Stable Matching: Dealing with Changes in Preferences

Rohith Reddy Gangam
University of California, Irvine
Irvine, USA
rgangam@uci.edu

Tung Mai
Adobe Research
San Jose, USA
tumai@adobe.com

Nitya Raju
University of Maryland, College Park
College Park, USA
nraju@umd.edu

Vijay V. Vazirani
University of California, Irvine
Irvine, USA
vazirani@ics.uci.edu

ABSTRACT

We study stable matchings that are robust to preference changes in the two-sided stable matching setting of Gale and Shapley [18]. Given two instances A and B on the same set of agents, a matching is said to be *robust* if it is stable under both instances. While prior work has considered the case where a *single agent* changes preferences between A and B, we allow *multiple agents on both sides* to update their preferences and ask whether three central properties of stable matchings extend to robust stable matchings: (i) Can a robust stable matching be found in polynomial time? (ii) Does the set of robust stable matchings form a lattice? (iii) Is the fractional robust stable matching polytope integral?

We show that all three properties hold when any number of agents on one side change preferences, as long as at most one agent on the other side does. For the case where two or more agents on both sides change preferences, we construct examples showing that both the lattice structure and polyhedral integrality fail—identifying this setting as a sharp threshold. We also present an XP-time algorithm for the general case, which implies a polynomial-time algorithm when the number of agents with changing preferences is constant. While these results establish the tractability of these regimes, closing the complexity gap in the fully general setting remains an interesting open question.

KEYWORDS

Stable matching; Robust solutions; Finite distributive lattice; Birkhoff's Representation Theorem

ACM Reference Format:
Rohith Reddy Gangam, Tung Mai, Nitya Raju, and Vijay V. Vazirani. 2026. Stable Matching: Dealing with Changes in Preferences. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/WMAU9924

1 INTRODUCTION

Matchings under preferences form a fundamental area of research in algorithmic game theory, with the seminal 1962 paper by Gale and Shapley [18] launching a rich line of work on stable matchings. Over the decades, this field has revealed elegant structural, algorithmic, and strategic properties, leading to real-world applications ranging from labor markets to school choice [15]. Unlike the setting defined in [18], we note that in practice, preferences of agents are often not static. They may change as agents gain more information, or because of external factors such as agents entering or leaving the market. Agents may also collude with others in hopes of securing better matches. Such considerations motivate the study of matchings that remain stable under preference changes.

Let instance A represent the original preferences of agents and instance B capture the changed preferences. We are interested in matchings that are stable in both A and B, called *robust stable matchings*. A recent thread of work considered the case where A and B are "nearby" instances— those differing only in the preference list of a *single agent*. Mai and Vazirani [29] and Gangam et al. [19] characterized the structure of stable matchings under such perturbations and gave efficient algorithms for identifying matchings that are stable under both instances. For this case, they showed that the intersection of the sets of the stable matchings, $\mathcal{M}_A \cap \mathcal{M}_B$, is a sublattice of the lattices of the two instances, $\mathcal{L}_A$ and $\mathcal{L}_B$. They also provided efficient algorithms to find and enumerate these matchings.

Building on [29], Chen et al. [12] introduced a model of robustness based on swap distance. A matching is defined to be d-robust if it remains stable under any d swaps in the preference lists, even over multiple agents. They gave an efficient algorithm for deciding the existence of such matchings by exploiting the structure of the stable matching lattice.

In this work, we allow multiple agents to make arbitrary changes and study structural as well as algorithmic issues. Interestingly enough, the threshold at which structural properties break is very sharp, and we identify it.

1.1 Our contributions

From our perspective, the following is a central question. Its importance stems from the fact that the stable matching lattice is *universal* in the class of finite distributive lattices in the following sense: Given any lattice $\mathcal{L}$ from this class, there exists a stable matching instance, A, of an appropriate size, such that its lattice $\mathcal{L}_A$ is isomorphic to $\mathcal{L}$. Consequently, an affirmative answer to the question below would imply a fundamental new property for the class of finite distributive lattices.

Question 1. *Is $\mathcal{M}_A \cap \mathcal{M}_B$ always a sublattice of $\mathcal{L}_A$ and $\mathcal{L}_B$ under arbitrary preference changes?*

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/WMAU9924

In Theorem 6 we show that, in general, the answer is "no". This leads to the natural question of identifying the cases in which this property does hold, and whether we can exploit it to obtain efficient algorithms.

Consider a robust stable matching instance (A, B) in which p of the n workers and q of the n firms change their preferences in going from A to B. We show that the above statement fails when $(p, q) = (2, 2)$.

In addition to structural questions, we study computational and geometric aspects of robust stable matchings:

- Is the decision problem $\mathcal{M}_A \cap \mathcal{M}_B \neq \emptyset$ solvable in polynomial time?
- Can the worker-optimal and firm-optimal matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ be efficiently computed?
- Does the set $\mathcal{M}_A \cap \mathcal{M}_B$ admit an efficient Birkhoff-partial order and support efficient enumeration?
- Is the intersection of stable matching polytopes integral, enabling linear programming (LP) based algorithms?

Our results, summarized in Figure 1, characterize the structural and algorithmic behavior of $\mathcal{M}_A \cap \mathcal{M}_B$ as a function of p and q, the number of workers and firms changing preferences. When $(p, q) = (n, 1)$ or $(1, n)$—that is, any number of agents on one side and at most one on the other—the intersection remains a sublattice, the robust stable matching polytope remains integral, and both the worker-optimal and firm-optimal matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ can be computed efficiently via combinatorial algorithms. In addition, LP-based methods can be used to find a robust stable matching in these cases. When changes are restricted to one side entirely, i.e., $(p, q) = (0, n)$ or $(n, 0)$, we also obtain a succinct Birkhoff representation that supports enumeration with polynomial delay. While the partial order used for enumeration is efficiently computable in these one-sided cases, its computation *remains open* for the two-sided $(1, n)$ or $(n, 1)$ settings. Here, we show that it can be computed in polynomial time for $(1, n)$ if and only if it can be for $(1, 1)$.

In contrast, we identify a sharp structural threshold at $(p, q) = (2, 2)$: beyond this point, the intersection $\mathcal{M}_A \cap \mathcal{M}_B$ may no longer form a sublattice, and the integrality of the associated polytope can fail. For general (p, q) instances, we present an $O(n^{p+q+2})$-time XP algorithm that decides whether a robust stable matching exists, constructs one if it does, and enumerates all such matchings with the same asymptotic delay. This yields a polynomial-time algorithm and enumeration when the number of agents with differing preferences is bounded by a constant k (i.e., $p + q = k$). Determining the precise threshold for the computational complexity in the general case remains an intriguing open question.

(p, q)	**Computation (P?)**	**Structure (Lattice?)**	**Geometry (Integral?)**
(0,1)	P [19]	Yes [19]	Yes [Thm. 12]
(0,n)	P [Thm. 11]	Yes [Thm. 9]	Yes [Thm. 12]
(1,1)	P [Thm. 11]	Yes [Thm. 7]	Yes [Thm. 12]
(1,n)	P [Thm. 11]	Yes [Thm. 7]	Yes [Thm. 12]
(2,2)	P [Thm. 15]	No [Thm. 6]	No [Thm. 13]
(p,q)	$O(n^{p+q+2})$ [Thm. 15]	-	-
(n,n)	NP-Complete [33]	No [Thm. 6]	No [Thm. 13]

Figure 1: Summary of our results on robust stable matchings. p workers and q firms permute their preference lists from instance A to B. The "Computation" column deals with the decision problem $\mathcal{M}_A \cap \mathcal{M}_B \neq \emptyset$. The "Structure" column answers if $\mathcal{M}_A \cap \mathcal{M}_B$ is a sublattice of $\mathcal{L}_A$ and $\mathcal{L}_B$. The "Geometry" column answers if the robust fractional stable matching polytope is integral.

2 RELATED WORK

The stable matching problem was introduced by Gale and Shapley [18], who also proposed the celebrated Deferred Acceptance (DA) algorithm. The DA algorithm computes a stable matching that is optimal for one side of the market and pessimal for the other. The set of stable matchings forms a distributive lattice, as shown in [26], and this structure has played a central role in many algorithmic developments. Key game-theoretic properties such as the Rural Hospitals Theorem [34] and incentive compatibility of the DA algorithm [14] have also been established. The integrality of the stable matching polytope was shown by Teo and Sethuraman [39], with alternative formulations given in [35, 40]. For comprehensive treatments of these topics, we refer the reader to [24, 27, 30].

Mai and Vazirani [29] introduced the notion of robustness used in this paper, defining a matching to be robust if it remains stable under both the original and a perturbed instance. They provided polynomial-time algorithms for the case where a single agent modifies preferences via a simple downshift. This was generalized by Gangam et al. [19] to arbitrary changes by one agent. Chen et al. [12] introduced the notion of d-robust stable matchings, where up to d swaps are allowed in the preference profiles, and also studied the trade-offs between social welfare and stability. Miyazaki and Okamoto [33] have shown that when all agents are allowed to change preferences between instances, finding a robust stable matching—which they term a jointly stable matching—is NP-hard.

Building on this work, we consider the case where multiple agents on both sides simultaneously change their preferences via arbitrary permutations. While earlier approaches were largely based on lattice structure, rotation posets, or variants of the DA algorithm, our approach also incorporates linear programming (LP) techniques into the analysis. When only one side of the market changes preferences, the setting is closely related to stable matching with strict preferences on one side and weak or partial preferences on the other (see details in the full version of our paper at [20]). In such settings, several refined notions of stability have been studied, including weak, strong, and super-strong stable matchings. Irving [25] showed that weakly stable matchings always exist and can be computed in polynomial time. Spieker [37] showed that the set of super-stable matchings forms a distributive lattice. Manlove [31] extended this to strong stability, showing that the corresponding set of matchings also retains a lattice structure. Kunysz et al. [28] gave an efficient characterization of strongly stable matchings via succinct partial orders.

Aziz et al. [1, 2] studied uncertain preferences and proposed robust solutions that perform well across all completions of partial orders. Genc et al. [22, 23] proposed (x, y)-supermatches, which allow re-stabilizing the matching after any x agents break up by rematching them while affecting at most y other pairs. While not robust in our sense, they provide a notion of ease of repair.

Chen et al. [11] introduced stable matching under multi-modal preferences, where each agent evaluates potential matches using multiple criteria. Menon and Larson [32] addressed the problem of minimizing the maximum number of blocking pairs across all completions of weak orders. Incremental changes to preferences over time have also been studied. Bredereck et al. [7] and Boehmer et al. [5, 6] explored the complexity of adapting stable matchings as inputs evolve, providing structural results and efficient algorithms in restricted settings.

Popularity is a voting-based relaxation of stability first introduced by Gärdenfors [21]. A matching is popular if it does not lose a head-to-head election against any other matching. Popular matchings have been extensively studied. Recently, Bullinger et al. [9] introduced the notion of robust popular matchings and gave polynomial-time and hardness results based on the extent of changes in preferences. Csáji [13] studied robust popular matchings in the presence of multi-modal and uncertain preferences.

Robustness to input perturbations has also been studied in voting theory [8, 17, 36], where it is often interpreted through the lens of swap-bribery. The cost of altering votes is typically measured via the swap distance [16]. Similar ideas have been applied to stable matchings [4], where preference manipulations are used to enforce or prevent certain matches. Bérczi et al. [10] study how one can deliberately change preferences to enforce the existence of stable matchings with desired properties.

3 PRELIMINARIES

3.1 The stable matching problem and the lattice of stable matchings

A stable matching problem instance consists of a set of n workers, $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$, and a set of n firms, $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$, collectively referred to as agents. Each agent $a \in \mathcal{W} \cup \mathcal{F}$ has a preference profile $>_a$, which is a strict total order over the agents of the opposite type: $w_i >_f w_j$ indicates that firm f strictly prefers worker w_i to w_j. Worker preferences are expressed analogously.

A matching M is a one-to-one correspondence between $\mathcal{W}$ and $\mathcal{F}$. For each pair $(w, f) \in M$, w is called the partner of f in M (or M-partner), and vice versa. A pair $(w, f) \notin M$ is a *blocking pair* if $f >_w M(w)$ and $w >_f M(f)$; that is, if both strictly prefer each other to their respective partners in M. A matching is *stable* under the instance if it admits no blocking pairs.

Let M and M' be two stable matchings. We say that M *dominates* M', denoted $M \preceq M'$, if every worker weakly prefers their partner in M to their partner in M'. In this case, M is also called a *predecessor* of M'. A stable matching M is a *common predecessor* of two stable matchings M_1 and M_2 if it is a predecessor of both. It is a *lowest common predecessor* of M_1 and M_2 if no other common predecessor M' satisfies $M \preceq M'$. Analogously, one can define the notions of *successor* and *highest common successor*.

This dominance partial order has the following key property: for any two stable matchings M_1 and M_2, their lowest common predecessor and highest common successor are unique. That is, the set of stable matchings forms a *lattice* under this partial order. The lowest common predecessor is called the *meet*, denoted $M_1 \wedge M_2$, and the highest common successor is called the *join*, denoted $M_1 \vee M_2$. One can show that $M_1 \wedge M_2$ is the matching obtained when each worker chooses their more preferred partner from M_1 and M_2; it is easy to verify that this matching is also stable. Interestingly, $M_1 \wedge M_2$ also results when each firm chooses their less preferred partner from M_1 and M_2. Similarly, $M_1 \vee M_2$ is the matching in which each worker (respectively, firm) chooses their less (respectively, more) preferred partner from M_1 and M_2, and this matching is also stable. Moreover, the lattice operations *join* and *meet* distribute: given three stable matchings M, M', M'',

$$M \vee (M' \wedge M'') = (M \vee M') \wedge (M \vee M'')$$
$$M \wedge (M' \vee M'') = (M \wedge M') \vee (M \wedge M'').$$

The lattice of stable matchings contains a unique matching $M_\top$ that dominates all others, and a unique matching $M_\bot$ that is dominated by all others. $M_\top$ is called the *worker-optimal stable matching*, as each worker is matched to their most preferred firm among all stable matchings. This is also the *firm-pessimal stable matching*. Similarly, $M_\bot$ is the *worker-pessimal* and *firm-optimal stable matching*. Since the number of stable matchings under an instance is finite, the stable matching lattice is a *finite distributive lattice*.

3.2 Birkhoff's Theorem, sublattices and compressions

Definition 1. *A closed set of a partially ordered set (poset) is a subset S where if an element is in S, all of its predecessors are in S.*

The collection of closed sets (also called lower sets) of a partial order Π is closed under union and intersection, and forms a distributive lattice, with join and meet corresponding to these two operations, respectively. Denote this lattice $L(\Pi)$. Birkhoff's theorem [3], also known as the *fundamental theorem for finite distributive lattices* (e.g., see [38]), states that for any finite distributive lattice $\mathcal{L}$, there exists a partial order Π such that $L(\Pi) \cong \mathcal{L}$, i.e., the lattice of closed sets of Π is isomorphic to $\mathcal{L}$. We say that Π *generates* $\mathcal{L}$.

Theorem 2 (Birkhoff [3]). *Every finite distributive lattice $\mathcal{L}$ is isomorphic to $L(\Pi)$, for some finite poset Π.*

A *join semi-sublattice* $\mathcal{L}_j$ of a distributive lattice $\mathcal{L}$ is a subset such that, for any $x, y \in \mathcal{L}_j$, the join $x \vee y$ also belongs to $\mathcal{L}_j$. Similarly, a *meet semi-sublattice* $\mathcal{L}_m$ is a subset such that, for any $x, y \in \mathcal{L}_m$, the meet $x \wedge y$ also belongs to $\mathcal{L}_m$. A *sublattice* $\mathcal{L}'$ of $\mathcal{L}$ is a subset that is both a join and meet semi-sublattice. The *Hasse diagram* of a poset is a directed graph with a vertex for each element in the poset and an edge from x to y if and only if $x \prec y$ and there is no z such that $x \prec z \prec y$, i.e., all precedence relations implied by transitivity are suppressed.

Let $\mathcal{L}$ be the lattice generated by a poset Π, and let $H(\Pi)$ denote the Hasse diagram of Π. To derive a new poset Π', consider the following operations: choose a set E of directed edges and add them to $H(\Pi)$. Let H_E be the resulting graph. Let H' be the graph obtained by shrinking the strongly connected components of H_E. Define Π' to be the partial order induced by the non-shrunk edges of H' on its nodes. The poset Π' is called a *compression* of Π.

Theorem 3 ([19], Theorem 1). *There is a one-to-one correspondence between the compressions of* Π *and the sublattices of* $L(\Pi)$*. Moreover, if a sublattice* $\mathcal{L}' \subseteq L(\Pi)$ *corresponds to a compression* Π'*, then* $\mathcal{L}'$ *is generated by* Π'*.*

We say that the edge set E defines $\mathcal{L}'$. Additional details about compressions, rotations and the rotation poset appear in the full version of our paper at [20], though they are not needed for the main results of this paper.

3.3 Robust Stable Matchings

A robust stable matching instance consists of k stable matching instances $A_1, A_2, \dots, A_k$ with $k \geq 2$, each defined on the same set of $2n$ agents: n workers $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ and n firms $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$, where all preference lists are strict and complete. We say that the instances are of type (p, q) if there exists A_i such that for every $j \in \{1, 2, \dots, k\}$, the preferences in A_j differ from those in A_i for at most p workers and q firms, and remain the same for the other $2n - p - q$ agents. The instance A_i is referred to as the *original* instance, and the others are called the *changed* instances.

Definition 2. *Given a robust stable matching instance* $(A_1, \dots, A_k)$*, a matching is said to be* robust stable *under these instances if it is stable under each instance* A_i *for all* $i \in \{1, \dots, k\}$*.*

We focus primarily on the case where $k = 2$, and let (A, B) be a robust stable matching instance of type (p, q), defined on $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ and $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$. Here, A is the original instance and B is the changed instance. The results are symmetric in p and q. For any stable matching instance I, let $\mathcal{M}_I$ denote the set of all stable matchings and $\mathcal{L}_I$ the corresponding lattice of stable matchings. Given a pair (A, B) of type (p, q), our goal is to understand the structure and computational tractability of the robust stable matching set $\mathcal{M}_A \cap \mathcal{M}_B$, i.e., the matchings that are stable in both A and B. Specifically, we investigate whether $\mathcal{M}_A \cap \mathcal{M}_B$ forms a sublattice of $\mathcal{L}_A$ (and of $\mathcal{L}_B$), whether such a matching can be found efficiently, and whether linear programming techniques can be leveraged to compute one.

3.4 Lattices of nearby instances

Consider two instances A and B that are of type $(0, 1)$ or $(1, 0)$. As the number of matchings stable under an instance can be exponential in n, checking the stability of every matching in $\mathcal{M}_A$ with respect to B may require exponential time. The results of [19], which reveal the structure of these robust stable matchings, are therefore essential and form the foundation for our work. We recall them below (statements are modified to match our notation).

Theorem 4 ([19], Theorem 29). *If a lattice* $\mathcal{L}$ *can be partitioned into a sublattice* $\mathcal{L}_1$ *and a semi-sublattice* $\mathcal{L}_2$*, and there is a polynomial-time oracle that determines whether any* $x \in \mathcal{L}$ *belongs to* $\mathcal{L}_1$ *or* $\mathcal{L}_2$*, then an edge set defining* $\mathcal{L}_1$ *can be found in polynomial time.*

Lemma 1 ([19]). *If instances A and B are of type* $(0, 1)$*, then:*

1. $\mathcal{L}' = \mathcal{M}_A \cap \mathcal{M}_B$ *is a sublattice of both* $\mathcal{L}_A$ *and* $\mathcal{L}_B$*.*
2. $\mathcal{M}_A \setminus \mathcal{M}_B$ *is a join (or meet) semi-sublattice of* $\mathcal{L}_A$*.*
3. *A set of edges defining the sublattice* $\mathcal{L}' = \mathcal{M}_A \cap \mathcal{M}_B$ *can be computed in polynomial time.*

These results can be extended to multiple instances in the following way. Let $A, B_1, B_2, \dots, B_k$ be such that each instance B_i is formed from A by arbitrarily permuting the preferences of one agent (possibly a different agent for each B_i), i.e., for all $i \in \{1, 2, \dots, k\}$, the pair (A, B_i) is of type $(1, 0)$ or $(0, 1)$. Lemma 1 tells us that each $\mathcal{L}_i = \mathcal{M}_A \cap \mathcal{M}_{B_i}$ is a sublattice of $\mathcal{L}_A$. Let the edge set defining $\mathcal{L}_i$ be E_i, and define $\mathcal{L}' = \mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \mathcal{M}_{B_2} \cap \ldots \cap \mathcal{M}_{B_k}$. Then:

Lemma 2.
1. *([19])* $\mathcal{L}'$ *is a sublattice of* $\mathcal{L}_A$*.*
2. *([19])* $E = \bigcup_i E_i$ *defines* $\mathcal{L}'$*.*
3. *([19]) E, and hence the partial order* Π' *generating* L'*, can be computed in polynomial time.*
4. *([24]) Matchings in* $\mathcal{L}'$ *can be enumerated efficiently.*

3.5 Linear programming formulation

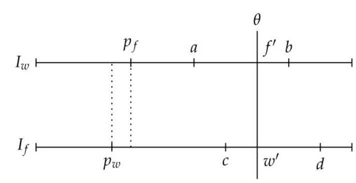

Figure 2: Finding integral stable matching from a fractional solution (Image from [15])

A well-known result about the stable matching problem is that it admits a linear programming formulation in which all optimal vertices are integral([39, 40]). This yields an efficient method to compute stable matchings. The LP formulation is given below as (1). The third constraint disallows blocking pairs: for any worker-firm pair (w, f), it ensures that if w is matched to someone less preferred than f, then f must be matched to someone more preferred than w. The remaining constraints ensure x is a fractional perfect matching.

$$
\begin{aligned}
\text{maximize} \quad & 0 \\
\text{subject to} \quad & \sum_{w} x_{wf} = 1 && \forall f \in \mathcal{F}, \\
& \sum_{f} x_{wf} = 1 && \forall w \in \mathcal{W}, \\
& \sum_{f >^A_w f'} x_{wf'} - \sum_{w' >^A_f w} x_{w'f} \leq 0 && \forall w \in \mathcal{W},\ \forall f \in \mathcal{F}, \\
& x_{wf} \geq 0 && \forall w \in \mathcal{W},\ \forall f \in \mathcal{F}.
\end{aligned} \tag{1}
$$

A solution x to LP 1 can be converted into an integral perfect matching as follows. Construct $2n$ unit intervals, each corresponding to an agent. For each worker w, divide its interval I_w into n subintervals of lengths x_{wf} (note that $\sum_f x_{wf} = 1$), ordered from w's most preferred firm to their least. Apply the same process to each firm f's interval I_f, but order the subintervals from f's least to most preferred worker. Choose a value θ uniformly at random from $[0, 1]$, and identify the subinterval in each agent's interval that contains θ. The probability that θ lies exactly on a boundary is zero, so we may ignore this case. Let $\mu_\theta : \mathcal{W} \rightarrow \mathcal{F}$ denote the firm corresponding to the subinterval containing θ in each worker's interval, and let $\mu'_\theta : \mathcal{F} \to \mathcal{W}$ denote the analogous mapping for firms. These functions define perfect matchings and are inverses of each other. Moreover, they are stable for any $\theta \in [0, 1]$.

Lemma 3 ([39]). *If $\mu_\theta(w) = f$ then $\mu'_\theta(f) = w$.*

Lemma 4 ([39]). *For each $\theta \in [0, 1]$, the matching μ_θ is stable.*

Theorem 5 ([40]). *The stable matching polytope defined by LP 1 has integral optimal vertices; that is, it is the convex hull of stable matchings.*

4 RESULTS

In this section, we present our results. We primarily consider the case of (p, q) robust stable matchings between instances (A, B) defined for workers $\mathcal{W} = \{w_1, \ldots, w_n\}$ and firms $\mathcal{F} = \{f_1, \ldots, f_n\}$. We assume $0 \leq p \leq q \leq n$, as the results are symmetric in p and q.

In Section 4.1, we investigate the lattice structure of robust stable matchings for various values of p and q. For cases where the lattice structure is preserved, Section 4.2 provides algorithms to compute worker-optimal and firm-optimal robust stable matchings. In Section 4.3, we use linear programming techniques to obtain robust stable matchings by characterizing when the corresponding polytope admits integral optimal vertices. Finally, in Section 4.4, we address the general setting and present an XP-time algorithm that decides the existence of a robust stable matching. We would like to note that, while the results focus primarily on two stable matching instances, they can all be extended to many instances(see Theorem 14). (Proofs of all theorems and lemmas marked † appear in the full version of our paper at [20].)

4.1 Lattice structure

The primary question we address is whether the set of robust stable matchings for (A, B), namely $\mathcal{M}_A \cap \mathcal{M}_B$, always forms a sublattice of both $\mathcal{L}_A$ and $\mathcal{L}_B$. While [19] showed that this set is a sublattice in the $(0, 1)$ case, their proof relies on the fact that $(\mathcal{M}_A \setminus \mathcal{M}_B)$ forms a semi-sublattice of $\mathcal{L}_A$. They prove that as long as changes are restricted to one side—i.e., when (A, B) is of type $(0, n)$ or $(n, 0)$—$\mathcal{M}_A \cap \mathcal{M}_B$ remains a sublattice of both $\mathcal{L}_A$ and $\mathcal{L}_B$ [19, Proposition 6].

A trial-and-error search for counterexamples when agents on both sides change preferences yielded surprising results. Often the sublattice property still held, as when only one side changes preferences, the join and meet operations in $\mathcal{L}_A$ and $\mathcal{L}_B$ coincide. When both sides change preferences, this alignment no longer holds. Surprisingly even in such cases, the set $\mathcal{M}_A \cap \mathcal{M}_B$ often remained a sublattice under both $\mathcal{L}_A$ and $\mathcal{L}_B$. Examples illustrating this behavior are shown in [20]. However, as Theorem 6 shows the intersection $\mathcal{M}_A \cap \mathcal{M}_B$ is not always a sublattice. An explicit example is provided in Figure 3. Thus, while the sublattice property holds for $(0, n)$, it no longer holds for $(2, 2)$.

Theorem 6. † *When (A, B) are of type (p, q) with $p, q \geq 2$, the set $\mathcal{M}_A \cap \mathcal{M}_B$ is not always a sublattice of $\mathcal{L}_A$ and $\mathcal{L}_B$.*

This raises the question: where does the sublattice property break down? We show that as long as no more than one agent changes preferences on one side, the sublattice property still holds. In particular, we establish that for $(1, n)$, the intersection $\mathcal{M}_A \cap \mathcal{M}_B$ remains a sublattice.

1	a	b	c	d
2	b	c	a	d
3	d	c	a	b
4	c	d	a	b

Worker preferences in A

a	2	4	1	3
b	1	2	3	4
c	3	4	2	1
d	4	3	2	1

Firm preferences in A

1	a	b	c	d
2	b	c	a	d
3	c	d	a	b
4	d	a	c	b

Worker preferences in B

a	2	4	1	3
b	1	2	3	4
c	4	2	3	1
d	3	4	1	2

Firm preferences in B

Figure 3: $\mathcal{M}_A \cap \mathcal{M}_B$ is not a sublattice of $\mathcal{L}_A$. Preference lists are in decreasing order. The agents changing preferences from A to B are in red. This notation is used for all examples.

Let A and B be a $(1, n)$ robust stable matching instance where only w_1 and all firms change their preferences from A to B.

Theorem 7. *If A and B are of type $(1, n)$, then $\mathcal{M}_A \cap \mathcal{M}_B$ is a sublattice of both $\mathcal{L}_A$ and $\mathcal{L}_B$.*

Proof. Without loss of generality, assume $|\mathcal{M}_A \cap \mathcal{M}_B| > 1$, and let M_1 and M_2 be two distinct matchings in $\mathcal{M}_A \cap \mathcal{M}_B$. Let $\vee_A$ and $\vee_B$ denote the join operations under A and B, respectively. Likewise, let $\wedge_A$ and $\wedge_B$ denote the meet operations under A and B. The join $M_1 \vee_A M_2$ is obtained by assigning each worker to their less preferred partner from M_1 and M_2, according to instance A. Since the preferences of workers $w_2, w_3, \ldots, w_n$ are identical in A and B, their less preferred partners under A and B coincide. Thus, for all $w \neq w_1$, we have:

$$(M_1 \vee_A M_2)(w) = (M_1 \vee_B M_2)(w).$$

Since $M_1 \vee_A M_2$ and $M_1 \vee_B M_2$ are both stable matchings under their respective instances, they are also perfect matchings. Hence, the remaining worker w_1 must be matched to the same firm in both matchings:

$$(M_1 \vee_A M_2)(w_1) = (M_1 \vee_B M_2)(w_1).$$

Therefore, $M_1 \vee_A M_2 = M_1 \vee_B M_2$. A similar argument shows that $M_1 \wedge_A M_2 = M_1 \wedge_B M_2$. Hence, the join and meet operations under instances A and B are equivalent, and both $M_1 \vee_A M_2$ and $M_1 \wedge_A M_2$ belong to $\mathcal{M}_A \cap \mathcal{M}_B$. The theorem follows. □

Thus, when the instances are of type $(1, n)$, Theorem 3 guarantees the existence of a poset that generates the corresponding sublattice. However, as Lemma 5 states, this is no longer true for the $(0, n)$ case - Figure 4 provides an example where A and B are $(0, n)$, but $\mathcal{M}_A \setminus \mathcal{M}_B$ is not a semi-sublattice of $\mathcal{L}_A$.

Lemma 5. † *If A and B are of type $(0, n)$, then $\mathcal{M}_A \setminus \mathcal{M}_B$ is not always a semi-sublattice of $\mathcal{L}_A$.*

This shows that Theorem 4 cannot be used directly to find Birkhoff's partial order in the $(0, n)$ setting. To circumvent this, we define n hybrid instances $B_1, \ldots, B_n$ such that for each $i \in$

1	a	b	d	c	e
2	b	a	c	d	e
3	c	d	b	a	e
4	b	d	c	a	e
5	c	e	a	b	d

Worker preferences in A and B

a	4	2	1	3	5
b	3	1	2	4	5
c	2	4	3	1	5
d	1	3	4	2	5
e	5	1	2	3	4

Firm preferences in A

a	4	2	1	3	5
b	1	4	2	3	5
c	4	5	3	1	2
d	1	3	4	2	5
e	5	1	2	3	4

Firm preferences in B

Figure 4: $\mathcal{M}_A \setminus \mathcal{M}_B$ is not a semi-sublattice of $\mathcal{L}_A$.

$\{1, 2, \ldots, n\}$, the pair (A, B_i) is of type $(0, 1)$. Specifically, the preference profile of agent x in instance B_i is defined as:

$$>_x^{B_i} = \left\{ \begin{array}{ll} >_x^B, & \text{if } x = f_i \\ >_x^A, & \text{if } x \neq f_i \end{array} \right\}$$

That is, instance B_i is identical to instance A, except that the preferences of firm f_i are replaced by those in instance B.

Defined this way, Theorem 8 shows that the matchings stable under both A and B are precisely those stable under A and all the hybrid instances $B_1, B_2, \ldots, B_n$.

Theorem 8. † $\mathcal{M}_A \cap \mathcal{M}_B = \mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \mathcal{M}_{B_2} \cap \ldots \cap \mathcal{M}_{B_n}$.

Theorem 9. *If A and B are of type $(0, n)$, then Birkhoff's partial order generating $\mathcal{M}_A \cap \mathcal{M}_B$ can be computed efficiently, and the matchings in this set can also be enumerated with polynomial delay.*

Proof. Since each B_i differs from instance A at exactly one agent, we apply Lemma 1 to compute the edge sets E_i defining $\mathcal{M}_A \cap \mathcal{M}_{B_i}$. By Lemma 2, the union $E = \bigcup_i E_i$ defines the sublattice $\mathcal{M}_A \cap \mathcal{M}_B$. and can be used to enumerate all the matchings efficiently. □

A similar technique can be attempted to the case where (A, B) is of type $(1, n)$. Let $w_1 \in \mathcal{W}$ and $f_1, \ldots, f_n \in \mathcal{F}$ be the worker and firms whose preferences differ between A and B. For any subset $X \subseteq \mathcal{W} \cup \mathcal{F}$, let B_X denote the instance in which each agent in X has the same preferences as in B, and all other agents have the same preferences as in A. Then:

Theorem 10. † $\mathcal{M}_A \cap \mathcal{M}_B = \mathcal{M}_A \cap \mathcal{M}_{B_{\{w_1, f_1\}}} \cap \mathcal{M}_{B_{\{w_1, f_2\}}} \cap \cdots \cap \mathcal{M}_{B_{\{w_1, f_n\}}}$.

Corollary 1. † *Computing Birkhoff's partial order for the $(1, n)$ case is in $\mathcal{P}$ if and only if computing it for the $(1, 1)$ case is in $\mathcal{P}$.*

Unlike the $(0, 1)$ case, the set $\mathcal{M}_A \setminus \mathcal{M}_B$ is not always a semi-sublattice when (A, B) is of type $(1, 1)$; see the example in Figure 5. As a result, Theorem 4 cannot be applied, and characterizing the sublattice structure remains an open problem.

Lemma 6. † *If A and B are of type $(1, 1)$, then $\mathcal{M}_A \setminus \mathcal{M}_B$ is not always a semi-sublattice of $\mathcal{L}_A$.*

1	a	b	c	d	e
2	b	c	a	d	e
3	c	a	b	d	e
4	d	c	e	a	b
5	e	d	a	b	c

Worker preferences in A

a	2	3	1	4	5
b	3	1	2	4	5
c	1	2	3	4	5
d	5	3	4	1	2
e	4	5	1	2	3

Firm preferences in A

1	a	b	c	d	e
2	b	c	a	d	e
3	c	d	a	b	e
4	d	c	e	a	b
5	e	d	a	b	c

Worker preferences in B

a	2	3	1	4	5
b	3	1	2	4	5
c	1	4	3	2	5
d	5	3	4	1	2
e	4	5	1	2	3

Firm preferences in B

Figure 5: $\mathcal{M}_A \setminus \mathcal{M}_B$ is not a semi-sublattice of $\mathcal{L}_A$, even when (A, B) is of type $(1, 1)$.

WorkerOptimal(A, B):
Input: Stable matching instances A and B on agents $W \cup F$.
Output: Perfect matching M or ⊠ (when no robust stable matching exists).
Assume there are two rooms, $\mathcal{R}_A$ and $\mathcal{R}_B$ corresponding to instances A and B respectively. Each worker has a list, initialized to all firms, that they look at while proposing.

(1) **While** there is no rejection in any room or some worker is rejected by all firms, do:
  (a) **For** instances $I \in \{A, B\}, \forall w \in W$, w proposes to their best-in-I uncrossed firm in $\mathcal{R}_I$.
  (b) **For** instances $I \in \{A, B\}, \forall f \in F$, f tentatively accepts their best-in-I proposal in room $\mathcal{R}_I$ and rejects the rest.
  (c) **For** workers $w \in W$, if w is rejected by a firm f in ***any*** room, they cross f off their list.

(2)(a) **If** some worker is rejected by all firms, output ⊠.
  (b) **Else**, the acceptances define perfect matchings in each room. If they are the same in all room, **return** the perfect matching M.

Algorithm 1: Algorithm to find the worker-optimal stable matching. Workers maintain a single list across rooms. They may propose to different firms in each room but update their list synchronously based on rejections from any room.

4.2 Worker and firm optimal stable matchings

Theorem 7 states that if the instance (A, B) is of type $(1, n)$, then the set of robust stable matchings forms a sublattice of both $\mathcal{L}_A$ and $\mathcal{L}_B$. This sublattice structure allows us to define worker-optimal and firm-optimal robust stable matchings. In this section, we provide algorithms to compute these matchings. We note that for the $(0, n)$ case—i.e., when only one side changes preferences—some algorithms from the literature can be adapted to compute these

FirmOptimal(A, B):
Input: Stable matching instances A and B on agents $W \cup F$.
Output: Perfect matching M or ⊠ (when no robust stable matching exists).
Assume there are two rooms, $\mathcal{R}_A$ and $\mathcal{R}_B$ corresponding to instances A and B respectively. Each firm has a list, initialized to all workers, that they look at while proposing.

(1) **While** there is no rejection in any room or some firm is rejected by all firms, do:
  (a) **For** instances $I \in \{A, B\}, \forall f \in F, f$ proposes to their best-in-I uncrossed worker in $\mathcal{R}_I$.
  (b) **For** instances $I \in \{A, B\}, \forall w \in W, w$ tentatively accepts their best-in-I proposal (call it f_w^I) in room $\mathcal{R}_I$ and rejects the rest.
  (c) **For** instances $I \in \{A, B\}, \forall w \in W, w$ sends ***preemptive rejections*** to all firms f' not yet rejected, that are worse than their current option in $\mathcal{R}_I$, i.e., $f_w^I \geq_w^I f'$.
  (d) **For** firms $f \in F$, **if** f is rejected by a worker w in some room, they cross w off their list.

(2)(a) **If** some firm is rejected by all workers, output ⊠.
  (b) **Else**, the acceptances define perfect matchings in each room. If they are the same in all rooms, **return** the perfect matching M.

Algorithm 2: Algorithm to find the firm-optimal stable matching. Note that in step 1(c), workers may send rejections to firms that have not yet proposed to them.

optimal matchings. We discuss these adaptations in [20]. These do not extend to the $(1, n)$ setting, for which we design new algorithms based on Gale and Shapley's deferred acceptance algorithm [18].

The Deferred Acceptance Algorithm proceeds in iterations. In each iteration, (a) The proposing side (e.g., workers) proposes to their most preferred firm that has not yet rejected them. (b) Each firm tentatively accepts its most preferred proposal received in that round and rejects all others. (c) Each worker eliminates the firms that rejected them from their preference list. The process continues until a perfect matching is formed, at which point the algorithm outputs it as a stable matching. The key idea is that whenever a rejection occurs, the corresponding worker-firm pair can never be part of any stable matching. We modify this algorithm to compute worker- and firm-optimal stable matchings in the intersection $\mathcal{M}_A \cap \mathcal{M}_B$ for the $(1, n)$ setting.

Theorem 11. † *Let A and B be two stable matching instances that are of type $(1, n)$. Then Algorithm 1 and Algorithm 2 find the worker- and firm-optimal robust stable matchings, respectively, or correctly report that no such matching exists.*

Algorithm 1 runs the DA algorithm simultaneously across rooms, where the preference lists for a room corresponds to an individual instance. However the rejections apply *across all rooms*: if a worker is rejected by a firm in room $\mathcal{R}_A$, that firm is removed from their list in *all* rooms. This ensures if a perfect matching is returned, it is stable with respect to all instances. The primary challenge is the possibility that the algorithm produces different matchings in different rooms. We show that this cannot occur, a fact that follows from the intersection $\mathcal{M}_A \cap \mathcal{M}_B$ having the same partial order in both lattices. The firm-optimal Algorithm 2 proceeds analogously. A key distinction is that workers issue *preemptive* rejections: in each room, a worker rejects *all* firms ranked below their current tentative match based on their local preference list, ensuring the perfect matching is identical in all rooms.

4.3 Integrality of the Polytope

Section 3.5 presented an efficient algorithm for finding stable matchings using linear programming. We now extend this approach to define a linear program for robust stable matchings. Our goal is to characterize when the associated polytope is integral, allowing us to efficiently compute robust stable matchings. We show that the robust stable matching polytope is integral when (A, B) is of type either $(n, 1)$ or $(1, n)$, which in turn yields efficient algorithms for computing robust matchings in these settings. The linear program for the robust stable matching instance (A, B) is given below. The first two and the last constraints ensure that any feasible solution x is a fractional perfect matching, while the middle two constraints prevent blocking pairs under instances A and B, respectively.

$$\begin{aligned}
\text{maximize} \quad & 0 \\
\text{subject to} \quad & \sum_{w} x_{wf} = 1 && \forall f \in F, \\
& \sum_{f} x_{wf} = 1 && \forall w \in W, \\
& \sum_{f >_w^A f'} x_{wf'} - \sum_{w' >_f^A w} x_{w'f} \leq 0 && \forall w \in W,\ \forall f \in F, \\
& \sum_{f >_w^B f'} x_{wf'} - \sum_{w' >_f^B w} x_{w'f} \leq 0 && \forall w \in W,\ \forall f \in F, \\
& x_{wf} \geq 0 && \forall w \in W,\ \forall f \in F.
\end{aligned} \tag{2}$$

A solution to LP 2 corresponds to a fractional matching that is stable under both A and B. Analogous to the single-instance case (see Section 3.5), let $\theta \in [0, 1]$ be chosen uniformly at random. Construct two integral perfect matchings μ_θ^A and μ_θ^B by performing the interval-based rounding procedure using the preference orders from instances A and B, respectively.

Lemma 7. † *The matchings μ_θ^A and μ_θ^B are identical for every $\theta \in [0, 1]$, i.e., $\mu_\theta^A = \mu_\theta^B$.*

Theorem 12. † *If (A, B) is of type $(1, n)$ then μ_θ^A is stable under both A and B and robust fractional stable matching polytope has integer optimal vertices.*

Hence in the $(n, 1)$ and $(1, n)$ settings, LP 2 provides an efficient method to compute a robust stable matching. This technique fails for $(2, 2)$ instances—the proof of Lemma 7 does not extend, and, in fact, the polytope may not even be integral, see Figure 6.

Theorem 13. † *If (A, B) is of type $(2, 2)$ the robust stable matching polytope is not always integral.*

1	a	c	b	d
2	b	a	c	d
3	c	d	a	b
4	d	c	a	b

Worker preferences in A

a	2	1	3	4
b	1	2	3	4
c	4	1	3	2
d	3	1	4	2

Firm preferences in A

1	b	d	a	c
2	a	b	c	d
3	c	d	a	b
4	d	c	a	b

Worker preferences in B

a	1	2	3	4
b	2	1	3	4
c	4	1	3	2
d	3	1	4	2

Firm preferences in B

Figure 6: The robust stable matching polytope for this $(2, 2)$ instance is not integral.

The results from this section extend to more than two instances.

Theorem 14. † *For $(1, n)$ instances $(A_1, \ldots, A_k)$ with $k \geq 2$ where the same workers change preferences, the set $\mathcal{L}' = \mathcal{M}_{A_1} \cap \ldots \cap \mathcal{M}_{A_k}$ forms a sublattice of each $\mathcal{L}_{A_i}$. The LP formulation finds the matchings in $\mathcal{L}'$ and the worker-optimal and firm-optimal robust stable matchings can be found in polynomial time. If the instances are $(0, n)$, Birkhoff's partial order generating $\mathcal{L}'$ can be found efficiently and its matchings can be enumerated with polynomial delay.*

4.4 A Robust Stable Matching Algorithm for the General Case

Sections 4.1, 4.2, and 4.3 establish that finding robust stable matchings for $(1, n)$ instances lies in $\mathcal{P}$, while [33] shows that the problem is NP-hard when (A, B) is of type (n, n). Consider the general setting when A and B differ arbitrarily on the preferences of agents $S \subseteq W \cup F$, where $|S \cap W| = p$ and $|S \cap F| = q$. Algorithm 3 find a robust stable matching if one exists. By Theorem 15, the algorithm runs in time $O(n^{p+q+2})$ and can be modified to enumerate all robust stable matchings with the same asymptotic delay, yielding an XP-time algorithm parameterized by the number of agents whose preferences differ across the two instances. Intuitively all possible partner assignments for agents in S are enumerated, there are at most $O(n^{p+q})$ possibilities. Each one defines a partial matching M for the agents in S, and the algorithm attempts to determine whether M can be extended to a full matching that is stable under A and B. First it verifies that M does not contain any blocking pairs within $T = S \cup M(S)$. If no blocking pair exists, define truncated instance X on the remaining agents $U = (W \cup F) \setminus T$. In X, the preference lists of agents in U are shortened to eliminate any potential blocking pairs involving agents in T. A stable matching M' is computed on X. If M' is a perfect matching on U, it can be shown that the combined matching $M^\star = M \cup M'$ is a stable matching on the full set of agents in both A and B.

Theorem 15. † *For instances A and B of type (p, q). Algorithm 3 finds a robust stable matching in $O(n^{p+q+2})$ time, or correctly reports that no such matching exists. All the set of robust stable matchings can be enumerated with $O(n^{p+q+2})$ delay.*

RobustStableMatching(A, B):
Input: Stable matching instances A and B on agents $W \cup F$.
Output: Perfect matching $M^\star$ or ⊠ (when no robust stable matching exists).
$S \subseteq W \cup F$ are agents whose preferences differ in A and B.

(1) **For** each assignment of partners to agents in S that defines a valid partial matching:
 (a) Let M be the resulting partial matching and $T \coloneqq S \cup M(S)$.
 (b) **If** there exists $w \in T \cap W$ and $f \in T \cap F$ such that (w, f) is a blocking pair with respect to M in either A or B, **continue** to the next iteration.
 (c) Remove all agents in T to obtain the instance X on $U \coloneqq (W \cup F) \setminus T$;
 (d) For each $a \in T$ and $b \in U$, if $b \succ_a M(a)$ in either A or B, then truncate b's preference list at a in X (i.e., remove a and all agents ranked below a).
 (e) Find stable matching M' in X.
 (f) If M' is a perfect matching on U, **return** $M^\star \coloneqq M \cup M'$.
(2) **Return** ⊠.

Algorithm 3: An XP-time algorithm to find a robust stable matching for small (p, q).

Corollary 2. *If a constant number of agents change their preferences, then deciding whether a robust stable matching exists and finding one can be achieved in polynomial time. All such matchings can be enumerated with polynomial delay.*

Theorem 15 implies Corollary 2 as the running time is polynomial if a constant number of agents change preferences. The decision and enumeration problems for robust stable matchings can be solved efficiently. This shows that the set of robust stable matchings can be expressed as a union of $O(n^{p+q})$ stable matching lattices, one for each consistent assignment of partners to agents in S. However the partial orders defining these lattices can differ significantly. It remains unclear whether this decomposition yields an efficient method for computing Birkhoff's partial order even in the $(1, 1)$ case—and by Corollary 1, also in the $(1, n)$ case. These observations naturally lead to the following open problems.

Remark 16. Open Problem. The problem of computing Birkhoff's partial order for the $(1, 1)$ case and by extension, for $(n, 1)$ is open.

Remark 17. Open Problem. Given a (p, q) robust stable matching instance (A, B) with $2 \leq p, q \leq n$, $p + q < 2n$, and $p + q = \omega(1)$, it remains open whether determining the existence of a robust stable matching—and computing one if it exists—is in $\mathcal{P}$.

ACKNOWLEDGMENTS

We thank all the reviewers for their valuable comments. We are especially grateful to Simon Murray for his insights, which led to Algorithm 3.

REFERENCES

[1] Haris Aziz, Péter Biró, Tamás Fleiner, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. 2022. Stable matching with uncertain pairwise preferences. *Theoretical Computer Science* 909 (2022), 1–11. https://doi.org/10.1016/j.tcs.2022.01.028

[2] Haris Aziz, Péter Biró, Tamás Fleiner, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. 2022. Stable matching with uncertain pairwise preferences. *Theoretical Computer Science* 909 (2022), 1–11. https://doi.org/10.1016/j.tcs.2022.01.028

[3] Garrett Birkhoff. 1937. Rings of sets. *Duke Mathematical Journal* 3, 3 (1937), 443–454.

[4] Niclas Boehmer, Robert Bredereck, Klaus Heeger, and Rolf Niedermeier. 2021. Bribery and control in stable marriage. *Journal of Artificial Intelligence Research* 71 (2021), 993–1048.

[5] Niclas Boehmer, Klaus Heeger, and Rolf Niedermeier. 2022. Deepening the (Parameterized) Complexity Analysis of Incremental Stable Matching Problems. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 241)*, Stefan Szeider, Robert Ganian, and Alexandra Silva (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 21:1–21:15. https://doi.org/10.4230/LIPIcs.MFCS.2022.21

[6] Niclas Boehmer, Klaus Heeger, and Rolf Niedermeier. 2022. Theory of and Experiments on Minimally Invasive Stability Preservation in Changing Two-Sided Matching Markets. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 5 (Jun. 2022), 4851–4858. https://doi.org/10.1609/aaai.v36i5.20413

[7] Robert Bredereck, Jiehua Chen, Dusan Knop, Junjie Luo, and Rolf Niedermeier. 2020. Adapting Stable Matchings to Evolving Preferences. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 1830–1837. https://doi.org/10.1609/AAAI.V34I02.5550

[8] Robert Bredereck, Piotr Faliszewski, Andrzej Kaczmarczyk, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. 2021. Robustness among multiwinner voting rules. *Artificial Intelligence* 290 (2021), 103403. https://doi.org/10.1016/j.artint.2020.103403

[9] Martin Bullinger, Rohith Reddy Gangam, and Parnian Shahkar. 2024. Robust Popular Matchings. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 225–233.

[10] Kristóf Bérczi, Gergely Csáji, and Tamás Király. 2024. Manipulating the outcome of stable marriage and roommates problems. *Games and Economic Behavior* 147 (2024), 407–428. https://doi.org/10.1016/j.geb.2024.08.010

[11] Jiehua Chen, Rolf Niedermeier, and Piotr Skowron. 2018. Stable Marriage with Multi-Modal Preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation* (Ithaca, NY, USA) *(EC '18)*. Association for Computing Machinery, New York, NY, USA, 269–286. https://doi.org/10.1145/3219166.3219168

[12] Jiehua Chen, Piotr Skowron, and Manuel Sorge. 2021. Matchings under Preferences: Strength of Stability And Tradeoffs. *ACM Trans. Econ. Comput.* 9, 4, Article 20 (oct 2021), 55 pages. https://doi.org/10.1145/3485000

[13] Gergely Csáji. 2024. Popular and Dominant Matchings with Uncertain and Multimodal Preferences. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2740–2747. https://doi.org/10.24963/ijcai.2024/303 Main Track.

[14] Lester E Dubins and David A Freedman. 1981. Machiavelli and the Gale-Shapley algorithm. *The American Mathematical Monthly* 88, 7 (1981), 485–494.

[15] Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani (Eds.). 2023. *Online and Matching-Based Market Design*. Cambridge University Press.

[16] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. 2009. On distance rationalizability of some voting rules. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge* (California) *(TARK '09)*. Association for Computing Machinery, New York, NY, USA, 108–117. https://doi.org/10.1145/1562814.1562831

[17] Piotr Faliszewski and Jörg Rothe. 2016. *Control and Bribery in Voting*. Cambridge University Press, 146–168.

[18] David Gale and Lloyd S Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15.

[19] Rohith Reddy Gangam, Tung Mai, Nitya Raju, and Vijay V. Vazirani. 2022. A Structural and Algorithmic Study of Stable Matching Lattices of "Nearby" Instances, with Applications. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 250)*, Anuj Dawar and Venkatesan Guruswami (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 19:1–19:20. https://doi.org/10.4230/LIPIcs.FSTTCS.2022.19

[20] Rohith Reddy Gangam, Tung Mai, Nitya Raju, and Vijay V. Vazirani. 2025. Stable Matching: Dealing with Changes in Preferences. arXiv:2304.02590 [cs.DM] https://arxiv.org/abs/2304.02590

[21] Peter Gärdenfors. 1975. Match Making: Assignments based on bilateral preferences. *Behavioral Science* 20, 3 (1975), 166–173.

[22] Begum Genc, Mohamed Siala, Barry O'Sullivan, and Gilles Simonin. 2017. Finding Robust Solutions to Stable Marriage. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 631–637. https://doi.org/10.24963/ijcai.2017/88

[23] Begum Genc, Mohamed Siala, Gilles Simonin, and Barry O'Sullivan. 2017. On the complexity of robust stable marriage. In *International Conference on Combinatorial Optimization and Applications*. Springer, 441–448.

[24] Dan Gusfield and Robert W Irving. 1989. *The stable marriage problem: structure and algorithms*. MIT press.

[25] Robert W. Irving. 1994. Stable marriage and indifference. *Discrete Applied Mathematics* 48, 3 (1994), 261–272. https://doi.org/10.1016/0166-218X(92)00179-P

[26] Donald Ervin Knuth. 1976. Marriages stables. *Technical report* (1976).

[27] Donald Ervin Knuth. 1997. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*. Vol. 10. American Mathematical Soc.

[28] Adam Kunysz, Katarzyna E. Paluch, and Pratik Ghosal. 2016. Characterisation of Strongly Stable Matchings. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, Robert Krauthgamer (Ed.). SIAM, 107–119. https://doi.org/10.1137/1.9781611974331.CH8

[29] Tung Mai and Vijay V. Vazirani. 2018. Finding Stable Matchings that are Robust to Errors in the Input. In *European Symposium on Algorithms*.

[30] David Manlove. 2013. *Algorithmics of Matching Under Preferences*. World Scientific.

[31] David F. Manlove. 2002. The structure of stable marriage with indifference. *Discrete Applied Mathematics* 122, 1 (2002), 167–181. https://doi.org/10.1016/S0166-218X(01)00322-5

[32] Vijay Menon and Kate Larson. 2018. Robust and Approximately Stable Marriages Under Partial Information. In *Web and Internet Economics*, George Christodoulou and Tobias Harks (Eds.). Springer International Publishing, Cham, 341–355.

[33] Shuichi Miyazaki and Kazuya Okamoto. 2019. Jointly Stable Matchings. *J. Comb. Optim.* 38, 2 (2019), 646–665.

[34] Alvin E Roth. 1982. The economics of matching: Stability and incentives. *Mathematics of operations research* 7, 4 (1982), 617–628.

[35] A. E. Roth. 1985. The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory* 36, 2 (1985), 277–288.

[36] Dmitry Shiryaev, Lan Yu, and Edith Elkind. 2013. On elections with robust winners. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (St. Paul, MN, USA) *(AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 415–422.

[37] Boris Spieker. 1995. The set of super-stable marriages forms a distributive lattice. *Discrete Applied Mathematics* 58, 1 (1995), 79–84. https://doi.org/10.1016/0166-218X(94)00080-W

[38] Richard Stanley. 1996. Enumerative Combinatorics, vol. 1, Wadsworth and Brooks/Cole, Pacific Grove, CA, 1986; second printing.

[39] Chung-Piaw Teo and Jay Sethuraman. 1998. The Geometry of Fractional Stable Matchings and Its Applications. *Mathematics of Operations Research* 23, 4 (1998), 874–891. https://doi.org/10.1287/moor.23.4.874 arXiv:https://doi.org/10.1287/moor.23.4.874

[40] John H. Vande Vate. 1989. Linear programming brings marital bliss. *Operations Research Letters* 8, 3 (1989), 147–153. https://doi.org/10.1016/0167-6377(89)90041-2