing the appearance of this literal to w_1 and match w_2 and w_3 arbitrarily.

No edge inside of a gadget blocks M, because it is a perfect matching inside each gadget and the preferences either form a cycle (variable gadget), or one side is indifferent (clause gadget). An interconnecting edge is strictly preferred to M at the clause gadget if and only if it corresponds to the chosen literal satisfying the clause. Our rules set exactly this literal to be satisfied in the variable gadget, i.e. this literal is matched to t, which is strictly preferred to the corresponding interconnecting edge. □

With this, we have completed our hardness proof. □

4 STRONG STABILITY

In strong stability, an edge outside of M blocks M if it is *strictly preferred* to the matching edge by *one* of its end vertices, while the other end vertex *does not prefer* its matching edge to it.

Definition 4.1 (blocking edge for strong stability). Edge uw blocks M, if

(1) $uw \notin M$;		(1) $uw \notin M$;
(2) $w \prec_u M(u)$ or $w \|\|_u M(u)$;	or	(2) $w \prec_u M(u)$;
(3) $u \prec_w M(w)$,		(3) $u \prec_w M(w)$ or $u \|\|_w M(w)$.

The largest set of relevant publications has appeared on strong stability, yet gaps were present in the complexity table, see Table 1. In this section we present a polynomial algorithm that is valid in all cases not yet resolved. We assume men to have preference lists with ties, and women to have asymmetric relations. Our algorithm returns a strongly stable matching or a proof for its nonexistence. It can be seen as an extended version of Irving's algorithm for strongly stable matchings in instances with ties on both sides [19]. Our contribution is a sophisticated rejection routine, which is necessary here, because of the lack of transitivity of preferences on the women's side. The algorithm in [11] solves the problem for strict lists on the men's side, and it is much simpler than ours. It was designed for super stable matchings, but strong and super stability do not differ if one side has strict lists. As soon as we allow ties instead of strict lists on the men's side, the two sets of matchings differ, and thus it will not be sufficient to apply an algorithm designed for super-stability.

4.1 Our algorithm

Intuitively, our algorithm alternates between two phases, both of which iteratively eliminate edges that cannot occur in a strongly stable matching. In the first phase, Gale-Shapley proposals and rejections happen, while the second phase focuses on finding a vertex set violating the Hall condition in a specified subgraph. Finally, if no edge can be eliminated any more, then we show that every maximum matching is either stable or it is a proof for the non-existence of stable matchings. Algorithm 1 and its subroutine Algorithm 2 below provide a pseudocode.

The second phase of the algorithm relies on the notion of the *critical set* in a bipartite graph, also utilized in [19], which we sketch here. For an exhaustive description we refer the reader to [30]. The well-known Hall-condition [16] states that there is a matching covering the entire vertex set U if and only if for each $X \subseteq U$, $|\mathcal{N}(X)| \geq |X|$. Intuitively, the reason for no matching being able to cover all the vertices in U is that a subset X of them has too few neighbors in W to cover X. The difference $\delta(X) = |X| - |\mathcal{N}(X)|$ is called the *deficiency of* X. It is straightforward that for any $X \subseteq U$, at least $\delta(X)$ vertices in X cannot be covered by any matching in G, if $\delta(X) > 0$. Let $\delta(G)$ denote the maximum deficiency over all subsets of U. Since $\delta(\emptyset) = 0$, we know that $\delta(G) \geq 0$. Moreover, it can be shown that the size of a maximum matching is $\nu(G) = |U| - \delta(G)$. If we let Z_1, Z_2 be two arbitrary subsets of U realizing the maximum deficiency, then $Z_1 \cap Z_2$ has maximum deficiency as well (see [30, Lemmas 1.3.2 and 1.3.3]). Therefore, the intersection of all maximum-deficiency subsets of U is the unique set with maximum deficiency with the following properties: it has the smallest cardinality and it is contained in all other subsets with maximum deficiency. This set is called the *critical set* of G. Last but not least, it is computationally easy to determine the critical set, since for any maximum matching M in G, the critical set consists of vertices in U not covered by M and vertices in U reachable from the uncovered ones via an alternating path.

We are now ready to state our theorem on our algorithm.

THEOREM 4.2. *If one side has tied preferences, while the other side has asymmetric pairwise preferences, then deciding whether the instance admits a strongly stable matching (and outputting one, if so) can be done in* $O\left(m^2\right)$ *time.*

Initialization. We assume men to have tied lists, while women provide asymmetric pairwise preferences. For the clarity of our proofs, we add a dummy partner w_u to the bottom of the list of each man u, where w_u is not acceptable to any other man (line 1). We call the modified instance $\mathcal{I}'$. This standard technical modification is to ensure that all men are matched in all stable matchings. At start, all edges are *inactive* (line 2). The possible states of an edge and the transitions between them are illustrated in Figure 2.

Algorithm 1 Strongly stable matching with ties and asymmetric relations

```
Input: $\mathcal{I} = (U, W, E, \mathcal{R}_U, \mathcal{R}_W)$; $\mathcal{R}_U$: lists with ties, $\mathcal{R}_W$: asymmetric.
 INITIALIZATION
 1: for each $u \in U$ add an extra woman $w_u$ at the end of his list; $w_u$ is only acceptable for $u$
 2: set all edges to be inactive
 PHASE 1
 3: while there exists a man with no active edge do
 4:     propose along all edges of each $u$ with no active edge in the next tie on his list
 5:     for each new proposal edge $uw$ do
 6:         reject all edges $u'w$ such that $u \prec_w u'$
 7:     end for
 8:     call STRONG_REJECT
 9: end while
 PHASE 2
10: let $G_A$ be the graph of active edges with $V(G_A) = U \cup W$
11: let $U' \subseteq U$ be the critical set of men with respect to $G_A$
12: if $U' \neq \emptyset$ then
13:     all active edges of each $u \in U'$ are rejected
14:     call STRONG_REJECT
15:     goto PHASE 1
16: end if
 OUTPUT
17: let M be a maximum matching in $G_A$
18: if $M$ covers all women who have ever had an active edge then
19:     STOP, OUTPUT $M \cap E$ and “There is a strongly stable matching.”
20: else
21:     STOP, OUTPUT “There is no strongly stable matching.”
22: end if
```

Algorithm 2 STRONG_REJECT

```
23: let $R$ be the set of men with no active edges
24: while $R$ has an element $u$ do
25:     reject all $u'w$ such that $w$ is in the proposal tie of $u$ and $u'||_w u$
26:     if $u'w$ was active and $u'$ has run out of active edges, then let $R := R \cup \{u'\}$
27:     let $R := R \setminus \{u\}$
28: end while
```

First phase. The first phase of our algorithm (lines 3-9) imitates the classical Gale-Shapley deferred acceptance procedure. In the first round each unmatched man simultaneously proposes to all women in his top tie (line 4). The so far inactive edges that now carry a proposal are called *active proposal edges*, or just *active edges*. Active edges stay active as long as they are accepted by the woman they run to, and they become *rejected proposal edges* as soon as they are rejected by her. The tie that a man has just proposed along is called the man's *proposal tie*. If all edges in the proposal tie are rejected (or more precisely, they become rejected proposal edges), then the man steps down on his list and proposes along all edges in the tie following the rejected tie in the man's preference order (lines 3 and 4).

Proposals cause two types of rejections in the graph (lines 5-6 and 8), based on the rules below. Notice that these rules are more sophisticated than in the Gale-Shapley or Irving algorithms [14, 19]. The most striking difference may be that rejected edges are not deleted from the graph, since they can very well carry a proposal later, which proposal then affects the state of other edges by triggering rejections. To be fully accurate, inactive edges that are rejected become *rejected inactive edges* (see Figure 2). Upon carrying a proposal later, they are converted to a *rejected proposal edge*. This latter is the same state an edge ends up in if it is first proposed along and then rejected.

Edges that carry a proposal in the current round, but have not carried a proposal in earlier rounds, i.e. edges that moved along the solid arrows in Figure 2 in this round, are called *new proposal edges* (lines 4-5). We remind the reader that these edges might or might not be active, depending on whether they have been rejected earlier.

- Rejections in line 6: For each new proposal edge uw, w rejects all edges to which uw is strictly preferred (lines 5-7). Note again that uw might have been rejected earlier than being proposed along, in which case uw is a proposal edge without being active.
- Rejections in line 8: A precise rejection routine is described in the procedure STRONG_REJECT in our pseudocode (Algorithm 2). At first, only men who have run out of active edges are considered (line 23), but the same mechanism is repeated for any man that runs short of active proposal edges later (line 26). For any man u in R, all edges that are incident to any neighbor w of u in his—now fully rejected—proposal tie and incomparable to uw at w are rejected. Again, if man u' has lost his last active edge during these operations, then u' is added back to the set R of men to be investigated in later rounds (line 26).

A: Nezd meg, mert eleg rosszul hangzik ez a masodik rejection fajta.

As mentioned earlier, men without any active edge proceed to propose along the next tie in their list. These operations are executed until there is no more edge to propose along or to reject, which marks the end of the first phase. It is straightforward that this process will eventually halt and the execution will undoubtedly proceed to the second phase, since each dummy woman finds her sole adjacent man acceptable, who thus cannot be rejected by this woman.

Second phase. In the second phase, the set of active edges induce the graph G_A, on which we examine the critical set $U' \subseteq U$ (lines 10-11). If U' is not empty, then all active edges of each $u \in U'$ are rejected (line 13). The mass rejections might trigger many more others, which are handled by calling Algorithm 2 as a subroutine (line 14). What is certain, at least one man is prompted to propose along his next tie, returning our algorithm to the first phase (line 15). Note that an empty critical set leads to producing the output, which is described just below.

Output. In the final set of active edges, an arbitrary maximum matching M is calculated (line 17). If M covers all women who have ever had an active edge, then we send it to the output (lines 18-19), otherwise we report that no stable matching exists (lines 20-21).

Before proving the correctness of the presented algorithm, we outline an illustrative example and the corresponding execution in Example 4.3.

Example 4.3. The instance under examination (depicted in Figure 3) consists of six men and an equal number of women. As expected, men's preferences make up lists with ties, while the preference structures associated to women form asymmetric relations. Throughout the entire figure, edges connected by dotted arcs represent incomparability, e.g. w_2 declares incomparability between m_1 and m_4. On the other hand, arrows mark strict preferences. For instance, w_1 strictly prefers m_2 to m_1.

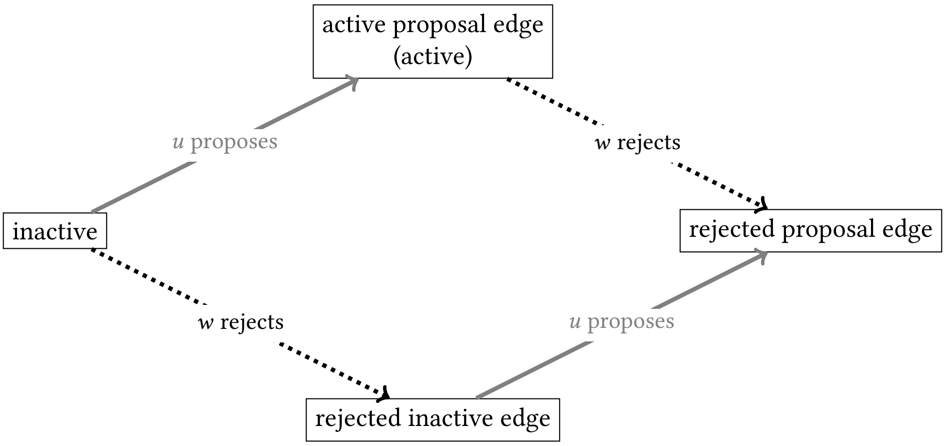

Fig. 2. The possible states of an edge uw in Algorithm 1. The solid gray edges between the states symbolize proposals coming from men, while the dotted black edges mark the rejections initiated by women. Edges that have neither been proposed along nor rejected are *inactive*. If a proposal is called first, then the edge moves along the upper path in the figure and becomes an *active proposal edge*. It can stay in this state until termination, or become a *rejected proposal edge* upon rejection. Otherwise, if the edge is rejected before a proposal is sent along it, the lower path is taken, and the edge becomes a *rejected inactive edge*, which might still carry a proposal later and become *rejected proposal edge*, but it will not necessarily be the case for all such edges. An edge can be rejected several times, but proposed along at most once.

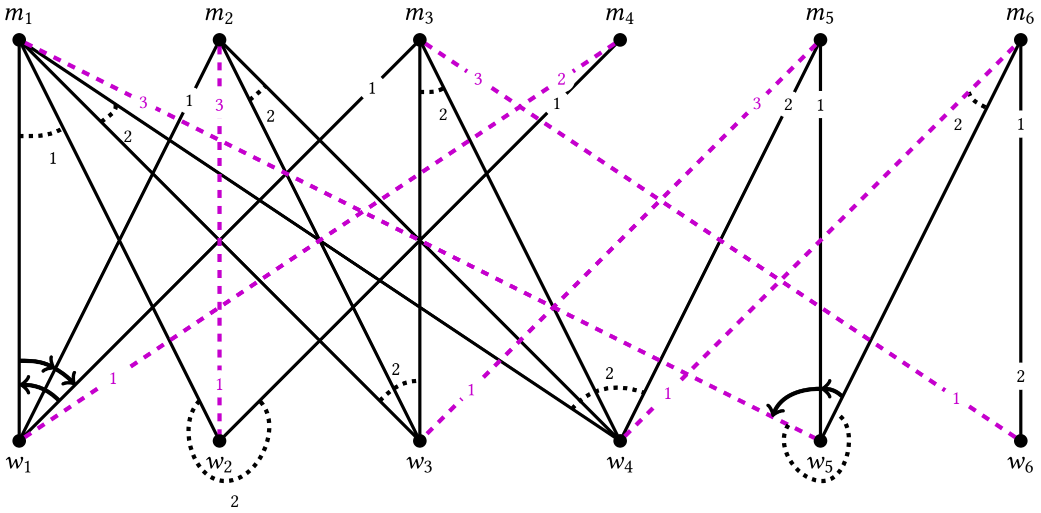

Fig. 3. The input for Example 4.3. Colored dashed edges mark the final matching.

In case of men, the order of ties is clearly marked by numbers. For example, m_1 likes women w_1 and w_2 the most, ranks w_3 and w_4 second, and considers woman w_5 as his least preferred choice. The preferences of women w_2, w_3, w_4 and w_6 can be represented by ties, hence the same interpretations of dotted lines and numbers apply as for the men. Woman w_1 possesses an asymmetric preference structure: m_4 is strictly preferred to each man acceptable to w_1, while m_1, m_2, and m_3 form a cycle.

Last but not least, woman w_5 has an intransitive preference structure as well: m_1 is strictly preferred to m_5, who is strictly preferred to m_6, yet m_1 and m_6 are considered incomparable.

Now we turn to the execution of our algorithm. To keep the figure as clear as possible, we forgo introducing dummy women. In fact, none of these women will be proposed to during the execution on this instance. Also, to keep the explanatory text short, we execute the Phase 1 proposal steps of several agents in batch.

Phase 1, proposals in line 4: Since no man has an active edge at the beginning, each man proposes to all women in his first tie. More precisely, m_1 proposes to w_1 and w_2, m_2 proposes to w_1, m_3 to w_1, m_4 to w_2, m_5 to w_5, and m_6 to w_6. These edges become active edges (see Figure 2).

Phase 1, rejections in line 6: Women check their preferences and reject all edges to which any of the proposal edges is strictly preferred to. In this manner w_1 rejects m_3, since she strictly prefers m_1 to m_3 (i.e. $m_1 \prec_{w_1} m_3$), rejects m_1, since $m_2 \prec_{w_1} m_1$ and rejects m_2 as well, because $m_3 \prec_{w_1} m_2$. These edges, having already been proposed along and subsequently rejected, become rejected proposal edges. For similar reasons, w_5 rejects m_6 in advance, thus edge m_6w_5 becomes a rejected inactive edge.

Phase 1, rejections in line 8 (strong rejections): Two men are left without an active edge: m_2 and m_3. Their proposal tie is identical and it consists of w_1 only. Since no man is incomparable to either m_2 or m_3 according to the preferences of w_1, no rejection occurs.

Phase 1, proposals in line 4: Men m_2 and m_3 are left without active edges, therefore they carry on proposing to women in their next tie, who are w_3 and w_4 for both men.

Phase 1, no rejections and strong rejections: These four new proposal edges are not preferred strictly to any other edge, thus no rejection happens. Since no man is left active-edge less, the STRONG_REJECTION procedure will return immediately.

Phase 2, searching for a critical set in line 11: The execution switches to the second phase, in which we search for a maximum matching in the graph G_A consisting of the active edges $\{m_1w_2, m_2w_3, m_2w_4, m_3w_3, m_3w_4, m_4w_2, m_5w_5, m_6w_6\}$. The neighborhood of $\{m_1, m_4\}$ is $\{w_2\}$, hence it is impossible to match both men with the one proposed women only. The man left out from this imperfect matching would form a blocking edge with w_2. This set $\{m_1, m_4\}$ is the unique set of men with positive deficiency, thus it is the critical set. All active edges of these men are rejected (lines 10-13), which are m_1w_2 and m_4w_2.

Phase 2, rejections in line 14 (strong rejections): Men m_1 and m_4 have no active edge. The proposal tie of m_1 consists of w_1 and w_2, while the proposal tie of m_4 consists of w_2 only. Since no man is incomparable to m_1 according to the preferences of w_1, and m_1 and m_4 are only incomparable to each other at w_2, only m_1w_2 and m_4w_2 are rejected (again). We switch to Phase 1 (line 15).

Phase 1, proposals in line 4: Man m_1 proposes to w_3 and w_4, because they build the next tie on his list. Man m_4 proposes to w_1. The new proposal edges in this step are $\{m_1w_3, m_1w_4, m_4w_1\}$.

Phase 1, rejections in line 6: Since w_1 strictly prefers her new proposal edge m_4w_1 to all other edges, she rejects m_1, m_2, and m_3. All of these rejections are second-time rejections. Neither w_3, nor w_4 prefers m_1 strictly to any other man, thus they do not reject any edge in this step.

Phase 1, rejections in line 8 (strong rejections): Once again, each man has at least one active edge, hence the STRONG_REJECT procedure returns and the execution switches to Phase 2.

Phase 2, searching for a critical set in line 11: The graph G_A is built by the active edges $\{m_1w_3, m_1w_4, m_2w_3, m_2w_4, m_3w_3, m_3w_4, m_4w_1, m_5w_5, m_6w_6\}$. The critical set in this graph is $\{m_1, m_2, m_3\}$, with neighborhood $\{w_3, w_4\}$. The edges $\{m_1w_3, m_1w_4, m_2w_3, m_2w_4, m_3w_3, m_3w_4\}$ are all rejected.

Phase 2, rejections in line 14 (strong rejections): Men m_1, m_2 and m_3 have no active edge. The only new rejection triggered by the recently rejected proposal ties of these three men happens along m_5w_4, because it is incomparable to all three rejected edges at w_4.

Phase 1, proposals in line 4: The three men with no active edge propose along edges m_1w_5, m_2w_2, and m_3w_6.

Phase 1, rejections in line 6: Woman w_5 rejects m_5, because she strictly prefers her new proposal edge m_1w_5 to m_5w_5. For an analogous reason, w_6 rejects m_6. Edges m_1w_2 and m_4w_2 are rejected again, because the new proposal edge m_2w_2 is strictly preferred to them by w_2.

Phase 1, rejections in line 8 (strong rejections): Men m_5 and m_6 have no active edge. Their proposal tie consists of one element each (w_5 and w_6, respectively), and this element appears in a strict preference order on the women's side. No rejection happens.

Phase 1, proposals in line 4: Man m_5 proposes to w_4 along a rejected inactive edge. This is a new proposal edge. Man m_6 proposes to w_4 and w_5.

Phase 1, rejections in line 6: Since w_4 received a proposal from her first-choice man, she rejects her other 4 (already rejected) edges. No rejection is triggered by m_5w_4 or m_6w_5.

Phase 1, rejections in line 8 (strong rejections): In the final round of Phase 1, only m_5 is left without an active edge. His proposal tie is w_4, thus edges m_1w_4, m_2w_4, and m_3w_4 are rejected again.

Phase 1, proposals in line 4: Finally, m_5 proposes to w_3.

Phase 1, rejections in line 6: As a consequence, w_3 rejects m_1, m_2, and m_3 (again).

Phase 1, rejections in line 8 (strong rejections): Every man possesses at least one active edge, hence the execution proceeds to the second phase.

Phase 2, searching for a critical set in line 11: Phase 2 starts with the set of active edges $\{m_1w_5, m_2w_2, m_3w_6, m_4w_1, m_5w_3, m_6w_4\}$, because each man has at least one active edge. These edges form a perfect matching, thus $U = \emptyset$. After all, the execution halts and marks the colored matching as stable, since it covers all women who have ever been proposed to.

4.2 Correctness

We prove the correctness of the proposed algorithm (Algorithm 1 and its subprocedure, Algorithm 2) via a number of claims, building up the proof as follows. The first three claims provide the technical footing for the last two claims. Claim 3 is a rather technical observation about the correctness of the input initialization. An edge appearing in some stable matching is called a *stable edge*. Claim 4 shows that no stable edge is ever rejected. Claim 5 proves that all stable matchings must cover all women who have ever received an offer. Then, Claim 6 proves that if the algorithm outputs a matching, then it must be stable, and Claim 7 along with its Corollary 4.4 conclude the opposite direction: if stable matchings exist, then one is outputted by our algorithm

Recall that in line 1, each man in the original instance $\mathcal{I}$ is supplemented by an extra woman. The instance formed is denoted by $\mathcal{I}'$.

Claim 3. *A matching in $\mathcal{I}'$ is stable if and only if it covers all men in $\mathcal{I}'$, and its restriction to $\mathcal{I}$ is stable.*

Proof. If a matching in $\mathcal{I}'$ leaves a man u unmatched, then uw_u blocks the matching. Thus all stable matchings in $\mathcal{I}'$ cover all men. Furthermore, the restriction to $\mathcal{I}$ of a stable matching in $\mathcal{I}'$ cannot be blocked by any edge in $\mathcal{I}$, because this blocking edge also exists in $\mathcal{I}'$.

A stable matching in $\mathcal{I}$, supplemented by the dummy edges for all unmatched men cannot be blocked by any edge in $\mathcal{I}'$, because dummy edges are last-choice edges and regular edges block in both instances simultaneously. □

Claim 4. *No stable edge in $\mathcal{I}'$ is ever rejected in the algorithm.*

Proof. Let us suppose that uw is the first rejected stable edge in $\mathcal{I}'$ and the corresponding stable matching is M. There are three rejection calls, in lines 6, 13 and 25 in Algorithm 2 (called by lines 8 and 14 in Algorithm 1). In all cases we derive a contradiction. Our arguments are illustrated in Figure 4.

- Line 6: uw was rejected because w received a proposal from a man u' such that $u' \prec_w u$. Since M is stable, u' must have a partner w' in M such that $w' \prec_{u'} w$. We also know that u' has reached w with his proposal tie, thus, due to the monotonicity of proposals, $u'w' \in M$ must have been rejected before uw was rejected. This contradicts our assumption that uw was the first rejected stable edge.
- Line 25: rejection was caused by a man u' such that $u' ||_w u$.
  We know that the whole proposal tie of u' was rejected. Since M is stable, u' must have a partner w' in M. Since $u'w'$ is a stable edge, it cannot have been rejected previously. Consequently, $w \prec_{u'} w'$. Thus, $u'w$ blocks M, which contradicts its stability.
- Line 13: uw was rejected as an active edge incident to the critical set U' in G_A.
  Let $W' = \mathcal{N}_{G_A}(U')$, $U'' = \{u \in U' : M(u) \in W' \text{ and } uM(u) \in E(G_A)\}$, and $W'' = \{w \in W' : M(w) \in U' \text{ and } wM(w) \in E(G_A)\}$. In words, W' is the neighborhood of the critical set U', while U'' and W'' represent the men and women in U' and W' who are matched in M and the corresponding matching edges are active. Due to our assumption, $u \in U''$ and $w \in W''$. We claim that $|U' \setminus U''| < |U'|$ and $\delta(U' \setminus U'') \geq \delta(U')$, which contradicts the fact that U' is critical. We remind the reader that the critical set is the unique set with maximum deficiency, so that it has the smallest cardinality and it is contained in all other subsets with maximum deficiency. Since $u \in U'' \neq \emptyset$, $|U' \setminus U''| < |U'|$ holds. From their definition we know that $|U''| = |W''|$, so it suffices to show that $\mathcal{N}_{G_A}(U' \setminus U'') \subseteq W' \setminus W''$, because in that case

$$
\begin{aligned}
\boldsymbol{\delta(U' \setminus U'')} \overset{\text{def}}{=} |U' \setminus U''| - |\mathcal{N}_{G_A}(U' \setminus U'')| \geq |U' \setminus U''| - |W' \setminus W''| &= \\
&= (|U'| - |U''|) - (|W'| - |W''|) = \\
&= |U'| - |W'| \overset{\text{def}}{=} \boldsymbol{\delta(U')},
\end{aligned}
$$

  which would prove the second part of our claim.
  What remains to show is that $\mathcal{N}_{G_A}(U' \setminus U'') \subseteq W' \setminus W''$. Suppose the contrary, i.e. that there exists an edge ab in G_A from $U' \setminus U''$ to W''. See the third graph in Figure 4. We know that $b \in W''$ by our indirect assumption, hence $a' = M(b) \in U''$ by the definition of U'', and $a' \neq a$, because $a \notin U''$. Moreover, ab and $a'b$ are edges in G_A, thus both of them are active. Therefore, $a||_b a'$, for otherwise b would have rejected one of them. In order to keep M stable, a must be matched in M with some woman b'. Since no stable edge has been rejected so far and ab does not block M, we know that $b'||_a b$, thus b' is in a's proposal tie. Edge ab' is stable and no stable edge has been rejected yet, thus ab' is active along with ab. Therefore, $ab' \in E(G_A)$ and $b' \in W'$. Moreover, $ab' \in M$, hence $a \in U''$ and $b' \in W''$ by the definition of U'' and W'', which contradicts the assumption that $a \notin U''$. □

Claim 5. *Women who have ever had an active edge must be matched in all stable matchings in $\mathcal{I}'$.*

Proof. Claim 4 shows that stable matchings allocate each man u a partner not better than his final proposal tie. If a man u proposed to woman w and yet w is unmatched in the stable matching M, then uw blocks M, which contradicts the stability of M. □

Claim 6. *If our algorithm outputs a matching, then it is stable in $\mathcal{I}'$.*

Proof. We need to show that any maximum matching M in G_A is stable, if it covers all women who have ever held a proposal. Let M be such a matching. Due to the exit criteria of the second

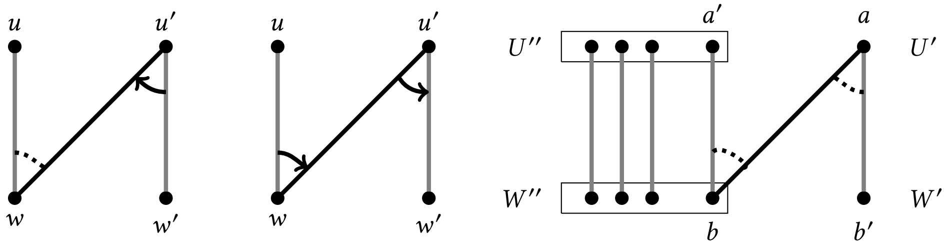

Fig. 4. The three cases in Claim 4. Gray edges are in M. The arrows point to the strictly preferred edges, while dotted arcs denote incomparability.

phase (lines 11 and 12), M covers all men. By contradiction, let us assume that M is blocked by an edge uw. This can occur in three cases.

- While w is unmatched, u does not prefer $M(u)$ to w.
  Since uw carried a proposal at the same time or before $uM(u) \in E(G_A)$ was activated, w is a woman who has held an offer during the course of the algorithm. We assumed that all these women are matched in M.
- While $w \prec_u M(u)$, w does not prefer $M(w)$ to u.
  The full tie at u containing uw must have been rejected in the algorithm, otherwise $uM(u)$ would not be an active edge. From our indirect assumption we know that either $u \prec_w M(w)$ or $u||_w M(w)$ holds. If $u \prec_w M(w)$, then $wM(w)$ had to be rejected when u proposed to w, which contradicts the fact that $wM(w) \in E(G_A)$. Hence, $u||_w M(w)$. Thus, after uw and its full tie was rejected at u, $M(w)w$ also should have been rejected in line 25 of a STRONG_REJECT procedure, which leads to the same contradiction with $wM(w) \in E(G_A)$.
- While $u \prec_w M(w)$, u does not prefer $M(u)$ to w.
  Since $uM(u)$ is an active edge, uw has carried a proposal, because $M(u)$ is not preferred to w by u. When uw was proposed along, w should have rejected $M(w)w$, to which uw is strictly preferred. This contradicts our assumption that $wM(w) \in E(G_A)$. □

Claim 7. *If $\mathcal{I}'$ admits a stable matching M', then any maximum matching M in the final G_A covers all women who have ever held a proposal.*

Proof. We first show that our algorithm always terminates. As already observed in the description, the algorithm is bound to proceed to the second phase. We now argue that even though from Phase 1 we proceed to Phase 2, and from Phase 2 it is possible to be sent back to Phase 1, the two phases cannot be iterated infinitely many times. Before each return to Phase 1 from Phase 2 (line 15), at least one active edge is rejected. Moreover, observe that a man actively proposing his last-choice dummy woman may not be in the critical set. Hence, such an edge may not be rejected. On the whole, the critical set will eventually become empty and the algorithm will proceed to the output phase (line 17).

From Claims 3 and 5 we know that M' covers all women who have ever held a proposal and all men. It is also obvious that matching M found in line 17 covers all men, for otherwise U' could not have been the empty set in line 12 and the execution would have returned to the first phase. This means that $|M| = |M'|$. On the other hand, all women covered by $M \subseteq E(G_A)$ are fit with active edges in G_A. Therefore, women covered by M represent only a subset of women who have ever had an active edge, i.e. the women covered by M'. In order to M and M' have the same cardinality, they must cover exactly the same women. Thus, M covers all women who have ever received a proposal. □

Corollary 4.4. *If $\mathcal{I}$ admits a stable matching then our algorithm outputs one.*

Proof. Courtesy of Claim 7, the output M covers all women who have ever received a proposal. According to Claim 6, this matching is stable in I', and according to Claim 3, we thus output a stable matching of $\mathcal{I}$. □

4.3 Implementation and time complexity

The proof of Theorem 4.2 is brought to conclusion by the suitable choice of data representation, a more elaborate discussion on the implementation and the time complexity analysis of Algorithm 1 (and its subroutine Algorithm 2).

4.3.1 Data representation. The cost of the execution of our algorithm on an instance $\mathcal{I}$ is estimated by the number of accesses to the data structures representing neighbors of vertices and the relations between them. We suppose that G is represented by adjacency lists belonging to $|U| + |W| = n$ vertices and that there are $|E| = m$ acceptable man-woman pairs. Since zero-degree vertices do not interfere with the existence or content of stable matchings, it may be assumed that each vertex has at least one edge, which results in $max\{|U|, |W|\} \leq m$, hence $n = |U| + |W| \leq 2m$ and $n = O(m)$. Relations in $\mathcal{R}_U$ are lists with ties, hence they can be incorporated into the adjacency lists by using a delimiter symbol between ties. However, relations in $\mathcal{R}_W$ are provided as sets of pairwise relations for each vertex $w \in W$, consisting of at most $\binom{\deg_G(w)}{2}$ ordered pairs of vertices adjacent to w, where $\deg_G(w)$ denotes the degree of w in G.

Since the algorithm queries the preference relations at women many times, it pays off to represent these relations strategically. We transform each woman w's preference structure into two bipartite graphs. These graphs are $G_{w,<}(U, \bar{U})$ and $G_{w,||}(U, \bar{U})$, where $\bar{U}$ denotes a copy of the vertex set U. For $G_{w,<}(U, \bar{U})$, there is an edge $u_1u_2 \in U \times \bar{U}$ if and only if $u_1 \prec_w u_2$, while for $G_{w,||}(U, \bar{U})$, there is an edge $u_1u_2 \in U \times \bar{U}$ if and only if $u_1||_w u_2$. The construction of these graphs in the form of adjacency lists takes $O\left(\deg_G(w)^2\right)$ time. Besides the two graphs, an indexing structure is constructed for each of them.

Furthermore, all information regarding edges in G are to be maintained. More specifically, the state of an edge as being an inactive, active, rejected inactive, or rejected proposal edge and whether it is a new proposal edge is stored. Moreover, for every $u \in U$, we store the fact whether u has been a vertex because of which edges of type $u'w$ are rejected where $u'||_w u$ (line ??). Reasonable work is spared if u happens to be in the same role again later.

4.3.2 Data representation. We suppose that the input $I = (U, W, E, \mathcal{R}_U, \mathcal{R}_W)$ is provided in the following form. The bipartite graph $G = (U \cup W, E)$ is represented by adjacency lists belonging to $|U| + |W| = n$ vertices and there are $|E| = m$ acceptable man-woman pairs. Since zero-degree vertices do not interfere with the existence or content of stable matchings, it may be assumed that each vertex has at least one edge, which results in $max\{|U|, |W|\} \leq m$, hence $n = |U| + |W| \leq 2m$. Relations in $\mathcal{R}_U$ are lists with ties, hence they are assumed to be incorporated into the adjacency lists by using a delimiter symbol between ties. Relations in $\mathcal{R}_W$ are provided separately as sets of pairwise relations for each vertex $w \in W$, consisting of at most $\binom{\deg_G(w)}{2}$ ordered pairs of vertices adjacent to w, where $\deg_G(w)$ denotes the degree of w in G.

Due to the frequent visit of the algorithm to the preference relations, these relations are represented strategically, i.e. in structures that guarantee $O(1)$ query access. As for men, their adjacency list containing women and delimiter symbols are used to create indexing arrays, which being queried using women provide the location of the women in question. Formally, if woman w is positioned at index i in man u's adjacency list, then the indexing array of u at index w contains the value i. Obviously, the identities of elements in W are coded by numbers between $1, \dots, |W|$.

As far as women are concerned, each woman w's preference structure is transformed into two sets of adjacency lists providing constant time access to men strictly worse or incomparable to an arbitrary man with respect to woman w. More precisely, these two structures are

$$G_{w,\prec} = \left\{(u, L^w_{u,\prec}) : u \in U \text{ and } L^w_{u,\prec} = \{u' \in U : u \prec_w u'\}\right\} \text{ and}$$
$$G_{w,||} = \left\{(u, L^w_{u,||}) : u \in U \text{ and } L^w_{u,||} = \{u' \in U : u||_w u'\}\right\}, \tag{1}$$

where $L^w_{u,\prec}$ ($L^w_{u,||}$) shows men strictly less preferred than (incomparable to) u with respect to w.

Furthermore, all information regarding edges in G are to be maintained. More specifically, the state of an edge as being an inactive, active, rejected inactive, or rejected proposal edge is stored in the adjacency list of the male partner. Also, the indices of the first and last element in the current tie of each man alongside with the current number of active edges is stored. A list $\mathcal{M}$ for men possessing no active edge is also maintained.

4.3.3 Implementation. Initially all men are added to the list of active-edgeless men, $\mathcal{M}$. At the start of each iteration of the loop in the first phase of Algorithms 1 (lines 3 and 4) men in $\mathcal{M}$ move onto their next tie and propose along those edges. This means that indices showing the first and last element of the current tie are updated accordingly and the number of active edges is also stored. Remember that some of these edges may have already been rejected, hence this number could be lower than the size of the tie.

The rejection of worse edges in lines 5 to 7 is similar to the rejections happening during the strong rejection phase in Algorithm 2. For a new proposal edge uw, edges $u'w$ strictly less preferred are found in the structure $G_{w,\prec}$ at the entry of u, in the list $L^w_{u,\prec}$. Analogously, edges $u'w$ incomparable to uw are found in the list $L^w_{u,||}$. Hence, pointing out partners to be rejected could be done in $O(1)$ time for each. The subsequent work of rejecting an edge $u'w$ includes: setting the state of w to the appropriate rejected state in the adjacency list of u', obviously preceded by locating w through the indexing array of u; if woman w lied on an active edge in the current tie of u' (checked based on the location and the indices of the first and last element of the tie) then the number of active edges is reduced. If the latter number has just got to zero, man u' is added back to $\mathcal{M}$ and, in case of Algorithm 2, to the set R. Therefore, the cost of a single rejection is $O(1)$. It is noteworthy that in the case of Algorithm 2, $\mathcal{M}$ differs from R, since men are added to R provisionally, and after the rejection of incomparable edges they are released.

The last issue to discuss is finding the critical set in the graph of active edges, G_A, in the second phase of Algorithm 1, in lines 10 and 11. As already stated in Section 4.1, the critical set is calculated from a maximum matching by taking the uncovered men and all men reachable from the uncovered men via an alternating path. The standard algorithm for determining maximum matchings launches parallel BFS-algorithms from uncovered men to find augmenting paths. An interesting property of the execution is that whenever it finishes—because no alternating path was augmenting—the critical set is computed as well [30]. Therefore critical sets are automatically yielded with the use of the Hungarian method [27], for which one only needs to store the occurring vertices. The graph of active edges, G_A, is constructed by iterating through adjacency lists of men and creating new lists only with the active edges. The adjacency lists of women in G_A are computed by inversion of the men's list.

Although we could apply the Hungarian method in each execution of the second phase, we wish to reduce the cost of execution by storing information from previous iterations. Note that the Hungarian method commences from an arbitrary matching and augments that one. Let the augmentation start from the remnants of the maximum matching found in the previous iteration. In each iteration, these remnants, more precisely the remaining man-woman pairs are stored at each of the partners in order to be accessible in $\mathcal{O}(1)$. Until the next iteration naturally further rejections are expected, but, again, it takes constant time to invalidate the stored partner, were the edge to be rejected.

4.3.4 Analysis. Firstly, a lower bound of the size of input is provided by the size of the graph, as usual. Note that the set of pairwise relations in $\mathcal{R}_W$ may be an empty set for any w, so this is a sharp lower bound. Hence, the input size is $\Omega(n+m)$.

Secondly, our algorithm uses the following two primitive operations:

- finding all men u' such that $u \prec_w u'$ for a woman w and rejecting these $u'w$;
- finding edges incomparable to uw at w and rejecting them.

Due to the data structures $G_{w,<}$ and $G_{w,||}$, these operations cost $\mathcal{O}(1)$ for each man u'.

The construction of the graphs $G_{w,<}, G_{w,||}$ costs $\mathcal{O}\left(\deg_G(w)^2\right)$ time for each woman. Therefore, the construction of all such graphs adds up to $\mathcal{O}\left(\sum_w \deg_G(w)^2\right)$. However,

$$\sum_w \deg_G(w)^2 \leq \left(\sum_w \deg_G(w)\right)^2 = m^2,$$

thus the cost of construction is $\mathcal{O}\left(m^2\right)$.

Adding dummy women to the list of men is done in $\mathcal{O}(n)$ time in total. Besides, each edge is proposed along at most once and proposals are to be made in order of the adjacency list of men, so the total cost of proposals is $\mathcal{O}(m)$. Furthermore, beware that for a given edge uw, rejecting edges $u'w$ to whom uw is strictly preferred, and rejecting incomparable edges $u'w$ are done at most once, each of them contributing a cost of $\mathcal{O}(1)$. The graph G_A need not be constructed separately, since active edges are marked due to our previous considerations. Subsequently, apart from finding maximum matchings and critical sets in G_A, the cost of our algorithm is bounded by $\mathcal{O}\left(m^2+n+m+2m\right) \subseteq \mathcal{O}\left(m^2\right)$.

As far as maximum matchings and critical sets are concerned, the well-founded technique described by Irving [19] is reapplied here.

Let $M_i, C_i, x_i, (i \geq 1)$ denote the maximum matching found in the i^{th} iteration of the second phase, the critical set with respect to M_i, and the number of edges rejected between the i^{th} and $(i+1)^{\text{th}}$ execution of the Hungarian method, respectively. In the first iteration the augmenting path algorithm is executed from scratch taking $\mathcal{O}(|U|m) \subseteq \mathcal{O}(nm)$ time. After the i^{th} iteration we reject x_i edges. Since each man in C_i had at least one edge in G_A, at least $(|U|-|C_i|)-(x_i-|C_i|) = |U|-x_i$ men are still paired to women via active edges, if that number is positive. In that case, the $(i+1)^{\text{th}}$ iteration starts BFS-algorithms from at most x_i vertices. Let L be the total number of iterations, in k of which $x_i \geq |U|$. In all such cases the computational complexity of calculating the maximum matching is still upper bounded by the cost of finding a maximum matching from scratch. The time complexity, therefore, is $\mathcal{O}(nm + k \cdot |U|m + m\sum_{L-k \text{ iter}} x_i)$, where the summation is done for the rest of x_i's corresponding to the remaining $L-k$ iterations. In the rest of the k iterations $|U| \leq x_i$, therefore $|U|k + \sum_{L-k \text{ iter}} x_i \leq \sum_{i=1}^{L} x_i \leq m$, because at most m edges may be rejected and no edge is rejected more than once. Hence the running time related to maximum matchings and critical sets is $\mathcal{O}(nm + m \cdot (|U|k + \sum_{L-k \text{ iter}} x_i)) \subseteq \mathcal{O}(nm + m \cdot m) \subseteq \mathcal{O}\left(m^2\right)$.

In conclusion, the total time complexity of the algorithm is $\mathcal{O}\left(m^2\right)$, while the size of the input is $\Omega(n+m)$. Hence, the algorithm is clearly polynomial.

4.3.5 Time complexity analysis. Firstly, a lower bound of the size of input is provided by the size of the graph, as usual. Note that the set of pairwise relations in $\mathcal{R}_W$ may be an empty set for any w, so this is a sharp lower bound. Hence, the input size is $\Omega(n+m) = \Omega(m)$.

We now go over the steps defined in subsections 4.3.2 and 4.3.3 regarding the data preprocess and implementation details. The construction of indexing arrays for men cost $O(n)$ for each man, hence the entire indexing table is produced in $O\left(n^2\right) \subseteq O\left(m^2\right)$. The two sets of adjacency lists possessed by any woman w, namely $G_{w,<}$ and $G_{w,||}$, takes $O\left(\deg_G(w)^2\right)$ time each to construct, since the preference relation corresponding to w needs to be processed. Therefore, the construction of all such structures adds up to $O\left(\sum_w \deg_G(w)^2\right)$. However,

$$\sum_w \deg_G(w)^2 \leq \left(\sum_w \deg_G(w)\right)^2 = m^2, \tag{2}$$

thus the cost of construction is $O\left(m^2\right)$.

Adding dummy women to the list of men is done in $O(n)$ time in total. Besides, each edge is proposed along at most once and proposals are to be made in order of the adjacency list of men, so the total cost of proposals (i.e. updating the indices showing the boundaries of ties, updating the number of active edges) is $O(m)$.

As we could see previously in subsection 4.3.3, pointing out edges that need to be rejected and the subsequent administrative work add up to an $O(1)$ cost. The number of required rejections remains to be discussed. It is straightforward that a rejection of an edge $u'w$ caused by a strictly preferred, freshly proposed edge uw could happen at most once. Moreover, it is noticed that throughout the execution, in the strong rejection procedure in Algorithm 2, set R may contain any man multiple times, but one fully rejected proposal tie is only processed once. Hence, any edge $u'w$ could be rejected due to an incomparable $u'w$ edge at most once. Hence, any edge $u'w$ is rejected at most $2\deg_G(w)$ times in line 6 and the strong rejection procedure in Algorithm 2. Lastly, any edge is rejected as an active edge incident to the critical set in lines 12 and 13 at most once. Thus, any edge is rejected at most $O\left(\deg_G(w)\right)$ times with an $O(1)$ cost. Summing up, the overall cost of edge rejections becomes

$$O\left(\sum_{\text{edge in } w \in W} \deg_G(w)\right) = O\left(\sum_{w \in W} \deg_G(w)^2\right) \subseteq O\left(m^2\right),$$

as already shown previously in (2).

The construction of G_A is, again, equivalent to iterating through the adjacency lists of men in U, thus the time complexity is $O(m)$. Nonetheless, each time the second phase fails to resume the execution in the output phase, at least one active edge is rejected. However, any edge may become active only once, thus the second phase is executed at most $O(m)$ time, making the overall cost of building these graphs $O\left(m^2\right)$.

Subsequently, apart from finding maximum matchings and critical sets in G_A, the cost of our algorithm is bounded by $O\left(m^2\right)$.

As far as maximum matchings and critical sets are concerned, the well-founded technique described by Irving [19] is reapplied here. Remember that the augmenting path algorithm is always launched from the remnants of the previously computed maximum matching. Loss of edges is imminent due to rejections based on nonempty critical sets and the reapplication of the first phase. Let M_i, C_i, x_i, $(i \geq 1)$ denote the maximum matching found in the i^{th} iteration of the second phase, the critical set with respect to M_i, and the number of edges rejected between the i^{th} and

$(i+1)^{\text{th}}$ execution of the Hungarian method, respectively. In the first iteration the augmenting path algorithm is executed from scratch taking $O(|U|m) \subseteq O(nm)$ time. After the i^{th} iteration we reject x_i edges. Since each man in C_i had at least one edge in G_A, at least $(|U|-|C_i|)-(x_i-|C_i|) = |U|-x_i$ men are still paired to women via active edges, if that number is positive. In that case, the $(i+1)^{\text{th}}$ iteration starts BFS-algorithms from at most x_i vertices. Let L be the total number of iterations, in k of which $x_i \geq |U|$. In all such cases the computational complexity of calculating the maximum matching is still upper bounded by the cost of finding a maximum matching from scratch. The time complexity, therefore, is $O(nm + k \cdot |U|m + m \sum_{L-k \text{ iter}} x_i)$, where the summation is done for the rest of x_i's corresponding to the remaining $L-k$ iterations. In the rest of the k iterations $|U| \leq x_i$, therefore $|U|k + \sum_{L-k \text{ iter}} x_i \leq \sum_{i=1}^{L} x_i \leq m$, because at most m edges may be rejected and no edge is rejected more than once. Hence the running time related to maximum matchings and critical sets is $O(nm + m \cdot (|U|k + \sum_{L-k \text{ iter}} x_i)) \subseteq O(nm + m \cdot m) \subseteq O(m^2)$.

In conclusion, the total time complexity of the algorithm is $O(m^2)$, while the size of the input is $\Omega(m)$. Hence, the algorithm is clearly polynomial.

5 SUPER-STABILITY

In super-stability, an edge outside of M blocks M if *neither* of its end vertices *prefer* their matching edge to it strictly.

Definition 5.1 (blocking edge for super-stability). Edge uw blocks M, if

(1) $uw \notin M$;
(2) $w \prec_u M(u)$ or $w||_u M(u)$;
(3) $u \prec_w M(w)$ or $u||_w M(w)$.

The set of already investigated problems is remarkable for super-stability, see Table 1. Up to posets on both sides, a polynomial algorithm is known to decide whether a stable solution exists [19, 33]. Even though it is not explicitly written there, a blocking edge in the super stable sense is identical to the definition of a blocking edge given in [11]. It is shown there that if one vertex class has strictly ordered preference lists and the other vertex class has arbitrary relations, then determining whether a stable solution exists is NP-complete, but if the second class has asymmetric lists, then the problem becomes tractable.

We first show that a polynomial algorithm exists up to partially ordered relations on one side and asymmetric relations on the other side. Our algorithm can be seen as an extension of the one in [11]. Our added contributions are a more sophisticated proposal routine and the condition on stability in the output. These are necessary as men are allowed to have acyclic preferences instead of strictly ordered lists, as in [11]. Finally, we prove that acyclic relations on both sides make the problem hard.

5.1 Algorithm

Theorem 5.2. *If one side has posets as preferences, while the other side has asymmetric pairwise preferences, then deciding whether the instance admits a super stable matching (and outputting one, if so) can be done in $O(m^2)$ time.*

We prove this theorem by designing an algorithm that produces a stable matching or a proof for its nonexistence. For a pseudocode, see Algorithm 3. We assume men to have posets as preferences and women to have asymmetric relations. We remark that non-empty posets always have a non-empty set of *maximal elements*: these are the ones that are not dominated by any other element. Women in the set of maximal elements are called *maximal* women.

At start, an arbitrary man proposes to one of his maximal women (lines 29-30). An offer from u is temporarily accepted by w if and only if $u \prec_w u'$ for every man $u' \neq u$ who has ever proposed to w (line 32). This rule forces each woman to reassess her current match every time a new proposal arrives. Accepted offers are called *engagements*. The proposal edges or engagements not meeting the above requirement are immediately deleted from the graph (lines 34-37), in other words, the corresponding pair is removed from the acceptability relation. Notice that a woman can reject a proposal even if she is currently not engaged. Each man then reexamines the poset of women still on his list. If any of the maximal women is not holding an offer from him, then he proposes to her. The process terminates and the output is generated when no man has maximal women he has not proposed to. Notice that while women hold at most one proposal at a time, men might have several engagements at termination. We output the set of engagements as a super stable matching if it is indeed a matching that covers all women who have ever received a proposal (line 42). In all other cases, no super stable matching exists (line 44).

Algorithm 3 Super stable matching with posets and asymmetric relations

```
Input: $\mathcal{I} = (U, W, E, \mathcal{R}_U, \mathcal{R}_W)$; $\mathcal{R}_U$: posets, $\mathcal{R}_W$: asymmetric.
29: while there is a man $u$ who has not proposed to a currently maximal woman $w$ do
30:     $u$ proposes to $w$
31:     if $u \prec_w u'$ for all $u' \in U$ who have ever proposed to $w$ then
32:         $w$ accepts the proposal of $u$, $uw$ becomes an engagement
33:     else
34:         $w$ rejects the proposal and deletes $uw$
35:     end if
36:     if $w$ had a previous engagement to $u'$ and $u \prec_w u'$ or $u||_w u'$ then
37:         $w$ breaks the engagement to $u'$ and deletes $u'w$
38:     end if
39: end while
40: let $M$ be the set of engagements
41: if $M$ is a matching that covers all women who have ever received a proposal then
42:     STOP, OUTPUT $M$ and "$M$ is a super stable matching."
43: else
44:     STOP, OUTPUT "There is no super stable matching."
45: end if
```

Example 5.3 illustrates our algorithm.

Example 5.3. The instance in Figure 5 consist of four men with posets as preferences, and four women with asymmetric preferences. Numbers on the edges express preferences wherever the pairwise relations translate into a strict order or a list with ties, which is the case for vertices $m_1, m_2, m_3 w_1, w_3$ and w_4. Three neighbors of m_4 can be ordered strictly, but there is no information on his preferences on w_1, thus this woman is incomparable to the other three. About w_2 we know that she ranks m_1 above all other neighbors, and she has a cyclic preference relation over these three other neighbors, as shown in the figure. Arrows point towards the preferred edge.

First, m_1 proposes to his sole maximal woman w_1, and they become engaged. Then, m_2 proposes to the same woman, who now reconsiders her match, rejects m_1 (because of line 37) and accepts m_2 instead (because of line 32). Now m_2 proposes to w_2 as well, because she is a maximal woman, and they become engaged. Similarly to m_2, m_3 also proposes to both of his neighbors, and gets engaged

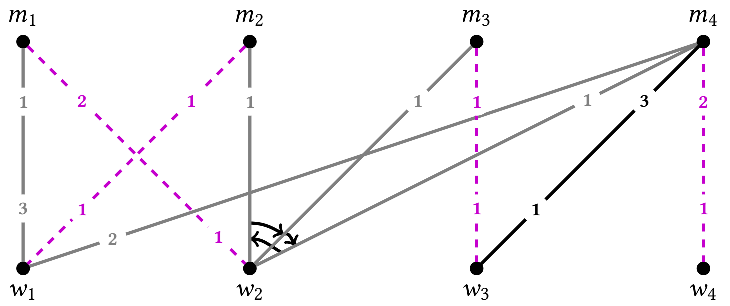

Fig. 5. The input for Example 5.3. Colored dashed edges mark the final matching, while solid gray edges are the ones rejected in our the algorithm.

to both of them. Then, w_2 rejects m_2, since $m_3 \prec_{w_2} m_2$. The final man, m_4 proposes to his maximal women w_1 and w_2. The first one rejects the proposal, while the latter one accepts the it and turns down the offer from m_2. The current set of engagements is thus $\{m_2w_1, m_3w_3, m_4w_2\}$. The only man who has a maximal woman he has not proposed to is m_1, therefore, he proposes next. This proposal is made to w_2, who accepts it and rejects her current partner m_4. This triggers a proposal along m_4w_4, which becomes an engagement. Since no man has a maximal woman he has not proposed to, the algorithm terminates here. The set of engagements $\{m_1w_2, m2w_1, m_3w_3, m_4w_4\}$ is indeed a matching that covers all women who ever received a proposal, thus it is a super stable matching.

We are now ready to prove the correctness of our algorithm. Theorem 5.4 is supported in one direction by Claim 8, in the other direction by Claims 9 to 12.

Theorem 5.4. *The output of Algorithm 3 is a matching that covers all women who ever received a proposal if and only if the instance admits a stable matching.*

Claim 8. *If the output of the algorithm is a matching that covers all women who ever received a proposal, then it is stable.*

Proof. Assume that an edge uw blocks the output matching M. We investigate two cases.

- Man u has proposed to w.
  Since uw was rejected and w is covered in M, we know that w got engaged to a man $M(w)$, for whom $M(w) \prec_w u$ holds. This contradicts our assumption on uw being a blocking edge.
- Man u has not proposed to w.
  The only reason for u not proposing to w is that w has never been a maximal woman for u, i.e. even at termination, u had an engagement edge strictly preferred to uw. Since M is a matching, this edge must be $uM(u)$. The relation $M(u) \prec_u w$ contradicts our assumption on uw being a blocking edge. □

Claim 9. *If an edge was deleted in the algorithm, then no stable matching contains it.*

Proof. Let uw be the first edge deleted by the algorithm, even though it is part of a stable matching S. The reason of the deletion was that w received an offer from u', for which $u' \prec_w u$ or $u'||_w u$. Since $u'w \notin S$ does not block S, u' is matched in S and $S(u') \prec_{u'} w$. Due to the monotonicity of proposals, u' had proposed to $S(u')$ before proposing to w, but $u'S(u')$ was deleted. This contradicts our assumption on uw being the first deleted stable edge. □

Claim 10. *If a woman w has ever received a proposal in our algorithm, then w must be matched in all stable matchings.*

Proof. Assume that uw carried a proposal at some point, yet w is unmatched in a stable matching S. In order to stop uw from blocking S, u needs to be matched in S and $S(u) \prec_u w$. This implies that $uS(u)$ was deleted before the proposal along uw was sent, which contradicts Claim 9. □

Claim 11. *If there is a stable matching S, then the set of engagements M computed in line 40 is a matching.*

Proof. As already mentioned, the only reason for M not being a matching can be that a man has more than one edge in M. On the other hand, each of the women covered by M have degree 1 in M and they are all matched in S, due to Claim 10. This is only possible if there is a man u who is matched in S, but not covered by M. To stay unmatched in M, u must have proposed to and be rejected by all women adjacent to him, including $S(u)$. This contradicts our Claim 9 on stable edges never being rejected by the algorithm. □

Claim 12. *If there is a stable matching S, then the set of engagements M computed in line 40 covers all women who have ever received a proposal.*

Proof. We will show a stronger statement, claiming that M covers all women matched in S. Due to Claim 10, which states that women covered by an arbitrarily chosen stable matching are a superset of women covered by M, M then covers all women who have ever received a proposal.

Claim 11 shows that M is in fact a matching, thus, the symmetric difference $M \triangle S$ consists of alternating paths and cycles. Our goal is to show that no alternating path can start at a $w \in W$ vertex with an edge in S.

Assume the contrary. Claim 9 proves that $uw \in S \setminus M$ has never been proposed along. Otherwise, $uw \notin M$ was deleted, in which case $uw \in S$ cannot hold. This implies that u has a partner in M and he prefers $M(u)$ strictly to w. To stop $uM(u)$ from blocking S, $M(u)$ must have a partner in S who is strictly preferred to u. Just as before, this edge has never carried a proposal, otherwise $uM(u)$ could not be in M. These arguments can be iterated, and thus the assumed alternating path can never end, which is a contradiction to the finiteness of the graph. □

Analysis and time complexity of Algorithm 3. We use a similar data structure to the one applied in the analysis of Algorithms 1 and 2. The only difference emerges from the poset preference structure on the men's side. We store the entire partial order for each man, given as a Hasse diagram of the underlying directed acyclic graph of the poset. The Hasse diagram provides a non-redundant representation of a poset. Formally, the Hasse diagram is a directed acyclic graph whose vertices are the elements of the finite poset and there is an edge between vertices a and b if and only if $a \prec b$ and there exists no element c such that $a \prec c \prec b$ holds. Each man is equipped with a dummy woman, from whom there is a directed edge to all initially maximal women. The cost of the execution is again measured in the number of accesses to these data structures.

Similarly to the implementation of our strongly stable algorithm, it is worth constructing a special structure $G_{w,\succeq}(U, \bar{U})$ for each woman w, where $\bar{U}$ again denotes a copy of the vertex set U and there is an edge $u_1 u_2 \in U \times \bar{U}$ if and only if $u_2 \prec_w u_1$ or $u_1 ||_w u_2$. The construction of this structure summed up for all women takes $O\left(m^2\right)$ time.

Since the set of relations can be empty as well, the size of the input is analogously lower bounded by $\Omega(n+m)$. The assumption of Hasse diagrams allows a straightforward check whether all maximal women have been proposed to. The initial maximal set is the women directly connected to the dummy woman. Each time a woman w turns down a proposal, the candidates for being promoted to maximal state are the women adjacent to w in the Hasse diagram. Therefore the cost of submitting proposals does not exceed $O(m)$. The rest of the while loop, from lines 31 to 37, concerns the asymmetric relations on the woman's side.

One needs to iterate through the relations belonging to woman w and check whether the new proposal from U is strictly preferred to all previous proposals, and whether the previous engager u' is strictly preferred to u. This primitive operation is done through the structure $G_{w,\succeq}(U, \bar{U})$. One only needs to analyze whether there is at least one neighbor of u who has already proposed to w. The same property is checked for u'. The cost of these operations is $O(n)$. Deleted edges are kept in our data structure with a label "rejected", so that previous proposals can be checked. Finally, the computation of M and the examination of the output condition can be done in $O(m)$ time, because engagements are already marked. Consequently, the time complexity of the algorithm is $O\left(m^2\right) + O(m \cdot n) + O(m) = O\left(m^2\right)$.

5.2 Hardness

We complete the study of all cases for super-stability with the following theorem.

Theorem 5.5. *If both sides have acyclic pairwise preferences, then determining whether a super stable matching exists is* NP*-complete, even if each agent finds at most four other agents acceptable.*

Proof. The NP-complete problem we reduce to our problem is again (2,2)-e3-sat [4]. Our construction follows the same logic as the one in the proof of Theorem 3.4, however, the preferences are set differently, see Figure 6.

The vertex and edge sets created to B is identical to the one in the proof of Theorem 3.4. To each variable x in B, we create vertices $t, \bar{x}, f$, and x, and to each clause in B, we create vertices u_1, u_2, u_3, w_1, w_2, and w_3. The edge set inside each variable gadget comprises a 4-cycle $t, \bar{x}, f, x$. A clause gadget consists of a complete bipartite graph on the six created vertices, where each vertex on the u-side is equipped with exactly one interconnecting edge, each of which runs from the u-vertex to vertex x or $\bar{x}$ in the variable gadget of the literal. If the literal is positive, then the other end vertex of this edge is x in the corresponding variable gadget, otherwise it is $\bar{x}$.

The preferences of each vertex restricted to the edges that are not interconnecting edges can be represented by a strictly ordered list. At x, each interconnecting edge xu_i is incomparable to xf and worse than xt, and the same holds for $\bar{x}$. At u_i, the interconnecting edge xu_i or $\bar{x}u_i$ is incomparable to u_iw_3, and it is worse than u_iw_1 and u_iw_2.

Claim 13. *If there is a truth assignment that satisfies the Boolean formula B, then there is a super stable matching in G.*

Proof. In each variable gadget belonging to a true variable, $\{tx, f\bar{x}\}$ is chosen, whereas all gadgets corresponding to a false variable contribute matching $\{fx, t\bar{x}\}$. In each clause, there is at least one true literal. The vertex representing the appearance of this literal is matched to w_3 in the clause gadget, while the remaining four vertices are coupled up in such a way that no edge inside of the gadget blocks. This is possible, because $\{u_1w_1, u_2w_3, u_3w_2\}$, $\{u_1w_2, u_2w_1, u_3w_3\}$, and $\{u_1w_3, u_2w_2, u_3w_1\}$ are all stable matchings. The reason why the literal satisfying the clause was chosen to be matched to w_3 is that its matching edge in the variable gadget is strictly preferred to its interconnecting edge, and thus the interconnecting edge does not block M. Due to the strict preferences inside gadgets, it is easy to check that no other edge blocks the constructed matching. □

Claim 14. *If there is a super stable matching M in G, then there is a truth assignment that satisfies B.*

Proof. If either t or f is unmatched in M, then at least one of their x and $\bar{x}$ vertices is either unmatched or it is matched along an interconnecting edge. In both cases, this vertex has a blocking edge leading to the unmatched t or f. With this we have already shown three statements:

(1) for each variable gadget, either $\{tx, f\bar{x}\} \in M$ or $\{fx, t\bar{x}\} \in M$;

		u_1 :	$w_1 \prec x/\bar{x}, w_2 \prec x/\bar{x}, x/\bar{x}\vert\vert w_3$; strict list: $w_1 \prec w_3 \prec w_2$
t :	strict list: $x \prec \bar{x}$	u_2 :	$w_1 \prec x/\bar{x}, w_2 \prec x/\bar{x}, x/\bar{x}\vert\vert w_3$; strict list: $w_3 \prec w_2 \prec w_1$
f :	strict list: $\bar{x} \prec x$	u_3 :	$w_1 \prec x/\bar{x}, w_2 \prec x/\bar{x}, x/\bar{x}\vert\vert w_3$; strict list: $w_2 \prec w_1 \prec w_3$
x :	$f \prec t, t \prec u, u\vert\vert f$	w_1 :	strict list: $u_2 \prec u_3 \prec u_1$
$\bar{x}$:	$t \prec f, t \prec u, u\vert\vert f$	w_2 :	strict list: $u_1 \prec u_2 \prec u_3$
		w_3 :	strict list: $u_3 \prec u_1 \prec u_2$

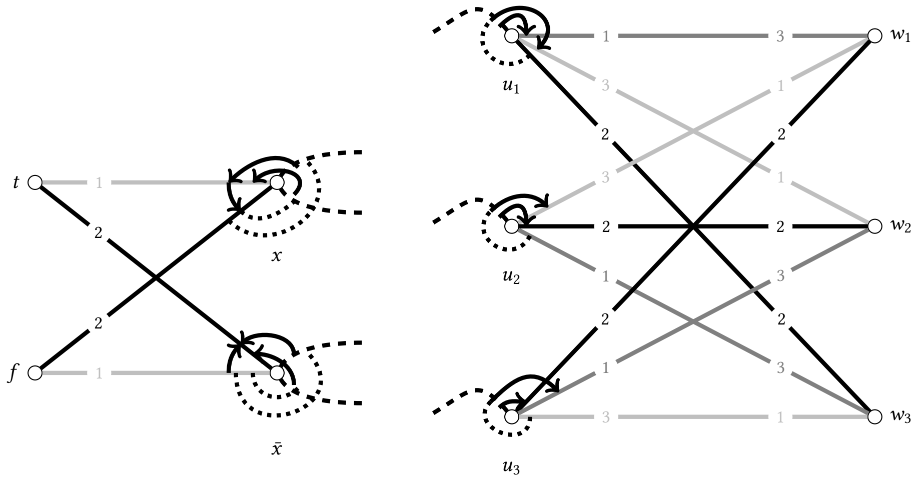

Fig. 6. A variable gadget to the left and a clause gadget to the right. Interconnecting edges are dashed. The arrows point to the preferred edge, while dotted arcs denote incomparability.

(2) no interconnecting edge appears in M;
(3) M is perfect in each clause gadget, since stable matchings are inclusionwise maximal matchings.

From the last point we can see that in each clause gadget, exactly two u-vertices are matched to partners strictly preferred to their interconnecting edge. Therefore, each clause gadget has exactly one interconnecting edge that is incomparable to the edge in M at the clause gadget. In order to ensure stability, this edge must be worse than the edge in M at its variable gadget. This only happens if the corresponding literal is satisfied in the truth assignment. We have now proved that each clause is satisfied. □

With this, we have completed our hardness proof. □

6 THE STABLE ROOMMATES PROBLEM

In this section, we determine the complexity of finding a weakly, strongly, and super stable matching in non-bipartite instances, depending on the preference structure of the agents. Then we extend the well-known Rural Hospitals Theorem for the most general setting, remark on structural results related to this theorem, and also pose some open questions.

6.1 Complexity results

The six degrees of orderedness can be interpreted in the non-bipartite stable roommates problem as well. For strictly ordered preferences, all three notions of stability reduce to the classical stable roommates problem, which can be solved in $O(m)$ time [18]. The weakly stable variant becomes

NP-complete already if ties are present [36]. The strongly stable version with ties can be solved in polynomial time [28, 37], but it is NP-complete for posets [23]. For super-stability, there is an $O(m)$ time algorithm for preferences ordered as posets [20], while the case with acyclic preferences was shown in Theorem 5.5 to be NP-complete for bipartite instances as well. Hence, because of the trivial reduction of the bipartite problem to the more general, non-bipartite problem, the super-stable roommates problem with structures that are at least as general as acyclic preferences, is NP-complete. Due to our new results, the complexity analysis of all cases has thus been completed. We summarize the known and the new results in Table 2.

	strict	ties	poset	acyclic	asymmetric	arbitrary
WEAK	$O(m)$ [14]	NPC [36]	NPC [36]	NPC [36]	NPC [36]	NPC [36]
STRONG	$O(m)$ [14]	$O(nm)$ [28, 37]	NPC [23]	NPC [23]	NPC [23]	NPC [23]
SUPER	$O(m)$ [14]	$O(m)$ [20]	$O(m)$ [20]	NPC	NPC	NPC

Table 2. The complexity table for the stable roommates problem with pairwise preferences. Unlike in Table 1, rows here represent the three stability notions, while columns stand for the degree of orderedness in each agent's preferences. The colored fields mark our contribution, and they follow from Theorem 5.5.

6.2 Structural results

The Rural Hospitals Theorem for strict lists [15] states that the set of matched vertices is identical in all stable matchings. In the case of preferences with ties, the theorem has been shown to hold for strongly and super stable matchings even for non-bipartite instances [21, 32], and fail for weak stability even for bipartite graphs. We show here that the positive results carry over even to asymmetric pairwise preferences.

Theorem 6.1. *In an instance of the strongly/super stable roommates problem with asymmetric pairwise preferences, all stable matchings cover the same set of vertices.*

Proof. We rely on the usual alternating path argument. Assume that there is a vertex v that is covered by a stable matching M_1, but left uncovered by another stable matching M_2. Then, for both strong and super stability, $M_1(v)$ must strictly prefer its partner in M_2 to v, otherwise edge $vM_1(v)$ blocks M_2. Iterating this argument, we derive that such a v cannot exist in a finite graph. □

We remark that the Rural Hospitals Theorem does not hold for arbitrary preferences, not even for bipartite instances. In the instance consisting of one man u and two women w_1 and w_2, so that $w_1 \sim_u w_2$, both uw_1 and uw_2 are strongly and super stable matchings.

The Rural Hospitals Theorem often indicates a rich underlying structure of the set of stable matchings. Such results were shown in the case of preferences with ties [28], and occasionally even for posets [32].

Bipartite instances. Strongly stable matchings are known to form a distributive lattice [32], and there is a partial order with $O(m)$ elements representing all strongly stable matchings [29]. However, once posets are allowed in the preferences, the lattice structure falls apart [32]. The set of super stable matchings has been shown to form a distributive lattice even if preferences are expressed in the form of posets [32, 38]. The questions arise naturally: does this distributive lattice structure carry over to more advanced preference structures in the super stable case? Also, even if no distributive lattice exists on the set of strongly stable matchings, is there any other structure and if so, how far does it extend in terms of orderedness of preferences?

Non-bipartite instances. For strong stability in the presence of ties in lists, Kunysz [28] showed that there exists a partial order with $O(m)$ elements representing the set of all strongly stable matchings, and he also gave an $\mathcal{O}(nm)$ algorithm for constructing such a representation. For super stable matchings with posets, Fleiner et al. [13] gave algorithms for computing all super stable pairs, enumerating all super stable matchings, and finding a minimum regret super stable matching. Similar structural results for preferences given as more involved relations are not known for either of the two stability notions, and they are left for future research.

7 OPEN PROBLEM

In Section 4 we presented an $\mathcal{O}(m^2)$-time algorithm computing a strongly stable matching or reporting that none exists, given an instance of the stable marriage problem with ties on one side, and asymmetric pairwise preferences on the other side. This matches the complexity shown by Irving [19] under the same stability criteria, but with ties on both sides in a complete instance. The completeness of the preferences was later relaxed by Manlove [31], who ensured the same time complexity. Kavitha et al. [26] designed a breakthrough $\mathcal{O}(nm)$-algorithm for the latter problem, by introducing the notion of a *level* defined on vertices, edges, and matchings. Each execution of their algorithm is a particular execution of the algorithm of Manlove [31]. However, the algorithm from [26] makes sure to always keep the actual matching *level-maximal*, which allows to share the cost of the execution among women and to upper bound the individual cost by $\mathcal{O}(m)$. It is an interesting question whether this approach could be applied in a way to reduce the complexity of our algorithm. It seems to be challenging to implement the idea in our case for the following reason. The upper bound given by Kavitha et al. [26] is proven by a simple observation that holds for the algorithms designed by Irving [19] and Manlove [31]: any woman may only keep proposals from her last tie. Hence, whenever a woman receives a new proposal edge, she either keeps all of her previous proposal edges, or drops all of them. This is a property that clearly does not even hold when applying partially ordered sets for women. Hence, the direct application of the technique is not effortless, and thus its extension would be a nice contribution.

Acknowledgment. The authors thank Tamás Fleiner, David Manlove, and Dávid Szeszlér for fruitful discussions on the topic. Our anonymous reviewers helped us to improve the presentation of the paper and to simplify the algorithm in Section 4. An earlier version of this paper has appeared at STACS 2019.

REFERENCES

[1] Abraham, D.J.: Algorithmics of two-sided matching problems. Master's thesis, University of Glasgow, Department of Computing Science (2003)

[2] Alesina, A., Rosenthal, H.: A theory of divided government. Econometrica 64(6), 1311–1341 (1996), http://www.jstor.org/stable/2171833

[3] Aziz, H., Biró, P., Fleiner, T., Gaspers, S., de Haan, R., Mattei, N., Rastegari, B.: Stable matching with uncertain pairwise preferences. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. pp. 344–352. International Foundation for Autonomous Agents and Multiagent Systems (2017)

[4] Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of MAX-3SAT (2003), electronic Colloquium on Computational Complexity Report, number 49

[5] Biró, P., Manlove, D.F., Mittal, S.: Size versus stability in the marriage problem. Theoretical Computer Science 411, 1828–1841 (2010)

[6] Biró, P.: Applications of matching models under preferences. Trends in Computational Social Choice p. 345 (2017)

[7] Biró, P., Kiselgof, S.: College admissions with stable score-limits. Central European Journal of Operations Research 23(4), 727–741 (2015)

[8] Blavatsky, P.: Content-dependent preferences in choice under risk: heuristic of relative probability comparisons. IIASA Interim Report IR-03-031 (2003)

[9] Chen, L.: University admission practices – Ireland, MiP country profile 8. http://www.matching-in-practice.eu/higher-education-in-ireland/ (2012)

[10] Condorcet, M.: Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. L'Imprimerie Royale (1785)

[11] Farczadi, L., Georgiou, K., Könemann, J.: Stable marriage with general preferences. Theory of Computing Systems 59(4), 683–699 (2016)

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

[13] Fleiner, T., Irving, R.W., Manlove, D.F.: Efficient algorithms for generalised stable marriage and roommates problems. Theoretical Computer Science 381, 162–176 (2007)

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

[15] Gale, D., Sotomayor, M.: Some remarks on the stable matching problem. Discrete Applied Mathematics 11, 223–232 (1985)

[16] Hall, P.: On representatives of subsets. Journal of the London Mathematical Society 10, 26–30 (1935)

[17] Humphrey, S.: Non-transitive choice: Event-splitting effects or framing effects? Economica 68(269), 77–96 (2001)

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

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

[20] Irving, R.W., Manlove, D.F.: The stable roommates problem with ties. Journal of Algorithms 43, 85–105 (2002)

[21] Irving, R.W., Manlove, D.F., Scott, S.: The Hospitals / Residents problem with ties. In: Halldórsson, M.M. (ed.) Proceedings of SWAT '00: the 7th Scandinavian Workshop on Algorithm Theory. Lecture Notes in Computer Science, vol. 1851, pp. 259–271. Springer (2000)

[22] Irving, R.W., Manlove, D.F., Scott, S.: Strong stability in the Hospitals / Residents problem. Tech. Rep. TR-2002-123, University of Glasgow, Department of Computing Science (2002), revised May 2005

[23] Irving, R.W., Manlove, D.F., Scott, S.: Strong stability in the Hospitals / Residents problem. In: Proceedings of STACS '03: the 20th Annual Symposium on Theoretical Aspects of Computer Science. Lecture Notes in Computer Science, vol. 2607, pp. 439–450. Springer (2003), full version available as [22]

[24] Kahn, A.B.: Topological sorting of large networks. Commun. ACM 5(11), 558–562 (Nov 1962), http://doi.acm.org/10.1145/368996.369025

[25] Kalenscher, T., Tobler, P.N., Huijbers, W., Daselaar, S.M., Pennartz, C.M.: Neural signatures of intransitive preferences. Frontiers in Human Neuroscience 4, 49 (2010)

[26] Kavitha, T., Mehlhorn, K., Michail, D., Paluch, K.E.: Strongly stable matchings in time $O(nm)$ and extension to the hospitals-residents problem. ACM Transactions on Algorithms 3(2), 15 (2007)

[27] Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)

[28] Kunysz, A.: The strongly stable roommates problem. In: LIPIcs-Leibniz International Proceedings in Informatics. vol. 57. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)

[29] 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. Society for Industrial and Applied Mathematics (2016)

[30] Lovász, L., Plummer, M.D.: Matching theory, vol. 367. American Mathematical Soc. (2009)

[31] Manlove, D.F.: Stable marriage with ties and unacceptable partners. Tech. Rep. TR-1999-29, University of Glasgow, Department of Computing Science (January 1999)

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

[33] Manlove, D.F.: Algorithmics of Matching Under Preferences. World Scientific (2013)

[34] May, K.O.: Intransitivity, utility, and the aggregation of preference patterns. Econometrica 22(1), 1–13 (1954)

[35] Ríos, I., Larroucau, T., Parra, G., Cominetti, R.: College admissions problem with ties and flexible quotas. Tech. rep., SSRN (2014)

[36] Ronn, E.: On the complexity of stable matchings with and without ties. Ph.D. thesis, Yale University (1986)

[37] Scott, S.: A study of stable marriage problems with ties. Ph.D. thesis, University of Glasgow, Department of Computing Science (2005)

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

[39] WWWUSelection: Huffington Post 2016 General Election: Trump vs. Sanders. https://elections.huffingtonpost.com/pollster/2016-general-election-trump-vs-sanders (2018), accessed 3 September 2018

[40] WWWUSelection: RealClearPolitics website. General Election: Trump vs. Sanders. https://www.realclearpolitics.com/epolls/2016/president/us/general_election_trump_vs_sanders-5565.html (2018), accessed 3 September 2018