A Structural and Algorithmic Study of Stable Matching Lattices of "Nearby" Instances, with Applications

Rohith Reddy Gangam ✉
University of California, Irvine, Irvine, CA, USA

Tung Mai ✉
Adobe Research, San Jose, CA, USA

Nitya Raju ✉
University of California, Irvine, Irvine, CA, USA

Vijay V. Vazirani ✉
University of California, Irvine, Irvine, CA, USA

Abstract

Recently [18] identified and initiated work on a new problem, namely understanding structural relationships between the lattices of solutions of two "nearby" instances of stable matching. They also gave an application of their work to finding a *robust stable matching*. However, the types of changes they allowed in going from instance A to B were very restricted, namely any one agent executes an *upward shift*.

In this paper, we allow any one agent to permute its preference list *arbitrarily*. Let M_A and M_B be the sets of stable matchings of the resulting pair of instances A and B, and let $\mathcal{L}_A$ and $\mathcal{L}_B$ be the corresponding lattices of stable matchings. We prove that the matchings in $M_A \cap M_B$ form a sublattice of both $\mathcal{L}_A$ and $\mathcal{L}_B$ and those in $M_A \setminus M_B$ form a join semi-sublattice. These properties enable us to obtain a polynomial time algorithm for not only finding a stable matching in $M_A \cap M_B$, but also for obtaining the partial order, as promised by Birkhoff's Representation Theorem [7]. As a result, we can generate all matchings in this sublattice.

Our algorithm also helps solve a version of the robust stable matching problem. We discuss another potential application, namely obtaining new insights into the incentive compatibility properties of the Gale-Shapley Deferred Acceptance Algorithm.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design

Keywords and phrases stable matching, robust solutions, finite distributive lattice, Birkhoff's Representation Theorem

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2022.30

Related Version *Full Version*: `https://arxiv.org/pdf/1804.05537.pdf`

Funding *Vijay V. Vazirani*: This work was supported in part by NSF grant CCF-1815901.

1 Introduction

The seminal 1962 paper of Gale and Shapley [14] introduced the stable matching problem and gave the Deferred Acceptance (DA) Algorithm for it. In the process, they initiated the field of matching-based market design. Over the years, numerous researchers unearthed the remarkably deep and pristine structural properties of this problem – this led to polynomial time algorithms for numerous problems, in particular those addressing various operations related to the lattice of stable matchings, see details below as well as in the books [17, 15, 20, 22, 12].

42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022).
Editors: Anuj Dawar and Venkatesan Guruswami; Article No. 30; pp. 30:1–30:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Recently [18] identified and initiated work on a new problem which appears to be fundamental and deserving of an in-depth study, namely understanding structural relationships between the lattices of solutions of two "nearby" instances. [18] had given an application of their work to finding a *robust stable matching* as described below. Let us say that two instance A and B of stable matching are *nearby instances* if B is obtained from A when one agent changes their preference list. Such pairs of instances arise naturally in an even more important context: the study of incentive compatibility of the DA Algorithm: one of the agents manipulates its preference list in order to get a better match. The types of manipulations allowed in [18] were very restricted, namely any one agent executes an *upward shift*, see definition below. They left the open problem of tackling more general changes.

[21] showed that finding a stable matching across k (≥ 2) arbitrary instances is NP-Hard. In this paper, we allow any one agent to permute its preference list *arbitrarily.* Let A and B be the resulting pair of instances, let M_A and M_B be the sets of their stable matchings and $\mathcal{L}_A$ and $\mathcal{L}_B$ be the corresponding lattices of stable matchings. We prove that the matchings in $M_A \cap M_B$ form a sublattice of both $\mathcal{L}_A$ and $\mathcal{L}_B$ and those in $M_A \setminus M_B$ form a join semi-sublattice, see definitions in Section 1.1. This enables us to obtain a polynomial time algorithm for not only finding a stable matching in $M_A \cap M_B$, but also to obtain the partial order, promised by Birkhoff's Representation Theorem [7], which helps generate all matchings in this sublattice. We also apply our algorithm to a more general setting for robust stable matching than the one given in [18].

The setting defined in [18] was the following: Let A be an instance of stable matching on n workers and n firms. A *domain of errors*, D, is defined via an operation called *upward shift*: For a firm f, assume its preference list in instance A is $\{\ldots, w_1, w_2, \ldots, w_k, w, \ldots\}$. Move up the position of worker w so f's list becomes $\{\ldots, w, w_1, w_2, \ldots, w_k, \ldots\}$. An analogous operation is defined on a worker w's list; again some firm f on its list is moved up. For each firm and each worker, consider all possible shifts to get the domain D; clearly, $|D| = \binom{2n}{1}\binom{n}{2} = O(n^3)$. Assume that *one error* is chosen from D via a given discrete probability distribution over D to obtain instance B. A *robust stable matching* is a matching that is stable for A and maximizes the probability of being stable for B. A polynomial time algorithm was given for finding such a matching.

Since we allow an *arbitrary permutation* to be applied to any *one worker or any one firm's* preference list, our domain of errors, say T, has size $2n(n!)$. Let $S \subseteq T$ and define a *fully robust stable matching w.r.t.* S to be a matching that is stable for A and for *each* of the $|S|$ instances obtained by introducing one error from S. We give an $O(|S|p(n))$ algorithm to determine if such a matching exists and if so to find one, where p is a polynomial function. In particular, if S is polynomial sized, then our algorithm runs in polynomial time. Clearly, this notion is weaker than the previous one, since we cannot extend it to the probabilistic setting; we leave that as an open problem, see Section 8.

In case all errors in S are on one side only, say the firms, it turns out that Algorithm D, which is a simple modification of the Deferred Acceptance Algorithm, works; this algorithm is given in Appendix D. However, extending this algorithm to the case that errors occur on both sides, workers and firms, results in an algorithm (Algorithm D) that has exponential runtime. Our polynomial time algorithm follows from a study of the sublattices of the lattice of stable matchings.

Conway, see [17], proved that the set of stable matchings of an instance forms a finite distributive lattice; see definitions in Section 2.2. Knuth [17] asked if every finite distributive lattice is isomorphic to the lattice arising from an instance of stable matching. A positive answer was provided by Blair [8]; for a much better proof, see [15]. A key fact about such lattices is Birkhoff's Representation Theorem [7], which has also been called *the fundamental theorem for finite distributive lattices*, e.g., see [23]. It states that corresponding to such a lattice, $\mathcal{L}$, there is a partial order, say Π, such that $\mathcal{L}$ is isomorphic to $L(\Pi)$, the lattice of closed sets of Π (see Section 2.2 for details). We will say that Π *generates* $\mathcal{L}$.

The following important question arose in the design of our algorithm: For a specified sublattice $\mathcal{L}'$ of $\mathcal{L}$, obtain partial order Π' from Π such that Π' generates $\mathcal{L}'$. Our answer to this question requires a study of Birkhoff's Theorem from this angle; we are not aware of any previous application of Birkhoff's Theorem in this manner. We define a set of operations called compressions; when a compression is applied to a partial order Π, it yields a partial order Π' on (weakly) fewer elements. The following implication of Birkhoff's Theorem is useful for our purposes:

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

A proof of Theorem 1, using stable matching lattices, is given in Section B for completeness. In the case of stable matchings, Π can be defined using the notion of *rotations*; see Section 2.2 for a formal definition. Since the total number of rotations of a stable matching instance is at most $O(n^2)$, Π has a succinct description even though $\mathcal{L}$ may be exponentially large. Our main algorithmic result is:

▶ **Theorem 2.** *There is an algorithm for checking if there is a fully robust stable matching w.r.t. any set* $S \subseteq T$ *in time* $O(|S|p(n))$*, where* p *is a polynomial function. Moreover, if the answer is yes, the set of all such matchings forms a sublattice of* $\mathcal{L}$ *and our algorithm finds a partial order that generates it.*

The importance of the stable matching problem lies not only in its efficient computability but also its good incentive compatibility properties. In particular, Dubins and Freedman [11] proved that the DA Algorithm is *dominant-strategy incentive compatible (DSIC)* for the proposing side. This opened up the use of this algorithm in a host of highly consequential applications, e.g., matching students to public schools in big cities, such as NYC and Boston, see [3, 1, 2]. In this application, the proposing side is taken to be the students; clearly, their best strategy is to report preference lists truthfully and not waste time and effort on "gaming" the system. In Section 8 we give a hypothetical situation regarding incentive compatibility in which Theorem 2 plays a role.

1.1 Overview of structural and algorithmic ideas

We start by giving a short overview of the structural facts proven in [18]. Let A and B be two instances of stable matching over n workers and n firms, with sets of stable matchings $\mathcal{M}_A$ and $\mathcal{M}_B$, and lattices $\mathcal{L}_A$ and $\mathcal{L}_B$, respectively. Let Π be the poset on rotations such that $L(\Pi) = \mathcal{L}_A$; in particular, for a closed set S, let $M(S)$ denote the stable matching corresponding to S. It is easy to see that if B is obtained from A by changing (upshifts only) the lists of only one side, either workers or firms, but not both, then the matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ form a sublattice of each of the two lattices (Proposition 6). Furthermore, if B is obtained by applying a shift operation, then $\mathcal{M}_{A\setminus B} = \mathcal{M}_A \setminus \mathcal{M}_B$ is also a sublattice of $\mathcal{L}_A$. Additionally, there is at most one rotation, ρ_{in}, that leads from $\mathcal{M}_A \cap \mathcal{M}_B$ to $\mathcal{M}_{A\setminus B}$ and at most one rotation, ρ_{out}, that leads from $\mathcal{M}_{A\setminus B}$ to $\mathcal{M}_A \cap \mathcal{M}_B$; moreover, these rotations can be found in polynomial time. Finally, for a closed set S of Π, $M(S)$ is stable for instance B

iff $\rho_{\text{in}} \in S \;\Rightarrow\; \rho_{\text{out}} \in S$.

With a view to extending the results of [18], we consider the following abstract question. Suppose instance B is such that $\mathcal{M}_A \cap \mathcal{M}_B$ and $\mathcal{M}_{A \setminus B}$ are both sublattices of $\mathcal{L}_A$, i.e., $\mathcal{M}_A$ is partitioned into two sublattices. Then, is there a polynomial time algorithm for finding a matching in $\mathcal{M}_A \cap \mathcal{M}_B$? Our answer to this question is built on the following structural fact: There exists a sequence of rotations $r_0, r_1, \ldots, r_{2k}, r_{2k+1}$ such that a closed set of Π generates a matching in $\mathcal{M}_A \cap \mathcal{M}_B$ iff it contains r_{2i} but not r_{2i+1} for some $0 \le i \le k$ (Proposition 19). Furthermore, this sequence of rotations can be found in polynomial time (see Section 4). Our generalization of Birkhoff's Theorem described in the Introduction is an important ingredient in this algorithm. At this point, we do not know of any concrete error pattern, beyond shift, for which this abstract setting applies.

Next, we address the case that $\mathcal{M}_{A \setminus B}$ is not a sublattice of $\mathcal{L}_A$. We start by proving that if B is obtained by permuting the preference list of any one worker, then $\mathcal{M}_{A \setminus B}$ must be a join semi-sublattice of $\mathcal{L}_A$ (Lemma 31); an analogous statement holds if the preference list of any one firm is permuted. Hence we study a second abstract question, namely lattice $\mathcal{L}_A$ is partitioned into a sublattice and a join semi-sublattice (see Section 5). These two abstract questions are called **Setting I and Setting II**, respectively, in this paper.

For Setting II, we characterize a compression that yields a partial order Π', such that Π' generates the sublattice consisting of matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ (Theorem 20). We also characterize closed sets of Π such that the corresponding matchings lie in this sublattice; however, the characterization is too elaborate to summarize succinctly (see Proposition 25). Edges forming the required compression can be found in polynomial time (Theorem 29), hence leading to an efficient algorithm for finding a matching in $\mathcal{M}_A \cap \mathcal{M}_B$.

Finally, consider the setting given in the Introduction, with T being the super-exponential set of all possible errors that can be introduced in instance A and $S \subset T$. We show that the set of all matchings that are stable for A and for each of the instances obtained by introducing one error from S forms a sublattice of $\mathcal{L}$ and we obtain a compression of Π that generates this sublattice (Section 7.2). Each matching in this sublattice is a fully robust stable matching. Furthermore, given a weight function on all worker-firm pairs, we can obtain, using the algorithm of [19], a maximum (or minimum) weight fully robust stable matching.

2 Preliminaries

2.1 The stable matching problem and the lattice of stable matchings

The stable matching problem takes as input a set of workers $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$ and a set of firms $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$; each agent has a complete preference ranking over the set of opposite side. A matching M is a one-to-one correspondence between $\mathcal{W}$ and $\mathcal{F}$. For each pair $wf \in M$, w is called the partner of f in M (or M-partner) and vice versa. For a matching M, a pair $wf \notin M$ is said to be *blocking* if they prefer each other to their partners. A matching M is *stable* if there is no blocking pair for M.

Let M and M' be two stable matchings. We say that M *dominates* M', denoted by $M \preceq M'$, if every worker weakly prefers his partner in M to M'. Define the relation *predecessor* as the transitive closure of dominates. The set of stable matchings forms a finite distributive lattice under the above definition of predecessor. The lattice contains a matching, M_0, that dominates all others and a matching M_z that is dominated by all others. M_0 is called the *worker-optimal matching*, since in it, each worker is matched to his most favorite firm among all stable matchings. Similarly, M_z is *firm-optimal matching.*

2.2 Birkhoff's Theorem and rotations

It is easy to see that the family of closed sets (also called lower sets, Definition 5) of a partial order, say Π, is closed under union and intersection and forms a distributive lattice, with join and meet being these two operations, respectively; let us denote it by $L(\Pi)$. Birkhoff's theorem [7], states that corresponding to any finite distributed lattice, $\mathcal{L}$, there is a partial order, say Π, whose lattice of closed sets $L(\Pi)$ is isomorphic to $\mathcal{L}$, i.e., $\mathcal{L} \cong L(\Pi)$. We will say that Π *generates* $\mathcal{L}$.

One way to define the partial orders generating stable matching lattices is using the concept of *rotation*. For a worker w let $s_M(w)$ denote the first firm f on w's list such that f strictly prefers w to her M-partner. Let $next_M(w)$ denote the partner in M of firm $s_M(w)$. A *rotation* ρ *exposed* in M is an ordered list of pairs $\{w_0f_0, w_1f_1, \ldots, w_{r-1}f_{r-1}\}$ such that for each i, $0 \le i \le r-1$, w_{i+1} is $next_M(w_i)$, where $i+1$ is taken modulo r. M/ρ is defined to be a matching in which each worker not in a pair of ρ stays matched to the same firm and each worker w_i in ρ is matched to $f_{i+1} = s_M(w_i)$. It can be proven that M/ρ is also a stable matching. The transformation from M to M/ρ is called the *elimination* of ρ from M.

Let $\rho = \{w_0f_0, w_1f_1, \ldots, w_{r-1}f_{r-1}\}$ be a rotation. For $0 \le i \le r-1$, we say that ρ *moves* w_i *from* f_i *to* f_{i+1}, and *moves* f_i *from* w_i *to* w_{i-1}. If f is either f_i or is strictly between f_i and f_{i+1} in w_i's list, then we say that ρ *moves* w_i *below* f. Similarly, ρ *moves* f_i *above* w if w is w_i or between w_i and w_{i-1} in f_i's list.

2.3 The rotation poset

A rotation ρ' is said to *precede* another rotation ρ, denoted by $\rho' \prec \rho$, if ρ' is eliminated in every sequence of eliminations from M_0 to a stable matching in which ρ is exposed. Thus, the set of rotations forms a partial order via this precedence relationship. The partial order on rotations is called *rotation poset* and denoted by Π.

▶ **Lemma 3** ([15], Lemma 3.2.1)**.** *For any worker w and firm f, there is at most one rotation that moves w to f, w below f, or f above w. Moreover, if ρ_1 moves w to f and ρ_2 moves w from f then $\rho_1 \prec \rho_2$.*

▶ **Lemma 4** ([15], Lemma 3.3.2)**.** Π *contains at most $O(n^2)$ rotations and can be computed in polynomial time.*

▶ **Definition 5.** *A* closed set *of a poset is a set S of elements of the poset such that if an element is in S then all of its predecessors are also in S.*

There is a one-to-one relationship between the stable matchings and the closed subsets of Π. Given a closed set S, the corresponding matching M is found by eliminating the rotations starting from M_0 according to the topological ordering of the elements in the set S. We say that S *generates* M.

Let S be a subset of the elements of a poset, and let v be an element in S. We say that v is a *minimal* element in S if there are no predecessors of v in S. Similarly, v is a *maximal* element in S if it has no successors in S. 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 $x \prec y$ and there is no z such that $x \prec z \prec y$. In other words, all precedences implied by transitivity are suppressed.

2.4 Sublattice and semi-sublattice

A *sublattice* $\mathcal{L}'$ of a distributive lattice $\mathcal{L}$ is subset of $\mathcal{L}$ such that for any two elements $x, y \in \mathcal{L}$, $x \vee y \in \mathcal{L}'$ and $x \wedge y \in \mathcal{L}'$ whenever $x, y \in \mathcal{L}'$, where $\vee$ and $\wedge$ are the join and meet operations of lattice $\mathcal{L}$. A *join semi-sublattice* $\mathcal{L}'$ of a distributive lattice $\mathcal{L}$ is subset of $\mathcal{L}$ such that for any two elements $x, y \in \mathcal{L}$, $x \vee y \in \mathcal{L}'$ whenever $x, y \in \mathcal{L}'$. Similarly, *meet semi-sublattice* $\mathcal{L}'$ of a distributive lattice $\mathcal{L}$ is subset of $\mathcal{L}$ such that for any two elements $x, y \in \mathcal{L}$, $x \wedge y \in \mathcal{L}'$ whenever $x, y \in \mathcal{L}'$. Note that $\mathcal{L}'$ is a sublattice of $\mathcal{L}$ iff $\mathcal{L}'$ is both join and meet semi-sublattice of $\mathcal{L}$.

▶ **Proposition 6.** *Let A be an instance of stable matching and let B be another instance obtained from A by changing the lists of only one side, either workers or firms, but not both. Then the matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ form a sublattice in each of the two lattices.*

▶ **Corollary 7.** *Let A be an instance of stable matching and let $B_1, \ldots, B_k$ be other instances obtained from A each by changing the lists of only one side, either workers or firms, but not both. Then the matchings in $\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots \cap \mathcal{M}_{B_k}$ form a sublattice in $\mathcal{M}_A$.*

This corollary gives another justification for Algorithm D, motivated by [21]. This modified Deferred Algorithm works when errors are only on one side. Algorithm D extends this to errors on both sides however it has exponential runtime.

This motivates us to characterize sublattices in the lattice of stable matchings. In Section 7.1, we show that for any instance B obtained by permuting the preference list of one worker or one firm, $\mathcal{M}_{A \setminus B}$ forms a semi-sublattice of $\mathcal{L}_A$ (Lemma 31). In particular, if the list of a worker is permuted, $\mathcal{M}_{A \setminus B}$ forms a join semi-sublattice of $\mathcal{L}_A$, and if the list of a firm is permuted, $\mathcal{M}_{A \setminus B}$ forms a meet semi-sublattice of $\mathcal{L}_A$. In both cases, $\mathcal{M}_A \cap \mathcal{M}_B$ is a sublattice of $\mathcal{L}_A$ and of $\mathcal{L}_B$ as shown in Proposition 6.

3 Birkhoff's Theorem on Sublattices

Let Π be a finite poset. For simplicity of notation, in this paper we will assume that Π must have *two dummy elements* s and t; the remaining elements will be called *proper elements* and the term *element* will refer to proper as well as dummy elements. The element s precedes all other elements and t succeeds all other elements in Π. A *proper closed set* of Π is any closed set that contains s and does not contain t. It is easy to see that the set of all proper closed sets of Π form a distributive lattice under the operations of set intersection and union. We will denote this lattice by $L(\Pi)$. The following has also been called *the fundamental theorem for finite distributive lattices.*

▶ **Theorem 8.** *(Birkhoff [7]) Every finite distributive lattice $\mathcal{L}$ is isomorphic to $L(\Pi)$, for some finite poset Π.*

Our application of Birkhoff's Theorem deals with the sublattices of a finite distributive lattice. First, in Definition 9 we state the critical operation of *compression of a poset.*

▶ **Definition 9.** *Given a finite poset Π, first partition its elements; each subset will be called a* meta-element*. Define the following precedence relations among the meta-elements: if x, y are elements of Π such that x is in meta-element X, y is in meta-element Y and x precedes y, then X precedes Y. Assume that these precedence relations yield a partial order, say Q, on the meta-elements (if not, this particular partition is not useful for our purpose). Let*

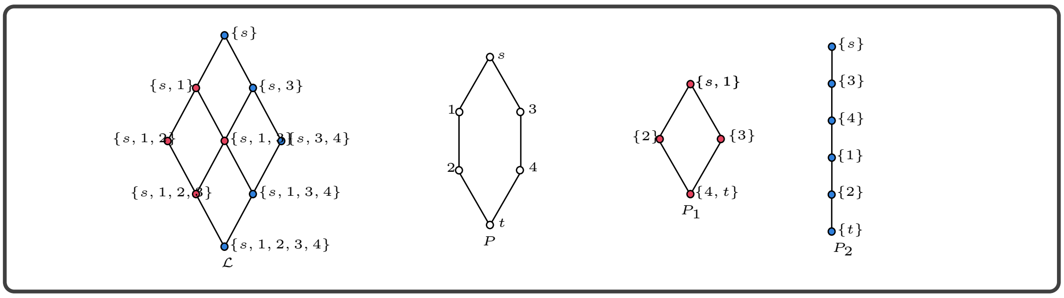

Figure 1 Two examples of compressions. Lattice $\mathcal{L} = L(P)$. P_1 and P_2 are compressions of P, and they generate the sublattices in $\mathcal{L}$, of red and blue elements, respectively. The black edges are directed from top to bottom so higher elements are predecessors of lower elements.

Π' *be any partial order on the meta-elements such that the precedence relations of* Q *are a subset of the precedence relations of* Π'. *Then* Π' *will be called a* compression *of* Π. *Let* A_s *and* A_t *denote the meta-elements of* Π' *containing* s *and* t, *respectively.*

For examples of compressions see Figure 1. Clearly, A_s precedes all other meta-elements in Π' and A_t succeeds all other meta-elements in Π'. Once again, by a *proper closed set of* Π' we mean a closed set of Π' that contains A_s and does not contain A_t. Then the lattice formed by the set of all proper closed sets of Π' will be denoted by $L(\Pi')$.

3.1 An alternative view of compression

In this section we give an alternative definition of compression of a poset; this will be used in the rest of the paper. The advantage of this definition is that it is much easier to work with for the applications presented later. Its drawback is that several different sets of edges may yield the same compression. Therefore, this definition is not suitable for stating a one-to-one correspondence between sublattices of $\mathcal{L}$ and compressions of Π. Finally we show that any compression Π' obtained using the first definition can also be obtained via the second definition and vice versa (Proposition 10), hence showing that the two definitions are equivalent for our purposes. See Appendix C for more details.

We are given a poset Π for a stable matching instance; let $\mathcal{L}$ be the lattice it generates. Let $H(\Pi)$ denote the Hasse diagram of Π. Consider the following operations to derive a new poset Π': Choose a set E of directed edges to add to $H(\Pi)$ and let H_E be the resulting graph. Let H' be the graph obtained by shrinking the strongly connected components of H_E; each strongly connected component will be called a meta-rotation of Π' as defined in Definition 9. The edges which are not shrunk will define a DAG, H', on the strongly connected components. These edges give precedence relations among meta-rotation for poset Π'.

Let $\mathcal{L}'$ be the sublattice of $\mathcal{L}$ generated by Π'. We will say that the set of edges E *defines* $\mathcal{L}'$. It can be seen that each set E uniquely defines a sublattice $L(\Pi')$; however, there may be multiple sets that define the same sublattice. See Figure 2 for examples of sets of edges which define sublattices.

▶ **Proposition 10.** *The two definitions of compression of a poset are equivalent.*

For a (directed) edge $e = uv \in E$, u is called the *tail* and v is called the *head* of e. Let I be a closed set of Π. Then we say that: I *separates* an edge $uv \in E$ if $v \in I$ and $u \notin I$; I

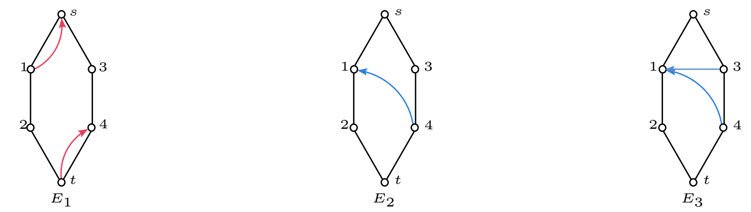

Figure 2 E_1 (red edges) and E_2 (blue edges) define the sublattices in Figure 1, of red and blue elements, respectively. E_2 and E_3 define the same compression and represent the same sublattice. All black edges in E_1, E_2 and E_3 are directed from top to bottom (not shown in the figure).

crosses an edge $uv \in E$ if $u \in I$ and $v \notin I$. If I does not separate or cross any edge $uv \in E$, I is called a *splitting* set w.r.t. E.

▶ **Lemma 11.** *Let $\mathcal{L}'$ be a sublattice of $\mathcal{L}$ and E be a set of edges defining $\mathcal{L}'$. A matching M is in $\mathcal{L}'$ iff the closed subset I generating M does not separate any edge $uv \in E$.*

▶ Remark 12. We may assume w.l.o.g. that the set E defining $\mathcal{L}'$ is *minimal* in the following sense: There is no edge $uv \in E$ such that uv is not separated by any closed set of Π. Observe that if there is such an edge, then $E \setminus \{uv\}$ defines the same sublattice $\mathcal{L}'$. Similarly, there is no edge $uv \in E$ such that each closed set separating uv also separates another edge in E.

▶ **Definition 13.** *W.r.t. an element v in a poset Π, we define four useful subsets of Π: $I_v = \{r \in \Pi : r \prec v\}, J_v = \{r \in \Pi : r \preceq v\} = I_v \cup \{v\}, I'_v = \{r \in \Pi : r \succ v\}, J'_v = \{r \in \Pi : r \succeq v\} = I'_v \cup \{v\}$. Notice that $I_v, J_v, \Pi \setminus I'_v, \Pi \setminus J'_v$ are all closed sets.*

▶ **Lemma 14.** *Both J_v and $\Pi \setminus J'_u$ separate uv for each $uv \in E$.*

Proof. Since uv is in E, u cannot be in J_v; otherwise, there is no closed subset separating uv, contradicting Remark 12. Hence, J_v separates uv for all uv in E. Similarly, since uv is in E, v cannot be in J'_u. Therefore, $\Pi \setminus J'_u$ contains v but not u, and thus separates uv. ◀

4 Setting I

Under Setting I, the given lattice $\mathcal{L}$ has sublattices $\mathcal{L}_1$ and $\mathcal{L}_2$ that partition $\mathcal{L}$. The main structural fact for this setting is:

▶ **Theorem 15.** *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be sublattices of $\mathcal{L}$ such that $\mathcal{L}_1$ and $\mathcal{L}_2$ partition $\mathcal{L}$. Then there exist sets of edges E_1 and E_2 defining $\mathcal{L}_1$ and $\mathcal{L}_2$ such that they form an alternating path from t to s.*

We will prove this theorem in the context of stable matchings. Let E_1 and E_2 be any two sets of edges defining $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively. We will show that E_1 and E_2 can be adjusted so that they form an alternating path from t to s, without changing the corresponding compressions.

▶ **Lemma 16.** *There must exist a path from t to s composed of edges in E_1 and E_2.*

Figure 3 Examples of: (a) canonical path, and (b) bouquet.

Let Q be a path from t to s according to Lemma 16. Partition Q into subpaths $Q_1, \ldots, Q_k$ such that each Q_i consists of edges in either E_1 or E_2 and $E(Q_i) \cap E(Q_{i+1}) = \emptyset$ for all $1 \leq i \leq k-1$. Let r_i be the rotation at the end of Q_i except for $i = 0$ where $r_0 = t$. Specifically, $t = r_0 \to r_1 \to \ldots \to r_k = s$ in Q. Lemma 11 can be used to show that each Q_i can be replaced by a direct edge from r_{i-1} to r_i, and furthermore, all edges not in Q can be removed.

▶ **Lemma 17.** *Let Q_i consist of edges in E_α ($\alpha = 1$ or 2). Q_i can be replaced by an edge from r_{i-1} to r_i where $r_{i-1}r_i \in E_\alpha$.*

▶ **Lemma 18.** *Edges in $E_1 \cup E_2$ but not in Q can be removed.*

By Lemma 17 and Lemma 18, $r_0r_1, \ldots, r_{k-2}r_{k-1}, r_{k-1}r_k$ are all edges in E_1 and E_2 and they alternate between E_1 and E_2. Therefore, we have Theorem 15. An illustration of such a path is given in Figure 3(a).

▶ **Proposition 19.** *There exists a sequence of rotations $r_0, r_1, \ldots, r_{2k}, r_{2k+1}$ such that a closed subset generates a matching in $\mathcal{L}_1$ iff it contains r_{2i} but not r_{2i+1} for some $0 \leq i \leq k$.*

5 Setting II

Under Setting II, the given lattice $\mathcal{L}$ can be partitioned into a sublattice $\mathcal{L}_1$ and a semi-sublattice $\mathcal{L}_2$. We assume that $\mathcal{L}_2$ is a join semi-sublattice. Clearly by reversing the order of $\mathcal{L}$, the case of meet semi-sublattice is also covered. The next theorem, which generalizes Theorem 15, gives a sufficient characterization of a set of edges E defining $\mathcal{L}_1$.

▶ **Theorem 20.** *There exists a set of edges E defining sublattice $\mathcal{L}_1$ such that:*
1. *The set of tails T_E of edges in E forms a chain in Π.*
2. *There is no path of length two consisting of edges in E.*
3. *For each $r \in T_E$, let $F_r = \{v \in \Pi : rv \in E\}$. Then any two rotations in F_r are incomparable.*
4. *For any $r_i, r_j \in T_E$ where $r_i \prec r_j$, there exists a splitting set containing all rotations in $F_{r_i} \cup \{r_i\}$ and no rotations in $F_{r_j} \cup \{r_j\}$.*

A set E satisfying Theorem 20 will be called a *bouquet*. For each $r \in T_E$, let $L_r = \{rv \mid v \in F_r\}$. Then L_r will be called a *flower*. Observe that the bouquet E is partitioned into flowers. These notions are illustrated in Figure 3(b). The black path, directed from s

to t, is the chain mentioned in Theorem 20 and the red edges constitute E. Observe that the tails of edges E lie on the chain. For each such tail, the edges of E outgoing from it constitute a flower.

Let E be an arbitrary set of edges defining $\mathcal{L}_1$. We will show that E can be modified so that the conditions in Theorem 20 are satisfied. Let S be a splitting set of Π. In other words, S is a closed subset such that for all $uv \in E$, either u, v are both in S or u, v are both in $\Pi \setminus S$. We can now replace paths with single edges as explained below.

▶ **Lemma 21.** *There is a unique maximal rotation in $T_E \cap S$.*

Denote by r the unique maximal rotation in $T_E \cap S$. Let $R_r = \{v \in \Pi : \text{there is a path from } r \text{ to } v \text{ using edges in } E\}, E_r = \{uv \in E : u, v \in R_r\}, G_r = \{R_r, E_r\}$. Note that $r \in R_r$. For each $v \in R_r$ there exists a path from r to v and $r \in S$. Since S does not cross any edge in the path, v must also be in S. Therefore, $R_r \subseteq S$.

▶ **Lemma 22.** *Let $u \in (T_E \cap S) \setminus R_r$ such that $u \succ x$ for $x \in R_r$. Then we can replace each $uv \in E$ with rv.*

Keep replacing edges according to Lemma 22 until there is no $u \in (T_E \cap S) \setminus R_r$ such that $u \succ x$ for some $x \in R_r$.

▶ **Lemma 23.** *Let $X = \{v \in S : v \succeq x \text{ for some } x \in R_r\}$. Then: $S \setminus X$ is a closed subset; $S \setminus X$ contains u for each $u \in (T_E \cap S) \setminus R_r$; $(S \setminus X) \cap R_r = \emptyset$; $S \setminus X$ is a splitting set.*

▶ **Lemma 24.** *E_r can be replaced by the following set of edges: $E'_r = \{rv : v \in R_r\}$.*

Proof of Theorem 20. To begin, let $S_1 = \Pi$ and let r_1 be the unique maximal rotation according to Lemma 21. Then we can replace edges according to Lemma 22 and Lemma 24. After replacing, r_1 is the only tail vertex in G_{r_1}. By Lemma 23, there exists a set X such that $S_1 \setminus X$ does not contain any vertex in R_{r_1} and contains all other tail vertices in T_E except r_1. Moreover, $S_1 \setminus X$ is a splitting set. Hence, we can set $S_2 = S_1 \setminus X$ and repeat.

Let $r_1, \dots, r_k$ be the rotations found in the above process. Since r_i is the unique maximal rotation in $T_E \cap S_i$ for all $1 \le i \le k$ and $S_1 \supset S_2 \supset \dots \supset S_k$, we have $r_1 \succ r_2 \succ \dots \succ r_k$. By Lemma 24, for each $1 \le i \le k$, E_{r_i} consists of edges $r_i v$ for $v \in R_{r_i}$. Therefore, there is no path of length two composed of edges in E and condition 2 is satisfied. Moreover, $r_1, \dots, r_k$ are exactly the tail vertices in T_E, which gives condition 1.

Let r be a rotation in T_E and consider $u, v \in F_r$. Moreover, assume that $u \prec v$. A closed subset I separating rv contains v but not r. Since I is a closed subset and $u \prec v$, I contains u. Therefore, I also separates ru, contradicting the assumption in Remark 12. The same argument applies when $v \prec u$. Therefore, u and v are incomparable as stated in condition 3.

Finally, let $r_i, r_j \in T_E$ where $r_i \prec r_j$. By the construction given above, $S_j \supset S_{j-1} \supset \dots \supset S_i$, $R_{r_j} \subseteq S_j \setminus S_{j-1}$ and $R_{r_i} \subseteq S_i$. Therefore, S_i contains all rotations in R_{r_i} but none of the rotations in R_{r_j}, giving condition 4 which can be restated as Proposition 25. ◀

▶ **Proposition 25.** *There exists a sequence of rotations $r_1 \prec \dots \prec r_k$ and a set F_{r_i} for each $1 \le i \le k$ such that a closed subset generates a matching in $\mathcal{L}_1$ if and only if whenever it contains a rotation in F_{r_i}, it must also contain r_i.*

FindBouquet(Π):
Input: A poset Π.
Output: A set E of edges defining $\mathcal{L}_1$.

1. Initialize: Let $S = \Pi, E = \emptyset$.
2. If M_z is in $\mathcal{L}_1$: go to Step 3. Else: $r = t$, go to Step 5.
3. r = FindNextTail(Π, S).
4. If r is not Null: Go to Step 5. Else: Go to Step 7.
5. F_r = FindFlower(Π, S, r).
6. Update:
   a. For each $u \in F_r$: $E \leftarrow E \cup \{ru\}$.
   b. $S \leftarrow S \setminus \bigcup_{u \in F_r \cup \{r\}} J'_u$.
   c. Go to Step 3.
7. Return E.

Figure 4 Algorithm for finding a bouquet.

6 Algorithm for Finding a Bouquet

In this section, we give an algorithm for finding a bouquet. Let $\mathcal{L}$ be a distributive lattice that can be partitioned into a sublattice $\mathcal{L}_1$ and a semi-sublattice $\mathcal{L}_2$. Then given a poset Π of $\mathcal{L}$ and a membership oracle, which determines if a matching of $\mathcal{L}$ is in $\mathcal{L}_1$ or not, the algorithm returns a bouquet defining $\mathcal{L}_1$.

By Theorem 20, the set of tails T_E forms a chain C in Π. The idea of our algorihm, given in Figure 4, is to find the flowers according to their order in C. Specifically, a splitting set S is maintained such that at any point, all flowers outside of S are found. At the beginning, S is set to Π and becomes smaller as the algorithm proceeds. Step 2 checks if M_z is a matching in $\mathcal{L}_1$ or not. If $M_z \notin \mathcal{L}_1$, the closed subset $\Pi \setminus \{t\}$ separates an edge in E according to Lemma 11. Hence, the first tail on C must be t. Otherwise, the algorithm jumps to Step 3 to find the first tail. Each time a tail r is found, Step 5 immediately finds the flower L_r corresponding to r. The splitting set S is then updated so that S no longer contains L_r but still contains the flowers that have not been found yet. Next, our algorithm continues to look for the next tail inside the updated S. If no tail is found, it terminates.

▶ **Lemma 26.** *Let v be a rotation in Π. Let $S \subseteq \Pi$ such that both S and $S \cup \{v\}$ are closed subsets. If S generates a matching in $\mathcal{L}_1$ and $S \cup \{v\}$ generates a matching in $\mathcal{L}_2$, v is the head of an edge in E. If S generates a matching in $\mathcal{L}_2$ and $S \cup \{v\}$ generates a matching in $\mathcal{L}_1$, v is the tail of an edge in E.*

Proof. Suppose that S generates a matching in $\mathcal{L}_1$ and $S \cup \{v\}$ generates a matching in $\mathcal{L}_2$. By Lemma 11, S does not separate any edge in E, and $S \cup \{v\}$ separates an edge $e \in E$. This can only happen if u is the head of e.

A similar argument can be given for the second case. ◀

▶ **Lemma 27.** *Given a splitting set S, FindNextTail(Π, S) (Figure 5) returns the maximal tail vertex in S, or Null if there is no tail vertex in S.*

FindNextTail(Π, S):
Input: A poset Π, a splitting set S.
Output: The maximal tail vertex in S, or Null if there is no tail vertex in S.

1. Compute the set V of rotations v in S such that:
   - $\Pi \setminus I'_v$ generates a matching in $\mathcal{L}_1$.
   - $\Pi \setminus J'_v$ generates a matching in $\mathcal{L}_2$.
2. If $V \neq \emptyset$ and there is a unique maximal element v in V: Return v.
   Else: Return Null.

Figure 5 Subroutine for finding the next tail.

FindFlower(Π, S, r):
Input: A poset Π, a tail vertex r and a splitting set S containing r.
Output: The set $F_r = \{v \in \Pi : rv \in E\}$.

1. Compute $X = \{v \in I_r : J_v$ generates a matching in $\mathcal{L}_1\}$.
2. Let $Y = \bigcup_{v \in X} J_v$.
3. If $Y = \emptyset$ and $M_0 \in \mathcal{L}_2$: Return $\{s\}$.
4. Compute the set V of rotations v in S such that:
   - $Y \cup I_v$ generates a matching in $\mathcal{L}_1$.
   - $Y \cup J_v$ generates a matching in $\mathcal{L}_2$.
5. Return V.

Figure 6 Subroutine for finding a flower.

▶ **Lemma 28.** *Given a tail vertex r and a splitting set S containing r, FindFlower(Π, S, r) (Figure 6) correctly returns F_r.*

▶ **Theorem 29** (h)**.** *FindBouquet(Π), given in Figure 4, returns a set of edges defining $\mathcal{L}_1$.*

Proof. From Lemmas 27 and 28, it suffices to show that S is udpated correctly in Step 6(b). To be precise, we need that

$$S \setminus \bigcup_{u \in F_r \cup \{r\}} J'_u$$

must still be a splitting set, and contains all flowers that have not been found. This follows from Lemma 23 by noticing that

$$\bigcup_{u \in F_r \cup \{r\}} J'_u = \{v \in \Pi : v \succeq u \text{ for some } u \in R_r\}.$$

◀

Clearly, a sublattice of $\mathcal{L}$ must also be a semi-sublattice. Therefore, FindBouquet can be used to find a canonical path described in Section 4. The same algorithm can be used to check if $M_A \cap M_B = \emptyset$. Let E be the edge set given by the FindBouquet algorithm and H_E be the corresponding graph obtained by adding E to the Hasse diagram of the original

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

firms' preferences in A — firms' preferences in B — workers' preferences in both instances

Figure 7 An example in which $\mathcal{M}_{A\setminus B}$ is not a sublattice of $\mathcal{L}_A$.

rotation poset Π of $\mathcal{L}_A$. If H_E has a single strongly connected component, the compression Π' has a single meta-element and represents the empty lattice.

7 Finding a Fully Robust Stable Matching

Consider the setting given in the Introduction, with S being the domain of errors, one of which is introduced in instance A. We show how to use the algorithm in Section 6 to find the poset generating all fully robust matchings w.r.t. S. We then show how this poset can yield a fully robust matching that maximizes, or minimizes, a given weight function.

7.1 Studying semi-sublattices is necessary and sufficient

Let A be a stable matching instance, and B be an instance obtained by permuting the preference list of one worker or one firm. Lemma 30 gives an example of a permutation so that $\mathcal{M}_{A\setminus B}$ is not a sublattice of $\mathcal{L}_A$, hence showing that the case studied in Section 4 does not suffice to solve the problem at hand. On the other hand, for all such instances B, Lemma 31 shows that $\mathcal{M}_{A\setminus B}$ forms a semi-sublattice of $\mathcal{L}_A$ and hence the case studied in Section 5 does suffice.

The next lemma pertains to the example given in Figure 7, in which the set of workers is $\mathcal{B} = \{a, b, c, d\}$ and the set of firms is $\mathcal{G} = \{1, 2, 3, 4\}$. Instance B is obtained from instance A by permuting firm 1's list.

▶ **Lemma 30.** *There exist stable matching instances A and B differing by one agent's preference list such that $M_{A\setminus B}$ is not a sublattice of $\mathcal{L}_A$.*

▶ **Lemma 31.** *For any instance B obtained by permuting the preference list of one worker or one firm, $\mathcal{M}_{A\setminus B}$ forms a semi-sublattice of $\mathcal{L}_A$.*

▶ **Proposition 32.** *A set of edges defining the sublattice $\mathcal{L}'$, consisting of matchings in $\mathcal{M}_A \cap \mathcal{M}_B$, can be computed in polynomial time.*

7.2 Proof of Theorem 2

In this section, we will prove Theorem 2 as well as a slight extension; the latter uses ideas from [18]. Let $B_1, \ldots, B_k$ be polynomially many instances in the domain $D \subset T$, as defined in the Introduction. Let E_i be the set of edges defining $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ for all $1 \le i \le k$. By Corollary 7, $\mathcal{L}' = \mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots \cap \mathcal{M}_{B_k}$ is a sublattice of $\mathcal{L}_A$.

▶ **Lemma 33.** $E = \bigcup_i E_i$ *defines* $\mathcal{L}'$.

Proof. By Lemma 11, it suffices to show that for any closed subset I, I does not separate an edge in E iff I generates a matching in $\mathcal{L}'$.

I does not separate an edge in E iff I does not separate any edge in E_i for all $1 \leq i \leq k$ iff the matching generated by I is in $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ for all $1 \leq i \leq k$ by Lemma 11. ◀

By Lemma 33, a compression Π' generating $\mathcal{L}'$ can be constructed from E as described in Section 3.1. By Proposition 32, we can compute each E_i, and hence, Π' in polynomial time. Clearly, Π' can be used to check if a fully robust stable matching exists. To be precise, a fully robust stable matching exists iff there exists a proper closed subset of Π'. This happens iff s and t belong to different meta-rotations in Π', an easy to check condition. Hence, we have Theorem 2.

7.3 Finding maximum weight fully robust stable matchings

We can use Π' to obtain a fully robust stable matching M maximizing $\sum_{wf \in M} W_{wf}$ by applying the algorithm of [19]. Specifically, let $H(\Pi')$ be the Hasse diagram of Π'. Then each pair wf for $w \in \mathcal{W}$ and $f \in \mathcal{F}$ can be associated with two vertices u_{wf} and v_{wf} in $H(\Pi')$ as follows:

- If there is a rotation r moving w to f, u_{wf} is the meta-rotation containing r. Otherwise, u_{wf} is the meta-rotation containing s.
- If there is a rotation r moving w from f, v_{wf} is the meta-rotation containing r. Otherwise, v_{wf} is the meta-rotation containing t.

By Lemma 3 and the definition of compression, $u_{wf} \prec v_{wf}$. Hence, there is a path from u_{wf} to v_{wf} in $H(\Pi')$. We can then add weights to edges in $H(\Pi')$, as stated in [19]. Specifically, we start with weight 0 on all edges and increase weights of edges in a path from u_{wf} to v_{wf} by w_{wf} for all pairs wf. A fully robust stable matching maximizing $\sum_{wf \in M} W_{bwf}$ can be obtained by finding a maximum weight ideal cut in the constructed graph. An efficient algorithm for the latter problem is given in [19].

8 Discussion

The primary focus of this paper is the study of "nearby" stable matching instances where a single agent permutes their preference list. A number of new questions arise: give a polynomial time algorithm for the problem mentioned in the Introduction, of finding a robust stable matching as defined in [19] — given a probability distribution on the domain of errors — even when the error is an arbitrary permutation; and extend to the stable roommate problem and incomplete preference lists [15, 20], as well as popular matchings [10, 16].

Next, we give a hypothetical setting to show potential application of our work to the issue of incentive compatibility. Let A be an instance of stable matching over n workers and n firms. Assume that all $2n$ agents have a means of making their preference lists public simultaneously and a dominant firm, say f, is given the task of computing and announcing a stable matching. Once the matching is announced, all agents can verify that it is indeed stable. It turns out that firm f can cheat and improve its match as follows: f changes its preference list to obtain instance B which is identical to A for all other agents, and computes a matching that is stable for A as well as B using Theorem 2. The other agents will be satisfied that this matching is indeed stable for instance A and f's cheating may go undetected.

Finally, considering the number of new and interesting matching markets being defined on the Internet, e.g., see [13], it will not be surprising if new, deeper structural facts about stable matching lattices find suitable applications. For this reason, the problem initiated in [18], which appears to be a fundamental one, deserves further work. In particular, we leave the question of extending our work to the case when the two instances A and B are not nearby but arbitrary, i.e., when multiple agents simultaneously change their preference lists.

References

1. Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. *American Economic Review*, 99(5):1954–78, 2009.
2. Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. The boston public school match. *American Economic Review*, 95(2):368–371, 2005.
3. Atila Abdulkadiroğlu and Tayfun Sonmez. School choice: A mechanism design approach. *American economic review*, 93(3):729–747, 2003.
4. H. Aziz, P. Biro, T. Fleiner, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable matching with uncertain pairwise preferences. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 344–352, 2017.
5. H. Aziz, P. Biro, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable matching with uncertain linear preferences. In *International Symposium on Algorithmic Game Theory*, pages 195–206, 2016.
6. A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.
7. Garrett Birkhoff. Rings of sets. *Duke Mathematical Journal*, 3(3):443–454, 1937.
8. Charles Blair. Every finite distributive lattice is a set of stable matchings. *Journal of Combinatorial Theory, Series A*, 37(3):353–356, 1984.
9. G. C. Calafiore and L. El Ghaoui. On distributionally robust chance-constrained linear programs. *Jour. of Optimization Theory and Applications*, 130(1), December 2006.
10. Agnes Cseh, Yuri Faenza, Telikepalli Kavitha, and Vladlena Powers. Understanding popular matchings via stable matchings. *SIAM Journal on Discrete Mathematics*, 36(1):188–213, 2022.
11. L. E. Dubins and D. A. Freedman. Machiavelli and the gale-shapley algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981.
12. Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani, editors. *Online and Matching-Based Market Design*. Cambridge University Press, to appear. `https://www.ics.uci.edu/~vazirani/Chapter1.pdf`.
13. Simons Institute for the Theory of Computing. Online and matching-based market design, 2019. `https://simons.berkeley.edu/programs/market2019`.
14. David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
15. Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*. MIT press, 1989.
16. Telikepalli Kavitha. Matchings, critical nodes, and popular solutions. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
17. Donald Ervin Knuth. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*. American Mathematical Soc., 1997.
18. Tung Mai and Vijay V. Vazirani. Finding stable matchings that are robust to errors in the input. In *European Symposium on Algorithms*, 2018.
19. Tung Mai and Vijay V. Vazirani. A natural generalization of stable matching solved via new insights into ideal cuts. In arXiv, 2018.
20. David Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2013.

21 Shuichi Miyazaki and Kazuya Okamoto. Jointly stable matchings. *Journal of Combinatorial Optimization*, 38(2):646–665, 2019.

22 Alvin E Roth and Marilda Sotomayor. Two-sided matching. *Handbook of game theory with economic applications*, 1:485–541, 1992.

23 Richard Stanley. Enumerative combinatorics, vol. 1, Wadsworth and Brooks/Cole, Pacific Grove, CA, 1986; second printing, 1996.

A Related Work

A.1 Related work

The two topics, of stable matching and the design of algorithms that produce solutions that are robust to errors, have been studied extensively for decades and there are today several books on each of them, e.g., see [17, 15, 20] and [9, 6]. Yet, there is a paucity of results at the intersection of these two topics. Indeed, before the publication of [18], we are aware of only two previous works [5, 4]. We remark that the notion of robustness studied in [18] was quite different from that of the previous two works as detailed below.

Aziz et al. [5] considered the problem of finding stable matching under uncertain linear preferences. They proposed three different uncertainty models:

1. Lottery Model: Each agent has a probability distribution over strict preference lists, independent of other agents.
2. Compact Indifference Model: Each agent has a single weak preference list in which ties may exist. All linear order extensions of this weak order have equal probability.
3. Joint Probability Model: A probability distribution over preference profiles is specified.

They showed that finding the matching with highest probability of being stable is NP-hard for the Compact Indifference Model and the Joint Probability Model. For the very special case that preference lists of one side are certain and the number of uncertain agents of the other side are bounded by a constant, they gave a polynomial time algorithm that works for all three models.

The joint probability model is the most powerful and closest to our setting. The main difference is that in their model, there is no base instance, which is called A in our model. The opportunity of finding new structural results arises from our model precisely because we need to consider two "nearby" instances, namely A and B as described above.

Aziz et al. [4] introduced a pairwise probability model in which each agent gives the probability of preferring one agent over another for all possible pairs. They showed that the problem of finding a matching with highest probability of being stable is NP-hard even when no agent has a cycle in its certain preferences (i.e., the ones that hold with probability 1).

B Proof of Birkhoff's Theorem using Stable Matching Lattices

Omitted proofs can be found in the *Arxiv version*.

C Other Omitted Proofs

Proof of Lemma 6. It suffices to show that $\mathcal{M}_A \cap \mathcal{M}_B$ is a sublattice of $\mathcal{L}_A$. Assume $|\mathcal{M}_A \cap \mathcal{M}_B| > 1$ and let M_1 and M_2 be two different matchings in $\mathcal{M}_A \cap \mathcal{M}_B$. Let $\vee_A$ and $\vee_B$ be the join operations under A and B respectively. Likewise, let $\wedge_A$ and $\wedge_B$ be the meet operations under A and B.

By definition of join operation in Section 2.1, $M_1 \vee_A M_2$ is the matching obtained by assigning each worker to its less preferred partner (or equivalently, each firm to its more preferred partner) from M_1 and M_2 according to instance A. Without loss of generality, assume that B is an instance obtained from A by changing the lists of only firms. Since the list of each worker is identical in A and B, its less preferred partner from M_1 and M_2 is also the same in A and B. Therefore, $M_1 \vee_A M_2 = M_1 \vee_B M_2$. A similar argument can be applied to show that $M_1 \wedge_A M_2 = M_1 \wedge_B M_2$.

Hence, $M_1 \vee_A M_2$ and $M_1 \wedge_A M_2$ are both in $\mathcal{M}_A \cap \mathcal{M}_B$ as desired. ◀

Proof of Corollary 7. Assume $|\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots \cap \mathcal{M}_{B_k}| > 1$ and let M_1 and M_2 be two different matchings in $\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots \cap \mathcal{M}_{B_k}$. Therefore, M_1 and M_2 are in $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ for each $1 \le i \le k$. By Proposition 6, $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ is a sublattice of $\mathcal{L}_A$. Hence, $M_1 \vee_A M_2$ and $M_1 \wedge_A M_2$ are in $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ for each $1 \le i \le k$. The claim then follows. ◀

Proof of Proposition 10. Let Π' be a compression of Π obtained using the first definition. Clearly, for each meta-rotation in Π', we can add edges to Π so the strongly connected component created is precisely this meta-rotation. Any additional precedence relations introduced among incomparable meta-rotations can also be introduced by adding appropriate edges.

The other direction is even simpler, since each strongly connected component can be defined to be a meta-rotation and extra edges added can also be simulated by introducing new precedence constraints. ◀

Proof of Lemma 11. Let Π' be a compression corresponding to $\mathcal{L}'$. By Theorem 1, the matchings in $\mathcal{L}'$ are generated by eliminating rotations in closed subsets of Π'.

First, assume I separates $uv \in E$. Moreover, assume $M \in \mathcal{L}'$ for the sake of contradiction, and let I' be the closed subset of Π' corresponding to M. Let U and V be the meta-rotations containing u and v respectively. Notice that the sets of rotations in I and I' are identical. Therefore, $V \in I'$ and $U \notin I'$. Since $uv \in E$, there is an edge from U to V in H'. Hence, I' is not a closed subset of Π'.

Next, assume that I does not separate any $uv \in E$. We show that the rotations in I can be partitioned into meta-rotations in a closed subset I' of Π'. If I cannot be partitioned into meta-rotations, there must exist a meta-rotation A such that $A \cap I$ is a non-empty proper subset of A. Since A consists of rotations in a strongly connected component of H_E, there must be an edge uv from $A \setminus I$ to $A \cap I$ in H_E. Hence, I separates uv. Since I is a closed subset, uv can not be an edge in H. Therefore, $uv \in E$, which is a contradiction. It remains to show that the set of meta-rotations partitioning I is a closed subset of Π'. Assume otherwise, there exist meta-rotation $U \in I'$ and $V \notin I'$ such that there exists an edge from U to V in H'. Therefore, there exists $u \in U$, $v \in V$ and $uv \in E$, which is a contradiction. ◀

Proof of Lemma 16. Let R denote the set of vertices reachable from t by a path of edges in E_1 and E_2. Assume by contradiction that R does not contain s. Consider the matching M generated by rotations in $\Pi \setminus R$. Without loss of generality, assume that $M \in \mathcal{L}_1$. By Lemma 11, $\Pi \setminus R$ separates an edge $uv \in E_2$. Therefore, $u \in R$ and $v \in \Pi \setminus R$. Since $uv \in E_2$, v is also reachable from t by a path of edges in E_1 and E_2. ◀

Proof of Lemma 17. A closed subset separating $r_{i-1}r_i$ must separate an edge in Q_i. Moreover, any closed subset must separate exactly one of $r_0r_1, \ldots, r_{k-2}r_{k-1}, r_{k-1}r_k$. Therefore, the set of closed subsets separating an edge in E_1 (or E_2) remains unchanged. ◀

Proof of Lemma 18. Let e be an edge in $E_1 \cup E_2$ but not in Q. Suppose that $e \in E_1$. Let I be a closed subset separating e. By Lemma 11, the matching generated by I belongs to $\mathcal{L}_2$. Since e is not in Q and Q is a path from t to s, I must separate another edge e' in Q. By Lemma 11, I can not separate edges in both E_1 and E_2. Therefore, e' must also be in E_1. Hence, the matching generated by I will still be in $\mathcal{L}_2$ after removing e from E_1. The argument applies to all closed subsets separating e. ◀

Proof of Lemma 21. Suppose there are at least two maximal rotations $u_1, u_2, \ldots u_k$ $(k \geq 2)$ in $T_E \cap S$. Let $v_1, \ldots v_k$ be the heads of edges containing $u_1, u_2, \ldots u_k$. For each $1 \leq i \leq k$, let $S_i = J_{u_i} \cup J_{v_j}$ where j is any index such that $j \neq i$. Since u_i and u_j are incomparable, $u_j \notin J_{u_i}$. Moreover, $u_j \notin J_{v_j}$ by Lemma 14. Therefore, $u_j \notin S_i$. It follows that S_i contains u_i and separates $u_j v_j$. Since S_i separates $u_j v_j \in E$, the matching generated by S_i is in $\mathcal{L}_2$ according to Lemma 11.

Since $\bigcup_{i=1}^{k} S_i$ contains all maximal rotations in $T_E \cap S$ and S does not separate any edge in E, $\bigcup_{i=1}^{k} S_i$ does not separate any edge in E either. Therefore, the matching generated by $\bigcup_{i=1}^{k} S_i$ is in $\mathcal{L}_1$, and hence not in $\mathcal{L}_2$. This contradicts the fact that $\mathcal{L}_2$ is a join semi-sublattice. ◀

Proof of Lemma 22. We will show that the set of closed subsets separating an edge in E remains unchanged.

Let I be a closed subset separating uv. Then I must also separate rv since $r \succ v$.

Now suppose I is a closed subset separating rv. We consider two cases:

- If $u \in I$, I must contain x since $u \succ x$. Hence, I separates an edge in the path from r to x.
- If $u \notin I$, I separates uv.

◀

Proof of Lemma 23. The lemma follows from the claims given below:

▷ Claim 34. $S \setminus X$ is a closed subset.

Proof. Let v be a rotation in $S \setminus X$ and u be a predecessor of v. Since S is a closed subset, $u \in S$. Notice that if a rotation is in X, all of its successor must be included. Hence, since $v \notin X$, $u \notin X$. Therefore, $u \in S \setminus X$. ◀

▷ Claim 35. $S \setminus X$ contains u for each $u \in (T_E \cap S) \setminus R_r$.

Proof. After replacing edges according to Lemma 22, for each $u \in (T_E \cap S) \setminus R_r$ we must have that u does not succeed any $x \in R_r$. Therefore, $u \notin X$ by the definition of X. ◀

▷ Claim 36. $(S \setminus X) \cap R_r = \emptyset$.

Proof. Since $R_r \subseteq X$, $(S \setminus X) \cap R_r = \emptyset$. ◀

▷ Claim 37. $S \setminus X$ does not separate any edge in E.

Proof. Suppose $S \setminus X$ separates $uv \in E$. Then $u \in X$ and $v \in S \setminus X$. By Claim 2, u can not be a tail vertex, which is a contradiction. ◀

▷ Claim 38. $S \setminus X$ does not cross any edge in E.

Proof. Suppose $S \setminus X$ crosses $uv \in E$. Then $u \in S \setminus X$ and $v \in X$. Let J be a closed subset separating uv. Then $v \in J$ and $u \notin J$.

Since $uv \in E$ and $u \in S$, $u \in T_E \cap S$. Therefore, $r \succ u$ by Lemma 21. Since J is a closed subset, $r \notin J$.

Since $v \in X$, $v \succeq x$ for $x \in R_r$. Again, as J is a closed subset, $x \in J$.

Therefore, J separates an edge in the path from r to x in G_r. Hence, all closed subsets separating uv must also separate another edge in E_r. This contradicts the assumption made in Remark 12. ◀

◀

Proof of Lemma 24. We will show that the set of closed subsets separating an edge in E_r and the set of closed subset separating an edge in E'_r are identical.

Consider a closed subset I separating an edge in $rv \in E'_r$. Since $v \in R_r$, I must separate an edge in E in a path from r to v. By definition, that edge is in E_r.

Now let I be a closed subset separating an edge in $uv \in E_r$. Since $uv \in E$, $u \in T_E \cap S$. By Lemma 21, $r \succ u$. Thus, I must also separate $rv \in E'_r$. ◀

Proof of Lemma 27. Let r be the maximal tail vertex in S.

First we show that $r \in V$. By Theorem 20, the set of tails of edges in E forms a chain in Π. Therefore $\Pi \setminus I'_r$ contains all tails in S. Hence, $\Pi \setminus I'_r$ does not separate any edge whose tails are in S. Since S is a splitting set, $\Pi \setminus I'_r$ does not separate any edge whose tails are in $\Pi \setminus S$. Therefore, by Lemma 11, $\Pi \setminus I'_r$ generates a matching in $\mathcal{L}_1$. By Lemma 14, $\Pi \setminus J'_r$ must separate an edge in E, and hence generates a matching in $\mathcal{L}_2$ according to Lemma 11.

By Lemma 26, any rotation in V must be the tail of an edge in E. Hence, they are all predecessors of r according to Theorem 20. ◀

Proof of Lemma 28. First we give two crucial properties of the set Y. By Theorem 20, the set of tails of edges in E forms a chain C in Π.

▷ Claim 39. Y contains all predecessors of r in C.

Proof. Assume that there is at least one predecessor of r in C, and denote by r' the direct predecessor. It suffices to show that $r' \in Y$. By Theorem 20, there exists a splitting set I such that $R_{r'} \subseteq I$ and $R_r \cap I = \emptyset$. Let v be the maximal element in $C \cap I$. Then v is a successor of all tail vertices in I. It follows that J_v does not separate any edges in E inside I. Therefore, $v \in X$. Since $J_v \subseteq Y$, Y contains all predecessors of r in C. ◀

▷ Claim 40. Y does not contain any rotation in F_r.

Proof. Since Y is the union of closed subset generating matching in $\mathcal{L}_1$, Y also generates a matching in $\mathcal{L}_1$. By Lemma 11, Y does not separate any edge in E. Since $r \notin Y$, Y must not contain any rotation in F_r. ◀

By Claim 1, if $Y = \emptyset$, r is the last tail found in C. Hence, if $M_0 \in \mathcal{L}_2$, s must be in F_r. By Theorem 20, the heads in F_r are incomparable. Therefore, s is the only rotation in C. FindFlower correctly returns $\{s\}$ in Step 3. Suppose such a situation does not happen, we will show that the returned set is F_r.

▷ Claim 41. $V = F_r$.

Proof. Let v be a rotation in V. By Lemma 26, v is a head of some edge e in E. Since Y contains all predecessors of r in C, the tail of e must be r. Hence, $v \in F_r$.

Let v be a rotation in F_r. Since Y contains all predecessors of r in C, $Y \cup I_v$ can not separate any edge whose tails are predecessors of r. Moreover, by Theorem 20, the heads in F_r are incomparable. Therefore, I_v does not contain any rotation in F_r. Since Y does not contain any rotation in F_r by the above claim, $Y \cup I_v$ does not separate any edge in E. It follows that $Y \cup I_v$ generates a matching in $\mathcal{L}_1$. Finally, $Y \cup J_v$ separates rv clearly, and hence generates a matching in $\mathcal{L}_2$. Therefore, $v \in V$ as desired. ◀

◀

Proof of Lemma 30. $M_1 = \{1a, 2b, 3d, 4c\}$ and $M_2 = \{1b, 2a, 3c, 4d\}$ are stable matching with respect to instance A. Clearly, $M_1 \wedge_A M_2 = \{1a, 2b, 3c, 4d\}$ is also a stable matching under A.

In going from A to B, the positions of workers b and c are swapped in firm 1's list. Under B, $1c$ is a blocking pair for M_1 and $1a$ is a blocking pair for M_2. Hence, M_1 and M_2 are both in $\mathcal{M}_{A \setminus B}$. However, $M_1 \wedge_A M_2$ is a stable matching under B, and therefore is it not in $\mathcal{M}_{A \setminus B}$. Hence, $\mathcal{M}_{A \setminus B}$ is not closed under the $\wedge_A$ operation. ◀

Proof of Lemma 31. Assume that the preference list of a firm f is permuted. We will show that $\mathcal{M}_{A \setminus B}$ is a join semi-sublattice of $\mathcal{L}_A$. By switching the role of workers and firms, permuting the list of a worker will result in $\mathcal{M}_{A \setminus B}$ being a meet semi-sublattice of $\mathcal{L}_A$.

Let M_1 and M_2 be two matchings in $\mathcal{M}_{A \setminus B}$. Hence, neither of them are in $\mathcal{M}_B$. In other words, each has a blocking pair under instance B.

Let w be the partner of f in $M_1 \vee_A M_2$. Then w must also be matched to f in either M_1 or M_2 (or both). We may assume that w is matched to f in M_1.

Let xy be a blocking pair of M_1 under B. We will show that xy must also be a blocking pair of $M_1 \vee_A M_2$ under B. To begin, the firm y must be f since other preference lists remain unchanged. Since xf is a blocking pair of M_1 under B, $x >_f^B w$. Similarly, $f >_x f'$ where f' is the M_1-partner of x. Let f'' be the partner of x in $M_1 \vee_A M_2$. Then $f' \geq_x f''$. It follows that $f >_x f''$. Since $x >_f^B w$ and $f >_x f''$, xf must be a blocking pair of $M_1 \vee_A M_2$ under B. ◀

Proof of Proposition 32. We have that $\mathcal{L}'$ and $\mathcal{M}_{A \setminus B}$ partition $\mathcal{L}_A$, with $\mathcal{M}_{A \setminus B}$ being a semi-sublattice of $\mathcal{L}_A$, by Lemma 31. Therefore, FindBouquet(Π) finds a set of edges defining $\mathcal{L}'$ by Theorem 29.

By Lemma 4, the input Π to FindBouquet can be computed in polynomial time. Clearly, a membership oracle checking if a matching is in $\mathcal{L}'$ or not can also be implemented efficiently. Since Π has $O(n^2)$ vertices (Lemma 4), any step of FindBouquet takes polynomial time. ◀

D Modified Deferred Acceptance Algorithms

Omitted algorithms and proofs can be found in the *Arxiv version*.