DISTRIBUTIVE LATTICES, STABLE MATCHINGS, AND ROBUST SOLUTIONS

A Dissertation
Presented to
The Academic Faculty

By

Tung Mai

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Algorithms, Combinatorics, and Optimization

Georgia Institute of Technology

August 2018

DISTRIBUTIVE LATTICES, STABLE MATCHINGS, AND ROBUST SOLUTIONS

Committee Members:

Dr. Vijay V. Vazirani, Advisor
Department of Computer Science
University of California, Irvine

Dr. Jugal Garg
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

Dr. Milena Mihail
School of Computer Science
Georgia Institute of Technology

Dr. Mohit Singh
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Robin Thomas
School of Mathematics
Georgia Institute of Technology

Defense Date: May 17, 2018

To my parents.

ACKNOWLEDGEMENTS

I would first and foremost like to thank my advisor Vijay Vazirani. Although I only started working with Vijay on my third year, his encouragement, support and guidance have been invaluable for me.

I would like to thank Robin Thomas for managing the ACO program and providing me support countless number of times.

I would like to thank Jugal Garg, Milena Mihail and Mohit Singh for agreeing to serve on my committee, and Ioannis Panageas for agreeing to be my thesis reader.

I would like to thank Chirs Peikert, Richard Peng, Prasad Tetali, Eric Vigoda for their encouragement and guidance.

Finally I would like to thank Samira Samadi, Saurabh Sawlani, Anup Rao and Peng Zhang for being awesome friends.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

The stable matching problem, first presented by mathematical economists Gale and Shapley, has been studied extensively since its introduction. As a result, a remarkably rich literature on the problem has accumulated in both theory and practice. In this thesis we further extend our understanding on several algorithmic and structural aspects of stable matching. We summarize the main contributions of the thesis as follows:

1. **Generalizing stable matching to maximum weight stable matching.** We study a natural generalization of stable matching to the maximum weight stable matching problem and we obtain a combinatorial polynomial time algorithm for it by reducing it to the problem of finding a maximum weight ideal cut in a DAG. We give the first polynomial time algorithm for the latter problem; this algorithm is also combinatorial.

   The combinatorial nature of our algorithms not only means that they are efficient but also that they enable us to obtain additional structural and algorithmic results:

   - We show that the set, $\mathcal{M}'$, of maximum weight stable matchings forms a sublattice $\mathcal{L}'$ of the lattice $\mathcal{L}$ of all stable matchings $\mathcal{M}$.
   - We give an efficient algorithm for finding boy-optimal and girl-optimal matchings in $\mathcal{L}'$.
   - We generalize the notion of rotation, a central structural notion in the context of the stable matching problem, to *meta-rotation*. Just as rotations help traverse the lattice $\mathcal{L}$, meta-rotations help traverse the sublattice $\mathcal{L}'$.

2. **Finding stable matchings that are robust to shifts.** We give a polynomially large class of errors, D, that can be introduced in a stable matching instance. Given an instance A of stable matching, let B be the instance that results after introducing one

error from D, chosen via a discrete probability distribution. We want to find a stable matching for A that maximizes the probability of being stable for B as well. Via new structural properties, related to the lattice of stable matchings, we give a polynomial time algorithm for this problem, where the domain of D consists of *shifts*, defined in Chapter 3.

3. **Generalizing Birkhoff's theorem, and an application on robust stable matching.** Birkhoff's theorem, which has also been called *the fundamental theorem for finite distributive lattices*, states that the elements of any such lattice $\mathcal{L}$ are isomorphic to the closed sets of a partial order, say Π. We generalize this theorem to showing that each sublattice of $\mathcal{L}$ is isomorphic to a distinct partial order that can be obtained from Π via the operation of *compression*, defined in Chapter 4.

   Let A be an instance of stable matching, with $\mathcal{L}$ being its lattice of stable matchings, and let B be the instance obtained by permuting the preference list of any one boy or any one girl. Let $\mathcal{M}_A$ and $\mathcal{M}_B$ be their sets of stable matchings. Our results are the following:

   - We show that $\mathcal{M}_A \cap \mathcal{M}_B$ is a sublattice of $\mathcal{L}$ and $\mathcal{M}_A \setminus \mathcal{M}_B$ is a semi-sublattice of $\mathcal{L}$.
   - Using our generalization of Birkhoff's Theorem, we give an efficient algorithm for finding the compression of Π that is isomorphic to the lattice of $\mathcal{M}_A \cap \mathcal{M}_B$.
   - Given a polynomial sized domain $\mathcal{D}$ of such errors (of permuting one of the preference lists), we give an efficient algorithm that checks if there is a stable matching for A that is stable for each such resulting instance B. We call this a *fully robust stable matching*.
   - If the answer is yes, the set of all such matchings forms a sublattice of $\mathcal{L}$ and our algorithm finds its partial order as well.

CHAPTER 1

INTRODUCTION AND BACKGROUND

The stable matching problem has attracted great interest for computer sciencetists, mathematicians and economists ever since its introduction in 1962 in a seminal paper of Gale and Shapley [1]. In the setting, there are a set of boys and a set of girls, where each agent has a total order perference over an agent of opposite sex. The problem is then to devise an matching of boys to girls in a way which takes into account their perferences. The notion of interest here is *stability*. A matching is said to be *stable* if it cannot be undermined by some unmatched pair. To be precise, there is no pair such that both agents in the pair prefer each other than their partners.

On the theoretical side, the problem acquires an elegant and profound mathematics structure. A fundamental result, due to Gale and Shapley, is that there always exists a stable matching in any stable matching instance. In fact, they gave an algorithm which yeilds a unique stable matching where each boy is matched to the best partner he can have in any stable matching. Such a matching is called *boy-optimal*. Obviously, one can change the role of boys and girls to obtain the *girl-optimal* matching. The set of stable matchings forms a distributive lattice where the boy-optimal and girl-optimal represent minimum and maximum elements of the lattice. From each matching in the lattice other than the girl-optimal matching, we can get one of its direct predecessors by applying a *rotation*. The set of rotations forms a poset whose closed subsets correspond to stable matchings. More details are explained later in this chapter.

On the practical side, stable matching problem is one of the rare instances where fancinating exercise in pure mathematics can be applied to real world situations. The applications of stable matching and its variations range from college admission, hospital residents, kidney exchange to market designs. In fact, the 2012 Nobel Prize in Economics was awarded to Lloyd S. Shapley and Alvin E. Roth "for the theory of stable allocations and the practice of market design."

In this thesis we present several new results on structural and algorithmic aspects of stable matching. Specifically,

- We generalize stable matching to maximum weight stable matching (Chapter 2).
- We introduce robust stable matching problem and solve it in the case where the input errors are shifts (Chapter 3).
- We generalize Birkhoff's theorem for distributive lattices, and apply it to find a robust stable matchings (Chapter 4).

1.1 Background

1.1.1 The stable matching problem

The stable matching problem takes as input a set of boys $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$ and a set of girls $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$; each person has a complete preference ranking over the set of opposite sex. The notation $b_i <_g b_j$ indicates that girl g strictly prefers b_j to b_i in her preference list. Similarly, $g_i <_b g_j$ indicates that the boy b strictly prefers g_j to g_i in his list.

A matching M is a one-to-one correspondence between $\mathcal{B}$ and $\mathcal{G}$. For each pair $bg \in M$, b is called the partner of g in M (or M-partner) and vice versa. For a matching M, we say that b is *above* (or *below*) g if he prefers his M-partner to g (or g to his M-partner). Similarly, g is said to be *above* (or *below*) b if she prefers her M-partner to b (or b to her M-partner). For a matching M, a pair $bg \notin M$ is said to be *blocking* if b is below g and g is below b, i.e., they prefer each other to their partners. A matching M is *stable* if there is no blocking pair for M.

1.1.2 The lattice of stable matchings

Let M and M' be two stable matchings. We say that M *dominates* M', denoted by $M \preceq M'$, if every boy weakly prefers his partner in M to M'. It is well known that the dominance partial order over the set of stable matchings forms a distributive lattice [2], with meet and join defined as follows. The *meet* of M and M', $M \wedge M'$, is defined to be the matching that results when each boy chooses his more preferred partner from M and M'; it is easy to show that this matching is also stable. The *join* of M and M', $M \vee M'$, is defined to be the matching that results when each boy chooses his less preferred partner from M and M'; this matching is also stable. These operations distribute, i.e., given three stable matchings M, M', M'',

$$M \vee (M' \wedge M'') = (M \wedge M') \vee (M \wedge M'') \text{ and } M \wedge (M' \vee M'') = (M \vee M') \wedge (M \vee M'').$$

It is easy to see that the lattice must contain a matching, M_0, that dominates all others and a matching M_z that is dominated by all others. M_0 is called the *boy-optimal matching*, since in it, each boy is matched to his most favorite girl among all stable matchings. This is also the *girl-pessimal matching*. Similarly, M_z is the *boy-pessimal* or *girl-optimal matching*.

1.1.3 Rotations help traverse the lattice

A crucial ingredient needed to understand the structure of stable matchings is the notion of a rotation, which was defined by Irving [3] and studied in detail in [4]. A rotation takes r matched pairs in a fixed order, say $\{b_0g_0, b_1g_1, \ldots, b_{r-1}g_{r-1}\}$ and "cyclically" changes the mates of these $2r$ agents, as defined below, to arrive at another stable matching. Furthermore, it represents a minimal set of pairings with this property, i.e, if a cyclic change is applied on any subset of these r pairs, with any ordering, then the resulting matching has a blocking pair and is not stable. After rotation, the boys' mates weakly worsen and the girls' mates weakly improve. Thus one can traverse from M_0 to M_z by applying a suitable sequence of rotations (specified by the rotation poset defined below). Indeed, this is precisely the purpose of rotations.

Let M be a stable matching. For a boy b let $s_M(b)$ denote the first girl g on b's list such that g strictly prefers b to her M-partner. Let $next_M(b)$ denote the partner in M of girl $s_M(b)$. A *rotation ρ exposed* in M is an ordered list of pairs $\{b_0g_0, b_1g_1, \ldots, b_{r-1}g_{r-1}\}$ such that for each i, $0 \leq i \leq r-1$, b_{i+1} is $next_M(b_i)$, where $i+1$ is taken modulo r. In this thesis, we assume that the subscript is taken modulo r whenever we mention a rotation. Notice that a rotation is cyclic and the sequence of pairs can be rotated. M/ρ is defined to be a matching in which each boy not in a pair of ρ stays matched to the same girl and each boy b_i in ρ is matched to $g_{i+1} = s_M(b_i)$. 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.

Lemma 1 ([2], Theorem 2.5.4)**.** *Every rotation appears exactly once in any sequence of elimination from M_0 to M_z.*

Let $\rho = \{b_0g_0, b_1g_1, \ldots, b_{r-1}g_{r-1}\}$ be a rotation. For $0 \leq i \leq r-1$, we say that ρ *moves b_i from g_i to g_{i+1}*, and *moves g_i from b_i to b_{i-1}*. If g is either g_i or is strictly between g_i and g_{i+1} in b_i's list, then we say that ρ *moves b_i below g*. Similarly, ρ *moves g_i above b* if b is b_i or between b_i and b_{i-1} in g_i's list.

1.1.4 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. If ρ' precedes ρ, we also say that ρ *succeeds* ρ'. If neither $\rho' \prec \rho$ nor $\rho' \succ \rho$, we say that ρ' and ρ are *incomparable* 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 2 ([2], Lemma 3.2.1)**.** *For any boy b and girl g, there is at most one rotation that moves b to g, b below g, or g above b. Moreover, if ρ_1 moves b to g and ρ_2 moves b from g then $\rho_1 \prec \rho_2$.*

Lemma 3 ([2], Lemma 3.3.2)**.** Π *contains at most $O(n^2)$ rotations and can be computed in polynomial time.*

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 correponding 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 and that Π *generates the lattice* $\mathcal{L}$ of all stable matchings of this instance.

Let S be a subset of the elements of a poset P, and let v be an element in S. We say that v is a *minimal* element in S if there is 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 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.

1.1.5 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 $x, y \in \mathcal{L}'$.

A *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 $x, y \in \mathcal{L}'$. We note that in the mathematics literature, two types of semilattices are defined: *meet semilattices* and *join semilattices*, which are sets that are closed under the meet and join operation, respectivey. For reasons of simplicity of notation, our definition of semi-sublattices has an asymmetry, since we only need those subsets of a lattice which are closed under the meet operation.

1.1.6 Robust Stable Matching

Let A be a stable matching instance, and let $\mathcal{D}$ be a discrete probability distribution over stable matching instances. A *robust stable matching* is a stable matching $M \in \mathcal{M}_A$ maximizing the probability that $M \in M_A \cap M_B$, where $B \sim \mathcal{D}$. We denote $x >_y^I x'$ if y prefers x to x' with respect to instance I. When the probability is 1, M is said to be a *fully robust stable matching*. In other words, $M \in \mathcal{M}_B$ for all B in the domain of $\mathcal{D}$.

1.2 Our results and contributions

1.2.1 Generalizing stable matching to maximum weight stable matching

The two problems of stable matching and cuts in graphs were introduced in the seminal papers of Gale and Shapley (1962) [1] and Ford and Fulkerson (1956) [5], respectively. Over the decades, remarkably deep and elegant theories have emerged around both these problems which include highly sophisticated efficient algorithms, not only for the basic problems but also several generalizations and variants, that have found numerous applications [6, 2, 7, 8].

We study a natural generalization of stable matching to the maximum weight stable matching problem and we obtain an efficient combinatorial algorithm for it; we remark that the linear programming formulation of stable matching [9, 10] can be used to show that the weighted version is in P. Our algorithm is obtained by reducing this problem to the problem of finding a maximum weight ideal cut in a DAG. We give the first polynomial time algorithm for the latter problem; this algorithm is also combinatorial. The combinatorial nature of our algorithms not only means that they are efficient but also that they enable us to obtain additional structural and algorithmic results:

- We show that the set, $\mathcal{M}'$, of maximum weight stable matchings forms a sublattice $\mathcal{L}'$ of the stable matching lattice $\mathcal{L}$.
- We give an efficient algorithm for finding boy-optimal and girl-optimal matchings in $\mathcal{M}'$.
- We generalize the notion of rotation, a central structural notion in the context of the stable matching problem, to *meta-rotation*. Analogous to the way rotations help traverse the lattice $\mathcal{L}$, meta-rotations help traverse the sublattice $\mathcal{L}'$.

The maximum weight stable matching problem has several applications. In 1987, Irving et. al. [11] gave a combinatorial polynomial time algorithm for the following problem which arose in the context of obtaining an *egalitarian stable matching* which, unlike the matching produced by the Gale-Shapley procedure, favors neither boys nor girls. Each boy b_i provides a preference weight $p(b_i, g_j)$ for each girl j and similarly, each girl g_i provides a preference weight $p(g_i, b_j)$ for each boy j. By ordering these weights, we get the preference orders for each boy and each girl. The problem is to find a matching that is stable under these preference orderings, say $(b_1, g_1), (b_2, g_2), \ldots, (b_n, g_n)$, such that it maximizes (or minimizes) $(\sum_i p(b_i, g_i) + \sum_i p(g_i, b_i))$. Clearly, this is a special case of our problem. Another application is: given a set D of desirable boy-girl pairs and a set U of undesirable pairs, find a stable matching that simultaneously maximizes the number of pairs in D and minimizes the number of pairs in U. This reduces to our problem by assigning each pair in D a weight of 1 and each pair in U a weight of -1.

Our results are based on deep properties of rotations and the manner in which closed sets in the rotation poset Π yield stable matchings in the lattice $\mathcal{L}$.

Problem definitions Let I denote an instance of the stable matching problem over sets B and G of n boys and n girls, respectively. Let $\boldsymbol{w}$ be a weight function $\boldsymbol{w} : \mathcal{B} \times \mathcal{G} \to \mathbb{Q}$. Then (I, w) defines an instance of the *maximum weight stable matching problem*; it asks for a stable matching of instance I, say M, that maximizes the objective function $\sum_{bg \in M} w_{bg}$.

In the *maximum weight ideal cut problem* we are given a directed acyclic graph $G = (V, E)$ with a source s and a sink t such that for each $v \in V$, there is a path from s to v and a path from v to t. We are also given a weight $w_{uv} \in \mathbb{Q}$ for each edge $(u, v) \in E$. An *ideal cut* is a partition of the vertices into sets S and $\overline{S} = V(G) \setminus S$ such that $s \in S$ and $t \in \overline{S}$ and there is no edge $uv \in E$ with $u \in \overline{S}$ and $v \in S$. We remark that such a set S is also called a *closed set*. The weight of the ideal cut $(S, \overline{S})$ is defined to be sum of weights of all edges crossing the cut i.e., $\sum_{uv: u \in S, v \notin S} w_{uv}$. The problem is to find an ideal cut of maximum weight.

Overview of results and technical ideas We start by giving an LP formulation for the problem of finding a maximum weight ideal cut in an edge-weighted DAG, G; we note that the weights can be positive as well as negative. We go on to showing that this LP always has integral optimal solutions, hence showing that the problem is in P (Proposition 3). We next study the polytope obtained from the constraints of this LP (Theorem 4). We first show that the set of vertices of this polytope is precisely the set of maximum weight ideal cuts in the DAG G. For this reason, we call this the ideal cut polyhedron. Next we characterize the edges of this polyhedron: we show that two cuts $(S, \overline{S})$ and $(S', \overline{S'})$ are adjacent in the polyhedron iff $S \subset S'$ or $S' \subset S$.

We then study the dual of this LP. We interpret it as solving a special kind of s-t flow problem in G in which the flow on each edge has to be a least the capacity of the edge and the objective is to minimize the flow from s to t. We show how to solve this flow problem combinatorially in polynomial time (Proposition 6). Next, we define the notion of a residual graph for our flow problem. After finding an optimal flow, the srongly connected components in the residual graph are shrunk to give an unweighted DAG D. We show that ideal cuts in D correspond to maximum weight ideal cuts in G (Theorem 5).

We also show that the set of maximum weight ideal cuts in G forms a lattice under the operations of set union and intersection (Theorem 5).

Finally, we move on to our main problem of finding a maximum weight stable matching. We start by showing that the set $\mathcal{M}'$ of such matchings forms a sublattice $\mathcal{L}'$ of the lattice $\mathcal{L}$ of all stable matchings (Theorem 9).

We then give what can be regarded as the main result of Chapter 2: a reduction from this problem to the problem of finding a maximum weight ideal cut in an edge-weighted DAG G (Section 2.3.1). This reduction goes deep into properties of rotations and the rotation poset Π. Closed sets of Π are in one-to-one correspondence with the stable matchings in the lattice $\mathcal{L}$. In particular, if matching M corresponds to closed set S, then starting from the boy-optimal matching in lattice $\mathcal{L}$ we will reach matching M by applying the set of rotations in S.

Let R be the set of rotations used in Π. We add new vertices s and t to Π; s dominates all remaining vertices and t is dominated by all remaining vertices. This yields the DAG G. The next task is to assign appropriate weights to the edges of G; this is done by using properties of rotations. Finally, let $(S, \overline{S})$ be a maximum weight ideal cut in weighted DAG G, and let M be the matching arrived at by starting from the boy-optimal matching in lattice $\mathcal{L}$ and applying the set of rotations in S. Then let us say that M corresponds to S. We show that in fact this is a one-to-one correspondence between maximum weight ideal cuts in G and maximum weight stable matchings for the given instance (Theorem 8).

Recall the definition of (unweighted) DAG D given above which was obtained from the edge-weighted DAG G. As stated above, ideal cuts in the D correspond to maximum weight ideal cuts in G, and hence to maximum weight stable matchings, $\mathcal{M}'$. A vertex in D corresponds to a set of vertices in G, and these sets form a partition of the set of rotations

R. We call these sets *meta-rotations*. As stated above, meta-rotations help traverse the sublattice $\mathcal{L}'$ in the same way that rotations help traverse the lattice $\mathcal{L}$ (Theorem 10).

1.2.2 Finding stable matchings that are robust to shifts

We initiate the study of stable matching problem from another angle, namely robustness to errors in the input. To the best of our knowledge, this issue has not been studied in the context of this problem even though the design of algorithms that produce robust solutions is already a very well established field, especially as pertaining to robust optimization, e.g., see the books [12, 13].

Our polynomial time algorithm for finding robust stable matchings follows from new structural properties related to the lattice of stable matchings. We remark that work by numerous researchers has revealed deep structural facts about this lattice, e.g., see the books mentioned above. Our work, of course, builds on many of these facts.

Consider the following situation: Alice has an instance A of the stable matching problem, over n boys and n girls, which she sends it to Bob over a channel that can introduce errors. Let B denote the instance received by Bob. Let D denote a polynomial sized domain from which errors are introduced by the channel; we will assume that the channel introduces at most one error from D. We are also given the discrete probability distribution, p over D, from which the channel picks one error. In addition, Alice sends to Bob a matching, M, of her choice, that is stable for instance A. Since M consists of only $O(n)$ numbers of $O(\log n)$ bits each, as opposed to A which requires $O(n^2)$ numbers, Alice is able to send it over an error-free channel. Now Alice wants to pick M in such a way that it has the highest probability of being stable in the instance received by Bob. Hence she picks M from the set

$$\arg\max_N \{Pr_p[N \text{ is stable for instance } B \mid N \text{ is stable for instance } A]\},$$

We will say that such a matching M is *robust*. We seek a polynomial time algorithm for finding such a matching.

Clearly, the domain of errors, D, will have to be well chosen to solve this problem. A natural set of errors is *simple swaps*, under which the positions of two adjacent boys in a girl's list, or two adjacent girls in a boy's list, are interchanged. We will consider a generalization of this class of errors, which we call *shift*. For a girl g, assume her preference list in instance A is $\{\ldots, b_1, b_2, \ldots, b_k, b, \ldots\}$. Move up the position of b so g's list becomes $\{\ldots, b, b_1, b_2, \ldots, b_k, \ldots\}$, and let B denote the resulting instance. Then we will say that B is obtained from A by a shift. An analogous operation is defined on a boy b's list. The domain D consists of all such shifts; clearly, D is polynomially bounded. We prove the following theorem.

Theorem 1. *There is a polynomial time algorithm which given an instance A of the stable matching problem and a probability distribution p over the domain, D, of errors defined above, finds a robust stable matching in A.*

Overview of results and technical ideas Let A and B be two instances of stable matching over n boys and n girls, with sets of stable matchings $\mathcal{M}_A$ and $\mathcal{M}_B$, respectively, and lattices $\mathcal{L}_A$ and $\mathcal{L}_B$, respectively. Then, it is easy to see that the matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ form a sublattice in each of the two lattices. Next assume that instance B results from applying a shift operation, defined above, to instance A. Then, we show that $\mathcal{M}_{AB} = \mathcal{M}_A - \mathcal{M}_B$ is also a sublattice of $\mathcal{L}_A$. We use this fact crucially to show that there is at most one rotation, ρ_{in}, that leads from $\mathcal{M}_{AB}$ to $\mathcal{M}_A \cap \mathcal{M}_B$ and at most one rotation, ρ_{out} that leads from $\mathcal{M}_A \cap \mathcal{M}_B$ to $\mathcal{M}_{AB}$. Moreover, we can obtain efficiently this pair of rotations for each of the polynomially many instances that result from the polynomially many shifts.

It is easy to see that a matching M corresponding to a closed set S is stable in instance B iff $\rho_{\text{in}} \in S$ and $\rho_{\text{out}} \notin S$. We next give an integer program whose optimal solution is a robust stable matching for the given probability distribution on shifts. The IP has one indicator variable, y_ρ, corresponding to each rotation ρ in Π. The constraints of the program ensure that the set S of rotations that are set to 0 form a closed set. The rest of the constraints and the objective function ensure that the corresponding matching maximizes the probability that it is stable in the erroneous instance B. Finally, we show that the LP-relaxation of this IP always has integral solutions. Hence we obtain a polynomial time algorithm for finding a robust stable matching.

1.2.3 A generalization of Birkhoffs theorem with applications to robust stable matchings

Birkhoff's theorem [14], which has also been called *the fundamental theorem for finite distributive lattices*, e.g., see [15], states that any such lattice is isomorphic to the closed sets of a partial order. It is easy to see that the latter form a distributive lattice with the join and meet operations being union and intersection, respectively. In this thesis, we state and prove a generalization of Birkhoff's theorem.

Let $\mathcal{L}$ denote a finite distributive lattice and let P be a partial order whose closed sets are isomorphic to $\mathcal{L}$. We define the operation of *compression of a partial order*, which yields another partial order. We prove that there is a one-to-one correspondence between the sublattices of $\mathcal{L}$ and compressions of P such that if $\mathcal{L}'$ is a sublattice of $\mathcal{L}$ and the compression of P corresponding to it is P', then $\mathcal{L}'$ is isomorphic to the closed sets of P'.

The theorem stated above was discovered in the context of the stable matching problem, which in turn is intimately connected to finite distributive lattices. Conway, see [6], proved that the set of stable matchings always forms a lattice, with the join and meet of two stable matchings being the operations of taking the boy-optimal choices and girl-optimal choices, respectively, of the two matchings. Knuth [6] asked if every finite distributive lattice is isomorphic to a stable matching lattice. A positive answer was provided by Blair [16]; for a much better proof, see [2].

In Section 1.2.2, we introduce the problem of finding stable matchings that are robust to errors introduced in the input. The domain, D, of errors is defined via an operation called *shift*. For a girl g, assume her preference list in instance A is $\{\ldots, b_1, b_2, \ldots, b_k, b, \ldots\}$. Move up the position of b so g's list becomes $\{\ldots, b, b_1, b_2, \ldots, b_k, \ldots\}$, and let B denote the resulting instance. An analogous operation is defined on a boy b's list. The domain D consists of all possible shifts for each girl and each boy.

Clearly, domain D is very restrictive and extending the domain is an open problem. Our attempt at extending the domain led us to seek deeper structural properties of the lattice of stable matchings which finally led to the generalization of Birkhoff's Theorem, a result that transcends the original application and is of independent interest in the theory of finite distributive lattices. Using this generalization, we extend the domain of errors to all permutations of the preference list of any girl or any boy, to find a fully robust stable matching, as defined below.

Let us state the main algorithmic result formally. Let A be a stable matching instance on n boys and n girls and let T denote the set of all possible instances B obtained by introducing one error of the following type in A: For any one girl or any one boy, permute of the preference list of the girl or the boy. Clearly $|T| = n^2 2^n$. Let $D \subset T$ be an arbitrary polynomial sized set. Define a *fully robust stable matching* to be a matching that is stable for A and for each of the instances in D. We prove the following.

Theorem 2. *For the setting given above, there is a polynomial time algorithm for checking if there is a fully robust stable matching. If the answer is yes, the set of all such matchings form a sublattice of $\mathcal{L}$ and our algorithm finds a compression of P that generates this sublattice.*

Overview of results and technical ideas

Generalizing Bhirkoff's Theorem

In Chapter 2, we gave a combinatorial polynomial time algorithm for: given a stable matching instance, I, and a weight function over all boy-girl pairs, find a maximum weight stable matching. We also showed that the set of maximum weight stable matchings form a sublattice $\mathcal{L}'$ of $\mathcal{L}$, the lattice of stable matchings for I, and we showed how to obtain a poset Π' from the poset Π of instance I, such that Π' generates the matchings of $\mathcal{L}'$. We observed that the elements of poset Π' are sets of rotations that partition the set of all rotations used in Π; however, at that point we did not have the notion of compression. This notion is introduced in Chapter 4; it arose in the process of seeking a better understanding in the relationship between Π and Π'.

We prove our generalization (Theorem 15) in the context of stable matching lattices since they are easier to handle because of the additional structural properties mentioned above. As remarked above, stable matching lattices are as general as arbitrary finite distributive lattices. Let $\mathcal{L}$ be a stable matching lattice which is generated by poset P. Our proof involves showing that each compression P_f of P generates a sublattice of $\mathcal{L}$ (Section 4.1.1), and corresponding to each sublattice $\mathcal{L}'$ of $\mathcal{L}$, there is a compression P_f of P that generates $\mathcal{L}'$ (Section 4.1.2).

The second part is quite non-trivial. It involves first identifying the correct partition of the set of rotations of P by considering pairs of matchings, $M,\ M'$ in $\mathcal{L}'$ such that M is a direct successor of the M', and obtaining the set of rotations that takes us from M' to M. This set will be a meta-rotation for P_f. Consider one such meta-rotation X. To obtain all predecessors of X in P_f, consider all paths that go from the boy-optimal matching in $\mathcal{L}$ to the girl-optimal matching by going through the lattice $\mathcal{L}'$. Find all meta-rotations that *always* occur before X does on all such paths. Then each of these meta-rotations precedes X. These are the precedence relations between meta-rotations in P_f.

A second definition of compression: Having derived our generalization of Birkhoff's Theorem using the definition alluded to above, we present a different, equivalent, definition of compression (Section 4.2). This definition is in terms of a set of directed edges, E, that needs to be added to P to yield, after some prescribed operations, the desired partial order P_f. Let $\mathcal{L}'$ be the sublattice generated by P_f. Then we will say that edges E *define* $\mathcal{L}'$.

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, there is no one-to-one correspondence between sublattices of $\mathcal{L}$ and the sets of edges that can be added to Π to yield compressions. Hence this definition is not suitable for proving the generalization of Birkhoff's Theorem.

Application to robust stable matchings

We start by giving a short overview of the structural facts proven in [17]. Let A and B be two instances of stable matching over n boys and n girls, 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 that is isomorphic to $\mathcal{L}_A$. It is easy to see that the matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ form a sublattice in each of the two lattices. For an instance B that results from applying a shift operation, [17] show that $\mathcal{M}_{AB} = \mathcal{M}_A \setminus \mathcal{M}_B$ is also a sublattice of $\mathcal{L}_A$. Using this fact, they show that there is at most one rotation, ρ_{in}, that leads from $\mathcal{M}_A \cap \mathcal{M}_B$ to $\mathcal{M}_{AB}$ and at most one rotation, ρ_{out} that leads from $\mathcal{M}_{AB}$ to $\mathcal{M}_A \cap \mathcal{M}_B$; moreover, these rotations can be efficiently found. Furthermore, a closed set S of Π generates a matching that is stable for instance B iff whenever $\rho_{\text{in}} \in S$, $\rho_{\text{out}} \in S$.

In order to extend the domain of errors, let us start by isolating out the essential structural fact stated above, namely, lattice $\mathcal{L}_A$ can be partitioned into two sublattices $\mathcal{L}_1$ and $\mathcal{L}_2$ (Section 4.3). A natural question then is: Assume we are given an oracle which given a matching $M \in \mathcal{L}_A$, tells us whether $M \in \mathcal{L}_1$ or $M \in \mathcal{L}_2$. Is there a polynomial time algorithm for finding a matching in $\mathcal{L}_1$?

Using our generalization of Birkhoff's Theorem, we first give a characterization of the set of edges E_1 and E_2 that define $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively (Theorem 18). Using this characterization, we prove that 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 $M \in \mathcal{L}_1$ iff it contains r_{2i} but not r_{2i+1} for some $0 \leq i \leq k$ (Proposition 19). Furthermore, this sequence of rotations can be found in polynomial time, hence giving an efficient algorithm for the question asked (we do not give details of this since it is subsumed by the more general case described next). However, so far we have been unable to find an error pattern, beyond shift, which when introduced in instance A yields B such that $\mathcal{M}_A \cap \mathcal{M}_B$ and $\mathcal{M}_{AB}$ partition lattice $\mathcal{L}_A$ into two sublattices.

Next, we address the case that $\mathcal{M}_{AB}$ 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 boy or any one girl, then $\mathcal{M}_{AB}$ must be a semi-sublattice of $\mathcal{L}_A$ (Lemma 42). Again, using our generalization of Birkhoff's Theorem, we obtain a (more elaborate) characterization of the set of edges that define the sublattice of $\mathcal{M}_A \cap \mathcal{M}_B$ (Theorem 20). Again, using this characterization, we give a (more elaborate) condition on rotations which is satisfied by a closed set of Π iff the corresponding matching is in the sublattice (Proposition 19). Furthermore, we show how to efficiently find these rotations (Theorem 22), 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 exponential set of all possible erroneous instances obtained by permuting the preference list of one boy or one girl, and $D \subset T$ a polynomial sized set of instances which the algorithm needs to consider. We show that the set of all such matchings that are stable for A and for each of these instances in D forms a sublattice of $\mathcal{L}$ and we obtain the compression of Π that generates this sublattice (Section 4.6.2). Each matching in this sublattice is a fully robust stable matching. Moreover, since we have obtained the poset generating it, we can go further: given a weight function on all boy-girl pairs, we can obtain, using the algorithm of [18], a matching that optimizes (maximizes or minimizes) the weight among all fully robust stable matchings.

CHAPTER 2

MAXIMUM WEIGHT STABLE MATCHING SOLVED VIA NEW INSIGHTS INTO IDEAL CUTS

In this chapter we study a natural generalization of stable matching to the maximum weight stable matching problem. We obtain a combinatorial polynomial time algorithm by reducing it to the problem of finding a maximum weight ideal cut in a DAG. The combinatorial algorithm for the later enables us to obtain additional structural results.

2.1 Maximum Weight Ideal Cuts: IP, LP and Polyhedron

In this section, we show how to find a maximum weight ideal cut using linear programming. We also prove some characteristics of the solution set and define a polyhedron whose vertices are precisely the ideal cuts.

2.1.1 A linear program for maximum weight ideal cut

Consider the following integer program which has a variable y_v for each vertex v of DAG $G = (V, E)$:

$$\begin{aligned} \max \quad & \sum_{uv \in E} w_{uv} \, (y_v - y_u) & \\ \text{s.t.} \quad & y_v \geq y_u & \forall e = uv \in E \\ & y_t = 1 & \\ & y_s = 0 & \\ & y_v \in \{0, 1\} & \forall v \in V. \end{aligned} \tag{2.1}$$

Lemma 4. *An optimal solution to (IP) is a maximum weight ideal cut in G.*

Proof. Let $S = \{v : y_v = 0\}$. The set of constraints

$$y_v \geq y_u \quad \forall e = uv \in E$$

guarantees that there are no edges coming into S. Hence, S forms an ideal cut. For each edge $e = uv \in E$,

$$y_v - y_u = \begin{cases} 1 & \text{if } u \in S \text{ and } v \notin S, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore,

$$\sum_{e \in E} w_e \left(y_v - y_u\right) = \sum_{uv: u \in S, v \notin S} w_{uv} = \sum_{e \text{ cross } S} w_e.$$

Thus, (IP) finds an ideal cut that maximizes the sum of weights of crossing edges as desired.

□

Now consider the following LP relaxation of (IP):

$$\begin{aligned} \max \quad & \sum_{uv \in E} w_{uv} \left(y_v - y_u\right) \\ \text{s.t.} \quad & y_v \geq y_u \qquad\qquad \forall e = uv \in E \\ & y_t = 1 \\ & y_s = 0. \end{aligned} \tag{2.2}$$

Note that the above constraints imply $0 = y_s \leq y_v \leq y_t = 1$ for each $v \in V$ since there is a directed path from s to v and a directed path from v to t. We show how to round a solution of (2.2) to an integral solution with the same objective function value. Later on we show that any basic feasible solution of (2.2) is integral anyway.

Let $\boldsymbol{y}$ be a (fractional) optimal solution of (2.2) and $\boldsymbol{y}^*$ be an integral solution such that

$$y_v^* = \begin{cases} 1 & \text{if } y_v > 0, \\ 0 & \text{if } y_v = 0. \end{cases}$$

Lemma 5. *$\boldsymbol{y}^*$ has the same objective value as $\boldsymbol{y}$.*

Proof. Assume that $\boldsymbol{y}$ is not integral, since otherwise the statement is trivially true. We will say that y_v is the potential of v. Now there must exist $v \in V$ such that $0 < y_v = a < 1$. Denote S_a by the set of all vertices having potential a. Let E_{in} be the set of edges going into S_a:

$$E_{\text{in}} = \{uv \in E : u \notin S_a, v \in S_a\}$$

and E_{out} be the set of edges going out of S_a:

$$E_{\text{out}} = \{uv \in E : u \in S_a, v \notin S_a\}.$$

Claim. $\sum_{e \in E_{\text{in}}} w_e = \sum_{e \in E_{\text{out}}} w_e$.

Consider adding to the potentials of all vertices in S_a an amount δ where $|\delta|$ is small enough so that no constraint is violated. Specifically, the potential of $v \in S_a$ after modification is $y'_v = y_v + \delta$. The change in objective function along edges in E_{in} is

$$\sum_{uv \in E_{\text{in}}} w_{uv}(y'_v - y_u) - \sum_{uv \in E_{\text{in}}} w_{uv}(y_v - y_u) = \sum_{uv \in E_{\text{in}}} w_{uv}(y'_v - y_v) = \sum_{uv \in E_{\text{in}}} w_{uv}\delta.$$

The change in objective function along edges in E_{out} is

$$\sum_{uv \in E_{\text{out}}} w_{uv}(y_v - y'_u) - \sum_{uv \in E_{\text{out}}} w_{uv}(y_v - y_u) = \sum_{uv \in E_{\text{out}}} w_{uv}(y_u - y'_u) = -\sum_{uv \in E_{\text{out}}} w_{uv}\delta.$$

The total change is

$$\sum_{uv \in E_{\text{in}}} w_{uv}\delta - \sum_{uv \in E_{\text{out}}} w_{uv}\delta = \delta \left(\sum_{uv \in E_{\text{in}}} w_{uv} - \sum_{uv \in E_{\text{out}}} w_{uv} \right).$$

If $\sum_{e \in E_{\text{in}}} w_e \neq \sum_{e \in E_{\text{out}}} w_e$, we can always pick a sign for δ so as to obtain a strictly better solution. Therefore, $\sum_{e \in E_{\text{in}}} w_e = \sum_{e \in E_{\text{out}}} w_e$.

Let a' be the smallest y-value that is greater than a. $\sum_{e \in E_{\text{in}}} w_e = \sum_{e \in E_{\text{out}}} w_e$ implies that we can increase the potentials of all vertices in S_a to a' and obtain the same objective value. The theorem follows by induction on the number of possible y-values. $\square$

Lemma 4 and Lemma 5 give:

Proposition 3. *A maximum weight ideal cut can be found in polynomial time.*

2.1.2 The ideal cut polytope

Consider the polyhedron P formed by the constraints on $\boldsymbol{y}$ in (2.2):

$$\begin{aligned} & y_v \geq y_u \quad \forall e = uv \in E \\ & y_t = 1 \\ & y_s = 0. \end{aligned}$$

Let n be the number of vertices in G. A vertex of P is a feasible solution having at least n linearly independent active constraints (constraints that are satisfied at equality). Let A be the set of those constraints. Notice that in any feasible solution, $y_s = 0$ and $y_t = 1$ must be active. Let G_a be a graph such that $V(G_a) = V(G)$ and

$$E(G_a) = \{e : \text{ the constraint corresponding to } e \text{ is in } A.\}$$

We call G_a an *active graph.*

Lemma 6. *G_a consists of two trees $T_1 \ni s$ and $T_2 \ni t$ such that $V(T_1) \cup V(T_2) = V(G)$ and $V(T_1) \cap V(T_2) = \emptyset$.*

Proof. We prove that G_a contains no cycle and no $s-t$ path. Since each edge of G_a corresponds to a constraint in A, G_a has at least $n-2$ edges. The lemma will then follow.

Claim. G_a contains no cycle.

Assume G_a contains cycle $(v_0, v_1 \ldots, v_k)$. Since edges in the cycle correspond to active constraints, $y_{v_i} = y_{v_j}$ for each edge $v_i v_j$ in the cycle. Therefore, $y_{v_0} = y_{v_1}, y_{v_1} = y_{v_2}, \ldots, y_{v_{k-1}} = y_{v_k}$, which implies $y_{v_0} = y_{v_k}$. It follows that the set of inequalities are not independent.

Claim. G_a contains no $s-t$ path.

Assume G_a contains a path $(s, v_0 \ldots, v_k, t)$. Since edges in the path correspond to active constraints, $y_{v_i} = y_{v_j}$ for each edge $v_i v_j$ in the path. Therefore, $y_s = y_{v_0} = \ldots = y_{v_k} = y_t$, which is a contradiction.

$\square$

An edge in polytope P is defined by the intersection of $n-1$ linearly independent inequalities. Two vertices, also called basic feasible solutions, of the polytope are *neighbors* if and only if they share an edge, i.e., the sets of inequalities that define them differ in only one inequality. Two cuts are said to be neighbors if two basic feasible solutions corresponding to them are neighbors.

Theorem 4. *All vertices of polyhedron P are integral, and the set of vertices is precisely the set of ideal cuts. Moreover, vertices of P corresponding to cuts $(S, \overline{S})$ and $(S', \overline{S'})$ are neighbors if and only if $S \subset S'$ or $S' \subset S$.*

Proof. By Lemma 6, G_a consists of two non-intersecting trees $T_1 \ni s$ and $T_2 \ni t$. So

$y_v = y_s = 0$ for all $v \in T_1$ and $y_u = y_t = 1$ for all $u \in T_2$. Therefore, $\boldsymbol{y}$ is integral.

Now consider an ideal cut defined by S. We can find a tree T_1 connecting all vertices in S, and a tree T_2 connecting all vertices in $V(G_a) \setminus S$. Consider the following set of inequalities:

1. $|S| - 1$ constraints corresponding to edges in tree T_1,
2. $n - |S| - 1$ constraints corresponding to edges in tree T_2,
3. $y_t = 1$ and $y_s = 0$.

Clearly, the set contains n linearly independent inequalities, and the basic feasible solution obtained by those inequalities is exactly the ideal cut $(S, \overline{S})$.

Next, we prove the second statement. If the cuts defined by S and S' are neighbors, the sets of inequalities defining them differ in only one inequality. Let G_a and G'_a be active graphs for S and S' respectively. By Lemma 6, G_a consists of two trees T_1, T_2, and G'_a consists of two trees T'_1, T'_2. Moreover, $V(T_1) \cup V(T_2) = V(T'_1) \cup V(T'_2) = V(G)$ and $V(T_1) \cap V(T_2) = V(T'_1) \cap V(T'_2) = \emptyset$. Since the sets of inequalities defining S and S' differ in only one inequality, G'_a results from G_a by removing an edge e_1 and adding an edge e_2. Consider the graph G' obtained by removing e_1 from G_a. Without loss of generality, assume that $e_1 \in E(T_1)$. Therefore, there exists $X \subset V(T_1)$ such that vertices in X are not reachable from s in G'. By the proof of Lemma 6, G'_a contains no cycle. Hence, e_2 can either connect X to a vertex in $V(T_1) \setminus X$ or a vertex in $V(T_2)$. If e_2 connects X to a vertex in $V(T_1)$, $S = V(T_1) = V(T'_1) = S'$, which contradicts the fact that S and S' are neighbors. If e_2 connects X to a vertex in $V(T_2)$, we have $S' = V(T'_1) = V(T_1) \setminus X \subset V(T_1) = S$.

On the other direction, assume that $S' \subset S$ without loss of generality. We will give a set of active inequalities defining S and a set of active inequalities defining S' such that they differ in only one inequality. Let $X = S \setminus S'$. Let $T_S, T_X, T_{\overline{S'}}$ be spanning trees of S, X and $V(G) \setminus S'$ respectively. Let v be a vertex in X. By assumption on G, there exists a path

Q from s to t containing v. Since S and S' are ideal cuts, there exist an edge $e_1 \in Q$ from S to X and an edge $e_2 \in Q$ from X to $V(G) \setminus S'$. Consider the set of inequalities for edges in $E(T_S) \cup E(T_X) \cup E(T_{\overline{S'}}) \cup \{e_2\}$. These inequalities define S. Similarly, the inequalities for edges in $E(T_S) \cup E(T_X) \cup E(T_{\overline{S'}}) \cup \{e_1\}$ define S'. The two sets of inequalities differ by only one inequality as desired. $\square$

Theorem 4 justifies calling the polytope defined in this section *the ideal cut polytope*.

2.2 Maximum Weight Ideal Cuts: Combinatorial Algorithm

2.2.1 The set of maximum weight ideal cuts forms a lattice

We first prove the following fact.

Lemma 7. *If S and S' are two subsets defining maximum weight ideal cuts in G then $S \cup S'$ and $S \cap S'$ also define maximum weight ideal cuts.*

Proof. Let E_1 be the set of edges going from $S \cap S'$ to $S \setminus S'$:

$$E_1 = \{uv \in E : u \in S \cap S', v \in S \setminus S'\}.$$

Let E_2 be the set of edges going from $S \cap S'$ to $S' \setminus S$:

$$E_2 = \{uv \in E : u \in S \cap S', v \in S' \setminus S\}.$$

Let E_3 be the set of edges going from $S \setminus S'$ to $V \setminus (S' \cup S)$:

$$E_3 = \{uv \in E : u \in S \setminus S', v \in V \setminus (S' \cup S)\}.$$

Let E_4 be the set of edges going from $S' \setminus S$ to $V \setminus (S' \cup S)$:

$$E_4 = \{uv \in E : u \in S' \setminus S, v \in V \setminus (S' \cup S)\}.$$

Let E_5 be the set of edges going from $S' \cap S$ to $V \setminus (S' \cup S)$:

$$E_5 = \{uv \in E : u \in S' \cap S, v \in V \setminus (S' \cup S)\}.$$

Note that there are no edges going between $S' \setminus S$ and $S \setminus S'$.

Therefore, the weight of the ideal cut defined by S is

$$w(S) = \sum_{e \in E_2} w_e + \sum_{e \in E_3} w_e + \sum_{e \in E_5} w_e.$$

The size of the ideal cut defined by S' is

$$w(S') = \sum_{e \in E_1} w_e + \sum_{e \in E_4} w_e + \sum_{e \in E_5} w_e.$$

Since both cuts are maximum weight ideal cuts,

$$\sum_{e \in E_2} w_e + \sum_{e \in E_3} w_e = \sum_{e \in E_1} w_e + \sum_{e \in E_4} w_e.$$

We will show that $\sum_{e \in E_1} w_e = \sum_{e \in E_3} w_e$ and $\sum_{e \in E_2} w_e = \sum_{e \in E_4} w_e$. Assume that $\sum_{e \in E_1} w_e < \sum_{e \in E_3} w_e$ and $\sum_{e \in E_2} w_e < \sum_{e \in E_4} w_e$. Then the weight of the cut defined by $S \cup S'$ is

$$w(S \cup S') = \sum_{e \in E_3} w_e + \sum_{e \in E_4} w_e + \sum_{e \in E_5} w_e > w(S).$$

Similarly, if $\sum_{e \in E_1} w_e > \sum_{e \in E_3} w_e$ and $\sum_{e \in E_2} w_e > \sum_{e \in E_4} w_e$,

$$w(S \cap S') = \sum_{e \in E_1} w_e + \sum_{e \in E_2} w_e + \sum_{e \in E_5} w_e > w(S).$$

Therefore, $\sum_{e \in E_1} w_e = \sum_{e \in E_3} w_e$, $\sum_{e \in E_2} w_e = \sum_{e \in E_4} w_e$ and

$$w(S \cup S') = w(S \cap S') = w(S) = w(S').$$

□

Lemma 7 gives:

Theorem 5. *The set of maximum weight ideal cuts forms a lattice under the operations of union and intersection.*

2.2.2 A flow problem in which capacities are lower bounds on edge-flows

To unveil the underlying combinatorial structure, we consider the dual program of (2.2). First, (2.2) can be rewritten as:

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e z_e \\ \text{s.t.} \quad & z_e = y_v - y_u \qquad \forall e = uv \in E \\ & y_t - y_s = 1 \\ & z_e \geq 0 \qquad\qquad \forall e \in E. \end{aligned} \tag{2.3}$$

while there exists an edge $uv \in E$ such that $w_{uv} > 0$ and $f_{uv} < w_{uv}$ **do**
1. Find a path Q from s to t containing uv.
2. Send flow of value w_{uv} along Q.

end while

Figure 2.1: Routine for Finding a Feasible Flow.

Let f_{uv} be the dual variable corresponding to edge uv. The dual linear program is:

$$\begin{aligned} \min \quad & f_{ts} \\ \text{s.t.} \quad & \sum_{u:uv \in E} f_{uv} = \sum_{u:vu \in E} f_{vu} && \forall v \in V \\ & f_{uv} \geq w_{uv} && \forall uv \in E \end{aligned} \tag{2.4}$$

We show that (2.4) can be interpreted as a flow problem. To be precise, f_{uv} represents the flow value on edge uv. The first set of inequalities guarantees flow conservation at each vertex. The second set of inequalities says that there is a lower bound w_{uv} on the amount of flow on uv. Note that f_{uv} as well as w_{uv} can be negative.

The problem is to find a minimum circulation in the graph obtained by adding an infinite capacity edge from t to s to G. Equivalently, without the introduction of ts, the problem can be seen as finding a minimum flow from s to t in G.

We give a combinatorial algorithm to solve the above flow problem. The high level idea is to first find a feasible flow, i.e, a flow $\boldsymbol{f}$ satisfying all inequalities. We then push flow back as much as possible from t to s, while maintaining flow feasibility.

It is easy to see that the routine in Figure 2.1 gives us a feasible flow. At the end of the routine, the value of flow from s to t is at most nW where $W = \max_e |w_e|$.

To push flow back from t to s, we construct the following residual graph $G_{\boldsymbol{f}}$ for a feasible flow $\boldsymbol{f}$. Since $\boldsymbol{f}$ is feasible, $f_{uv} \geq w_{uv} \, \forall uv \in E$. For each $uv \in E$ such that $f_{uv} > w_{uv}$, we create a residual edge from v to u with capacity $f_{uv} - w_{uv} > 0$. Notice that the capacity on vu is exactly the amount we can push back on uv without violating the lowerbound constraint. Finally, all edges in E still have infinite capacity.

Let $\boldsymbol{x} > 0$ be a feasible flow in $G_{\boldsymbol{f}}$. In other words, $\boldsymbol{x}$ satisfies flow conservation and capacity constraints. Let $\overline{\boldsymbol{f}} = \boldsymbol{f} \oplus \boldsymbol{x}$ be a flow constructed as follows:

$$\overline{f}_{uv} = \begin{cases} f_{uv} + x_{uv} - x_{vu} & \text{if } vu \text{ is an edge in } G_{\boldsymbol{f}}, \\ f_{uv} + x_{uv} & \text{otherwise.} \end{cases} \tag{2.5}$$

Lemma 8. $\overline{\boldsymbol{f}}$ *is a feasible solution to (2.4).*

Proof. Flow conservation is satisfied trivially. It suffices to show that no lower bound constraint is violated. Consider 2 cases:

- if vu is an edge in $G_{\boldsymbol{f}}$, the capacity of vu is $f_{uv} - w_{uv}$. Therefore, $f_{uv} + x_{uv} - x_{vu} \geq f_{uv} - (f_{uv} - w_{uv}) = w_{uv}$.
- if vu is not an edge in $G_{\boldsymbol{f}}$, $f_{uv} + x_{uv} \geq f_{uv} = w_{uv}$.

□

Lemma 9. $\boldsymbol{f}$ *is an optimal solution of (2.4) if and only if there is no path from t to s in $G_{\boldsymbol{f}}$.*

Proof. Suppose that there exists a path from t to s in $G_{\boldsymbol{f}}$. Sending flow along the path gives a feasible flow by Lemma 8. Moreover, the objective function has a smaller value. Therefore, $\boldsymbol{f}$ is not an optimal solution of (2.4).

If there is no path from t to s, let T be the set of vertices that are reachable from t by a path in $G_{\boldsymbol{f}}$:

$$T = \{v \in V : \exists \text{ path } p \text{ from } t \text{ to } v \text{ in } G_{\boldsymbol{f}}\}.$$

Consider $uv \in E$ such that $v \in T$ and $u \notin T$. Since vu is not an edge in $G_{\boldsymbol{f}}$ and $\boldsymbol{f}$ is feasible, $f_{uv} = w_{uv}$.

1. Find a feasible flow $\boldsymbol{f}$.
2. Find a maximum flow $\boldsymbol{x}$ from t to s in $G_{\boldsymbol{f}}$.
3. Return $\overline{\boldsymbol{f}} = \boldsymbol{f} \oplus \boldsymbol{x}$ as shown in 2.5.

Figure 2.2: Combinatorial Algorithm for Finding Flow.

Let $\boldsymbol{y}$ be the primal solution such that $y_v = 1$ for all $v \in T$ and $y_v = 0$ otherwise. With respect to $\boldsymbol{y}$, $z_{uv} > 0$ if and only if $v \in T$ and $u \notin T$ if and only if $f_{uv} = w_{uv}$. Therefore, $\boldsymbol{f}$ and $\boldsymbol{y}$ satisfy complementarity. Hence, $\boldsymbol{f}$ is an optimal solution of (2.4). $\square$

By Lemma 8 and Lemma 9, a natural algorithm, given a feasible flow $\boldsymbol{f}$, is the following: Iteratively find a path from t to s in $G_{\boldsymbol{f}}$. If there exists such a path, send maximal flow back on this path without violating feasibility, update $\boldsymbol{f}$ and repeat. Otherwise, $\boldsymbol{f}$ is an optimal flow by Lemma 9.

Notice that the above routine is very similar to the FordFulkerson algorithm for finding maximum s-t flow. A more straight forward way is to compute a maximum flow in $G_{\boldsymbol{f}}$ for a feasible flow $\boldsymbol{f}$ as shown in Figure 2.2.

Proposition 6. *The algorithm in Figure 2.2 finds an optimal flow for (2.4).*

Proof. By Lemma 9, it suffices to show that there is no path from t to s in $G_{\overline{\boldsymbol{f}}}$ if and only if $\boldsymbol{x}$ is a maximum flow from t to s in $G_{\boldsymbol{f}}$.

If there exists a path from t to s in $G_{\overline{\boldsymbol{f}}}$, then there exists $\boldsymbol{x}'$ such that $\overline{\boldsymbol{f}} \oplus \boldsymbol{x}' = (\boldsymbol{f} \oplus \boldsymbol{x}) \oplus \boldsymbol{x}' = \boldsymbol{f} \oplus (\boldsymbol{x} + \boldsymbol{x}')$ is a flow of from s to t smaller value than $\overline{\boldsymbol{f}}$. Therefore, $\boldsymbol{x} + \boldsymbol{x}'$ is a flow from t to s in $G_{\boldsymbol{f}}$ of greater value than $\boldsymbol{x}$, which is a contradiction.

If $\boldsymbol{x}$ is not a maximum flow, there exists $\boldsymbol{x}'$ such that $\boldsymbol{x} + \boldsymbol{x}'$ is a feasible flow from t to s in $G_{\boldsymbol{f}}$ of greater value. Therefore, there exists a path from t to s in $G_{\boldsymbol{f} \oplus \boldsymbol{x}}$. $\square$

2.2.3 Generating all maximum weight ideal cuts

The process is similar to finding the Picard-Queyranne structure, whose ideal cuts are in one-to-one correspondence with the minimum s-t cuts in a graph. Given an optimal flow solution $\boldsymbol{f}$, we shrink the strongly connected components of $G_{\boldsymbol{f}}$. The resulting graph is a DAG D. Now, ideal cuts in D are in one-to-one correspondence with maximum weight cuts in the original graph. Hence we get:

Theorem 7. *There is a combinatorial polynomial time algorithm for constructing a DAG D such that an ideal cut in D bijectively corresponds to a maximum weight ideal cut in G.*

2.3 Maximum Weight Stable Matching Problem

2.3.1 The reduction

Given an instance I of maximum weight stable matching problem, we show how to obtain an instance J of maximum weight ideal cut problem such that there is a bijection between the set of solutions to I and those to J.

For this purpose, we show how to construct a DAG G with an edge-weight function w. We start with the rotation poset Π that generates all stable matchings for I. This can be obtained in polynomial time by Lemma 3. Next, we construct an edge-weighted DAG G as follows:

1. Keep all vertices and edges in the natural DAG representation of Π. Let v_i be the vertex that corresponds to ρ_i.

2. Add a source s and an edge from s to every v_i such that ρ_i is not dominated by any other rotation.

3. Add a sink t and an edge from every v_i to t, such that ρ_i does not dominate any other rotation.

Next, consider all pairs bg that appears in the stable matchings of the given instance. Ignore a pair bg if it appears in all stable matchings. With each of the remaining pairs bg, we associate a directed path P_{bg} in G as follows:

- **Case 1,** $bg \in M_0, bg \notin M_z$: There exists a rotation ρ_i that moves b away from g. Choose P_{bg} to be an arbitrary path in G from s to v_i.
- **Case 2,** $bg \in M_z, bg \notin M_0$: There exists a rotation ρ_i that moves b to g. Choose P_{bg} to be an arbitrary path in G from v_i to t.
- **Case 3,** $bg \notin M_0, bg \notin M_z$: There exist a rotation ρ_i moving b to g and a rotation ρ_j moving b from g. By Lemma 2, ρ_i dominates ρ_j, and therefore there is at least one path in G from v_i to v_j. Choose P_{bg} to be an arbitrary such path.

Finally, we assign weights to the edges of G as follows. Initialize all edge weights to 0. Then, for each pair bg, we add w_{bg} to the weights of all edges in P_{bg}. We also say that $w_{P_{bg}} = w_{bg}$

Clearly, an ideal cut in G corresponds to a closed subset in Π. To be precise, for a non-empty vertex set S such that $s \in S$ and there are no incoming edges to S,

$$C = \{\rho_i : v_i \in S \setminus \{s\}\}$$

is clearly a closed subset in Π. We prove a simple yet crucial lemma:

Lemma 10. *S cuts P_{bg} if and only if the matching generated by C contains bg.*

Proof. We will use the following key observation: for any pair u, v of vertices in a DAG such that there exist paths from u to v, an ideal cut separates u and v if and only if it cuts each of these paths exactly one. We consider 3 cases:

- **Case 1,** $bg \in M_0, bg \notin M_z$: There exists a unique rotation ρ_i that moves b away from g. S cuts P_{bg} if and only if C does not contain ρ_i. This happens if and only if the matching generated by C contains bg.

- **Case 2,** $bg \notin M_0, bg \in M_z$: There exists a unique rotation ρ_i that moves b to g. S cuts P_{bg} if and only if C contains ρ_i. This happens if and only if the matching generated by C contains bg.

- **Case 3,** $bg \notin M_0, bg \notin M_z$: There exist a unique rotation ρ_i moving b to g and a unique rotation ρ_j moving b from g. S cuts P_{bg} if and only if C contains ρ_i and does not contain ρ_j. This happens if and only if the matching generated by C contains bg.

□

Theorem 8. *The maximum weight stable matchings in I are in one-to-one correspondence with the maximum weight ideal cuts in J.*

Proof. We show that the weight of an ideal cut generated my S is equal to the weight of the matching generated by C. By Lemma 10,

$$\begin{aligned} w(S) &= \sum_{e=uv:u\in S, v\notin S} w_e = \sum_{e=uv:u\in S, v\notin S} \sum_{e\in P_{bg}} w_{bg} \\ &= \sum_{S \text{ cuts } P_{bg}} w_{bg} = \sum_{bg \in \text{ the matching generated by } C} w_{bg}. \end{aligned}$$

The theorem follows. □

2.3.2 The sublattice, and using meta-rotations to traversing it

By Theorem 5 and Theorem 8 we get:

Lemma 11. *If M and M' are maximum stable matchings in $\mathcal{M}$ then so are $M \vee M'$ and $M \wedge M'$.*

This gives:

Theorem 9. *The set of maximum weight stable matchings forms a sublattice $\mathcal{L}'$ of the lattice $\mathcal{L}$.*

We next give the notion of a meta-rotation. These help traverse the sublattice $\mathcal{L}'$ in the same way that rotations help traverse the lattice $\mathcal{L}$. Let R be the set of all rotations used in the rotation poset Π. Let G be the graph obtained from Π by adding vertices s and t and assigning weights to edges, as described in Section 2.3.1. Let D be the DAG constructed in Theorem 7; ideal cuts in D correspond to a maximum weight ideal cuts in G. A vertex, v, in D corresponds to a set of vertices in G. Hence we may view v as a subset of the rotations in R; clearly, the subsets represented by the set of all vertices in D form a partition of R.

By analogy with the rotation poset Π, let us represent D by $\overline{\Pi}$ and call it the *meta-rotation poset*. Each vertex in $\overline{\Pi}$ (and D) is a subset of R and is called a *meta-rotation*. Let S be the element in $\overline{\Pi}$ containing s, and T be the element in $\overline{\Pi}$ containing t. For any closed subset, P, of $\overline{\Pi}$, let R_P be the set of all rotations contained in the meta-rotations of P. Eliminating these rotations starting from M_0, according to the topological ordering of the rotations given in Π, we arrive at a maximum weight stable matching, say M_P. In this manner, the meta-rotations help us traverse the sublattice. Combining with Proposition 6 and Theorem 8, we get:

Theorem 10. *There is a combinatorial polynomial time algorithm for finding a maximum weight stable matching. The meta-rotation poset $\overline{\Pi}$ can also be constructed in polynomial time. Each closed subset of $\overline{\Pi}$ containing S and not containing T generates a maximum weight stable matching.*

The running time of the algorithm described above is dominated by the time required to find a max-flow in the graph obtained from Π which has $O(n^2)$ vertices.

2.3.3 Further applications of the structure

Finding boy-optimal and girl-optimal matchings in $\mathcal{L}'$

Notice that for two closed subsets C and C', the matching generated by C dominates the matching generated by C' if $C \subset C'$. Hence we have:

Lemma 12. *The closed subset containing only the meta-rotation S generates the boy-optimal stable matching and the one containing all meta-rotations other than T generates the girl-optimal stable matching in the sublattice $\mathcal{L}'$.*

Bi-objective stable matching

In the *bi-objective stable matching problem* we are given sets B and G, of n boys and n girls and, for each boy and each girl, a complete preference ordering over all agents of the opposite sex. However, unlike the maximum weight stable matching problem, we are given two weight functions $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)} : \; B \times G \to \mathbb{R}$. The problem is to find a stable matching M that maximizes $\sum_{bg \in M} w_{bg}^{(2)}$ among the ones maximizing $\sum_{bg \in M} w_{bg}^{(1)}$.

To solve this problem, first we find a poset $\overline{\Pi}$ that generates the set of stable matchings maximizing $\sum_{bg \in M} w_{bg}^{(1)}$. Then we form a maximum ideal cut instance in the same way as described in section 2.3.1 with respect to $\boldsymbol{w}^{(2)}$. Let G be the DAG in the instance. For each meta-rotation V in $\overline{\Pi}$, contract all vertices in G corresponding to the rotations in V. Let $\overline{G}$ be the resulting graph. By a similar argument to the one in Section 2.3.1, we have:

Lemma 13. *The maximum weight ideal cuts in $\overline{G}$ are in one-to-one correspondence with the solutions of bi-objective stable matching problem.*

CHAPTER 3

FINDING STABLE MATCHINGS THAT ARE ROBUST TO SHIFTS

In this chapter we initiate the study of stable matching problem with respect to robustness to errors in the input. Our polynomial time algorithm for finding robust stable matchings follows from new structural properties related to the lattice of stable matchings. In this chapter we assume that B is an instance resulted from applying a shift on A.

3.1 Structural Results

3.1.1 The stable matchings in $\mathcal{M}_A \setminus \mathcal{M}_B$ form a sublattice

Let $\mathcal{M}_A$ and $\mathcal{M}_B$ be the sets of all stable matchings under instance A and B respectively. Let $\mathcal{M}_{AB} = \mathcal{M}_A \setminus \mathcal{M}_B$. In other words, $\mathcal{M}_{AB}$ is the set of stable matchings in A that become unstable in B. In this section we show that $\mathcal{M}_{AB}$ forms a lattice. We first prove a simple observation.

Lemma 14. *Let $M \in \mathcal{M}_{AB}$. The only blocking pair of M under instance B is bg.*

Proof. Since $M \notin \mathcal{M}_B$, there must be a blocking pair $xy \notin M$ under B. Assume xy is not bg, we will show that xy must also be a blocking pair in A. Let y' be the partner of x and x' be the partner of y in M. Since xy is a blocking pair in B, $x >_y^B x'$ and $y >_x^B y'$. The preference list of x remain unchanged from A to B, so $y >_x^A y'$. Next, we consider two cases:

- If y is not g, the preference list of y does not change. Therefore, $x >_y^A x'$, and hence, xy is also a blocking pair in A.

- If y is g, for all pairs x, x' such that $x >_y^B x'$ and $x \neq b$, we also have $x >_y^A x'$. Therefore, xy is a blocking pair in A.

This contradicts the fact that M is stable under A. $\square$

Recall that $b_1 \geq_g b_2 \geq_g \ldots \geq_g b_k$ are k boys right above b in g's list such that the position of b is shifted up to be above b_k in B. From Lemma 14, we can then characterize the set $\mathcal{M}_{AB}$.

Lemma 15. *$\mathcal{M}_{AB}$ is the set of all stable matchings in A that matches g to a partner between b_1 and b_k in g's list, and matches b to a partner below g in b's list.*

Proof. Assume M is a stable matching in A that contains $b_i g$ for $1 \leq i \leq k$ and bg' such that $g >_b g'$. In B, g prefers b to b_i, and hence bg is a blocking pair. Therefore, M is not stable under B and $M \in \mathcal{M}_{AB}$.

To prove the other direction, let M be a matching in $\mathcal{M}_{AB}$. By Lemma 14, bg is the only blocking pair of M in B. For that to happen, $p_M(b) <_b^B g$ and $p_M(g) <_g^B b$. We will show that $p_M(g) = b_i$ for $1 \leq i \leq k$. Assume not, then $p_M(g) <_g^B b_k$, and hence, $p_M(g) <_g^A b$. Therefore, bg is a blocking pair in A, which is a contradiction. $\square$

Let $\mathcal{L}_A$ be the boy-optimal lattice formed by $\mathcal{M}_A$.

Theorem 11. *$\mathcal{M}_{AB}$ forms a sublattice of $\mathcal{L}_A$.*

Proof. Assume $\mathcal{M}_{AB}$ is not empty. Let M_1 and M_2 be two matchings in $\mathcal{M}_{AB}$. By Lemma 15, M_1 and M_2 both match g to a partner between b_1 and b_k in g's list, and match b to a partner below g in b's list. Since $M_1 \wedge M_2$ is the matching resulting from having each boy choose the more preferred partner and each girl choose the least preferred partner, $M_1 \wedge M_2$ also belongs to the set characterized by Lemma 15. A similar argument can be applied to the case of $M_1 \vee M_2$. Therefore $\mathcal{M}_{AB}$ form a sublattice of $\mathcal{L}_A$. $\square$

3.1.2 Rotations going into and out of a sublattice

Let M be a stable matching in $\mathcal{M}_A$ and ρ be a rotation exposed in M. If $M \notin \mathcal{S}$ and $M/\rho \in \mathcal{S}$ for a set $\mathcal{S}$, we say that ρ *goes into* $\mathcal{S}$. Similarly, if $M \in \mathcal{S}$ and $M/\rho \notin \mathcal{S}$, we say that ρ *goes out of* S. Let the set of all rotations going into $\mathcal{S}$ and out of $\mathcal{S}$ be $I_{\mathcal{S}}$ and $O_{\mathcal{S}}$, respectively.

Lemma 16. *In $\mathcal{M}_A$, any rotation in $I_{\mathcal{M}_{AB}}$ either moves g to b_i for some $1 \leq i \leq k$ or moves b below g or both. Moreover, any rotation in $O_{\mathcal{M}_{AB}}$ moves g from b_i for some $1 \leq i \leq k$.*

Proof. Consider a rotation $\rho \in I_{\mathcal{M}_{AB}}$. Let $M \in \mathcal{M}_A \setminus \mathcal{M}_{AB}$ be a stable matching where ρ is exposed such that $M/\rho \in \mathcal{M}_{AB}$. By Lemma 15, M/ρ matches g to a partner between b_1 and b_k in g's list, and match b to a partner below g in b's list. Moreover, M either does not contain $b_i g$ for all $1 \leq i \leq k$ or contains bg' where $g' \geq_b g$ or both. Therefore, ρ must either moves b_i to g for some $1 \leq i \leq k$ or moves b' below g or both.

Consider a rotation $\rho \in I_{\mathcal{M}_{AB}}$ such that $M \in \mathcal{M}_{AB}$ and $M/\rho \in \mathcal{M}_A \setminus \mathcal{M}_{AB}$. Again, by Lemma 15, M contains $b_i g$ for $1 \leq i \leq k$ and bg' where $g' <_b g$. Since M dominates M/ρ in the boy optimal lattice, b must prefer g' to his partner in M/ρ. Therefore, M/ρ does not contain $b_i g$ for all $1 \leq i \leq k$, and ρ must moves b_i from g for some $1 \leq i \leq k$. $\square$

Let $\{b_{i_1}, \ldots b_{i_l}\}$ be the set of possible partners of g in any stable matching such that $1 \leq i_1 \leq \ldots \leq i_l \leq k$. Let ρ_1 be a rotation moving g to b_{i_l}, ρ_2 be the rotation moving b below g and ρ_3 be a rotation moving g from b_{i_1}. Note that each of ρ_1, ρ_2 and ρ_3 might not exist.

Lemma 17. *If both ρ_1 and ρ_2 exist. Then $\rho_1 \preceq \rho_2$.*

Proof. Assume that $\rho_1 \neq \rho_2$ and there exists a sequence of rotation eliminations, from M_0 to a stable matching M in which ρ_2 is exposed, that does not contain ρ_1. Since ρ_2 moves b below g, g is matched a partner higher than b in her list in M/ρ_2. Therefore, the partner can only be b_{i_l} or a boy higher than b_{i_l} in g's list.

Consider any sequence of rotation eliminations from M/ρ to M_z. In the sequence, the position of g's partner can only go higher in her list. Therefore, ρ_1 can not be exposed in any matching in the sequence. It follows that ρ_1 is not exposed in a sequence of eliminations from M_0 to M_z, which is a contradiction by Lemma 1. □

Theorem 12. *There is at most one rotation in $I_{\mathcal{M}_{AB}}$ and at most one rotation in $O_{\mathcal{M}_{AB}}$. Moreover, the rotation in $I_{\mathcal{M}_{AB}}$ must be either ρ_1 or ρ_2, and the rotation in $O_{\mathcal{M}_{AB}}$ must be ρ_3.*

Proof. If g does not have any partner in $\{b_1, \ldots, b_k\}$ in any stable matching, $\mathcal{M}_{AB} = \emptyset$ by Lemma 15. Therefore, we may assume that g has at least one partner b_i for $i \in [1, k]$. In other words, the set $\{b_{i_1}, \ldots b_{i_l}\}$ is non-empty. Hence, ρ_1 exists. Let r_j be the rotation that moves g to b_{i_j} for $1 \leq j \leq l$. We have

$$\rho_1 = r_l \prec r_{l-1} \prec \ldots \prec r_1 \prec \rho_3.$$

If ρ_2 exists, $\rho_1 \prec \rho_2$ by Lemma 17. If $\rho_3 \preceq \rho_2$, then $\mathcal{M}_{AB} = \emptyset$. Otherwise, ρ_2 is the unique rotation in $I_{\mathcal{M}_{AB}}$.

If ρ_2 does not exist, ρ_1 is the unique rotation in $I_{\mathcal{M}_{AB}}$.

Notice that eliminating ρ_j for any $1 \leq j \leq l$ gives a matching in which g is matched to b_{i_j}. By Lemma 15, ρ_3 is the only possible rotation in $O_{\mathcal{M}_{AB}}$. □

By Theorem 12, there is at most one rotation ρ_{in} coming into $\mathcal{M}_{AB}$ and at most one rotation ρ_{out} coming out of $\mathcal{M}_{AB}$. Since we can compute Π_A efficiently, ρ_{in} and ρ_{out} can also be computed efficiently.

Corollary 1. *ρ_{in} and ρ_{out} can be computed in polynomial time.*

Lemma 18. *Let M be the a matching in $\mathcal{M}_{AB}$ and S be the corresponding closed subset in Π_A. If ρ_1 exists, S must contain ρ_1. If ρ_2 exists, S must contain ρ_2. If ρ_3 exists, S must* *not contain* ρ_3.

Proof. If ρ_1 exists, M_0 does not contain $b_i g$ for all $i \in [1, k]$. Since $M \in \mathcal{M}_{AB}$, by Lemma 15 M matches g to a boy between b_1 and b_k in her list. the set of rotations eliminated from M_0 to M must include ρ_1.

If ρ_2 exists, b can not be below g in M_0. Since b is below g in M, by Lemma 15 the set of rotations eliminated from M_0 to M must include ρ_2.

Assume that ρ_3 exists and S contains ρ_3. Since ρ_3 moves g up from b_{i_1}, M can not contain $b_i g$ for all $i \in [1, k]$. This is a contradiction. □

3.1.3 The rotation poset for the sublattice M_{AB}

From the previous section we know that M_{AB} is a sublattice of M_A. In this section we give the rotation poset that generates all stable matchings in the sublattices.

We may assume that $M_{AB} \neq \emptyset$. If ρ_{in} exists, let $\Pi_{\text{in}} = \{\rho \in \Pi_A : \rho \preceq \rho_{\text{in}}\}$ and M_{boy} be the matching generated by Π_{in}. Otherwise, let $M_{\text{boy}} = M_0$. Similarly, let M_{girl} be the matching generated by $\Pi_A \setminus \Pi_{\text{out}}$, where $\Pi_{\text{out}} = \{\rho \in \Pi_A : \rho \succeq \rho_{\text{out}}\}$, if ρ_{out} exists, and $M_{\text{girl}} = M_z$ otherwise.

Lemma 19. *M_{boy} is the boy-optimal matching in $\mathcal{M}_{AB}$, and M_{girl} is the girl-optimal matching in $\mathcal{M}_{AB}$.*

Proof. Let M be a matching in $\mathcal{M}_{AB}$ generated by a closed subset $S \subseteq \Pi_A$. By Lemma 18, if ρ_{in} exists, S must contain ρ_{in}. Since Π_{in} is the minimum set containing ρ_{in}, $\Pi_{\text{in}} \subseteq S$. Therefore, $M_{\text{boy}} \preceq M$.

To prove that $M \preceq M_{\text{girl}}$, we show $S \subseteq \Pi_A \setminus \Pi_{\text{out}}$. Assume otherwise, then there exists a rotation $\rho \in S$ such that $\rho \notin \Pi_A \setminus \Pi_{\text{out}}$. It follows that $\rho \in \Pi_{\text{out}}$, and hence $\rho \succeq \rho_{\text{out}}$. Since

S contains ρ and S is a closed subset, S must also contain ρ_{out}. This is a contradiction by 18. $\square$

Theorem 13. $\Pi_{AB} = \Pi_A \setminus (\Pi_{in} \cup \Pi_{out})$ *is the rotation poset generating* $\mathcal{M}_{AB}$.

Proof. Let M be a matching in $\mathcal{M}_{AB}$ generated by a closed subset $S \subseteq \Pi_A$. Let $S' = S \setminus \Pi_{\text{in}}$. We show that S' is a closed subset of Π_{AB} and eliminating the rotations in S' starting from M_{boy} according to the topological ordering of the elements gives M.

First $S' \cap \Pi_{\text{in}} = \emptyset$ trivially. Since $M \in \mathcal{M}_{AB}$, S does not contain ρ_{out} by Lemma 18. Therefore, S' does not contain ρ_{out}, and $S' \cap \Pi_{\text{out}} = \emptyset$. It follows that S' is a closed subset of Π_{AB}.

Next observe that we can eliminate rotations in S from M_0 by eliminating rotations in Π_{in} first and then eliminating rotations in $S \setminus \Pi_{\text{in}}$. This can be done because Π_{in} is a closed subset of Π_A. Since Π_{in} generates M, the lemma follows. $\square$

3.2 Algorithm for finding a robust stable matching

We now use the structural properties described in Section 3.1 to give a polynomial time algorithm for finding a robust stable matching. Clearly, the results in Section 3.1 can be reproduced when we make a shift in a boy's list. Recall that given a discrete probability distribution $\mathcal{D}$ on all possible shifts, a robust stable matching is a stable matching $M \in \mathcal{M}_A$ that minimizes the probability that $M \in \mathcal{M}_{AB}$, where $B \sim \mathcal{D}$.

For a shift B, let ρ^B_{in} and ρ^B_{out} be the rotation going into $\mathcal{M}_{AB}$ and out of $\mathcal{M}_{AB}$ respectively. By Corollary 1, ρ^B_{in} and ρ^B_{out} can be computed efficiently for each B.

By Lemma 3, Π_A can be computed in polynomial time. We create two additional vertices, a source s and a sink t. For a shift B such that ρ^B_{in} does not exist, let $\rho^B_{\text{in}} = s$. Similarly, for a shift B such that ρ^B_{out} does not exist, let $\rho^B_{\text{out}} = t$.

Let p_B be the probability that instance B is chosen according to D. Consider the following integer program:

$$
\begin{aligned}
\min \quad & \sum_B x_B p_B \\
\text{s.t.} \quad & y_{\rho_1} \le y_{\rho_2} && \forall \rho_1, \rho_2 : \rho_1 \prec \rho_2 \\
& y_t = 1 \\
& y_s = 0 \\
& x_B \ge y_{\rho^B_{\text{out}}} - y_{\rho^B_{\text{in}}} && \forall B \\
& x_B \ge 0 && \forall B \\
& y_\rho \in \{0,1\} && \forall \rho \in \Pi_A.
\end{aligned}
\tag{IP}
$$

Lemma 20. *(IP) gives a solution to a robust stable matching.*

Proof. Let $S = \{\rho : y_\rho = 0\}$. The set of constraints:

$$y_{\rho_1} \le y_{\rho_2} \quad \forall \rho_1, \rho_2 : \rho_1 \prec \rho_2$$

guarantees that S is a closed subset.

Notice that $x_B = 1$ if and only if $y_{\rho^B_{\text{out}}} = 1$ and $y_{\rho^B_{\text{in}}} = 0$. This, in turn, happens if and only if the matching generated by S is in $\mathcal{M}_{AB}$.

Therefore, by minimizing $\sum_{e \in E} x_B p_B$, we can find a closed subset that generates a robust stable matching. $\square$

Lemma 21. *(IP) can be solved in polynomial time.*

Proof. Consider relax the constraint $y_\rho \in \{0,1\}$ to $0 \le y_\rho \le 1$. We show how to round a solution of this natural LP-relaxation of (IP) to have an integral solution of the same objective function. It suffices to just consider $\boldsymbol{y}$ as x_B will always be set to $\max(0, y_{\rho^B_{\text{out}}} -$

$y_{\rho^B_{\text{in}}}$) for any given $\boldsymbol{y}$.

Let $\boldsymbol{y}$ be a fractional optimal solution of the relaxation. Let $1 = a_0 > a_1 > a_2 > \ldots > a_k > a_{k+1} = 0$ be all the possible y-values. Since $\boldsymbol{y}$ is fractional, $k \geq 1$. Denote S_i by the set of all rotations having y-value equal to a_i, where $1 > a_i > 0$.

Let $\mathcal{B}^+$ be the set of instances B such that:

- $x_B = y_{\rho^B_{\text{out}}} - y_{\rho^B_{\text{in}}} > 0$.
- $y_{\rho^B_{\text{out}}} = a_i$.
- $y_{\rho^B_{\text{in}}} \neq a_i$.

Let $\mathcal{B}^-$ be the set of instances B such that:

- $x_B = y_{\rho^B_{\text{out}}} - y_{\rho^B_{\text{in}}} > 0$.
- $y_{\rho^B_{\text{in}}} = a_i$.
- $y_{\rho^B_{\text{out}}} \neq a_i$.

Consider perturbing the y-value of all rotations in S_a by a small amount ϵ:

$$y_\rho \leftarrow y_\rho + \epsilon = a_i + \epsilon \quad \forall \rho \in S_a.$$

Here ϵ is chosen so that $a_i + \epsilon < a_{i-1}$ and $a_i + \epsilon > a_{i+1}$. The net change in the objective function is

$$\sum_{B \in \mathcal{B}^+} \epsilon p_B - \sum_{B \in \mathcal{B}^-} \epsilon p_B = \epsilon \left(\sum_{B \in \mathcal{B}^+} p_B - \sum_{B \in \mathcal{B}^-} p_B \right).$$

We claim that

$$\sum_{B \in \mathcal{B}^+} p_B - \sum_{B \in \mathcal{B}^-} p_B = 0.$$

Assume otherwise, we can pick a sign of ϵ to have a strictly smaller objective function. Since $\sum_{B \in \mathcal{B}^+} p_B - \sum_{B \in \mathcal{B}^-} p_B = 0$, we can choose $\epsilon = a_{i-1} - a_i$ and obtain another optimal solution where the value of k decreases by 1. Keep going until $k = 0$ gives an integral solution. $\square$

Finally, Theorem 1 follows from Lemmas 20 and 21.

CHAPTER 4

A GENERALIZATION OF BIRKHOFF'S THEOREM FOR DISTRIBUTIVE LATTICES, WITH APPLICATIONS TO ROBUST STABLE MATCHINGS

In this chapter we state and prove a generalization of Birkhoff's theorem. From the generalization, we present a structural property arising when a lattice is partitioned into a sublattice and a semi-sublattice. Finally, we show how to apply the obtained property to find a fully robust stable matching.

4.1 A Generalization of Birkhoff's Theorem

Let P be a finite poset. For simplicity of notation, in this chapter we will assume that P 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. Further, s precedes all other elements and t succeds all other elements in P. A *proper closed set* of P 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 P form a distributive lattice under the operations of set intersection and union. We will denoted this lattice by $L(P)$. Birkhoff's Theorem states that every finite distributive lattice is isomorphic to the proper closed sets of some poset. On occassion we will also say that poset P *generates* lattice $\mathcal{L}$

Theorem 14. *(Birkhoff [14]) Every finite distributive lattice $\mathcal{L}$ is isomorphic to $L(P)$, for some finite poset P.*

Our generalization of Birkhoff's Theorem deals with the sublattices of a finite distributive lattice. First, in Definition 1 we state the critical operation of *compression of a poset*.

Definition 1. *Given a finite poset P, first partition its elements; each subset will be called a*

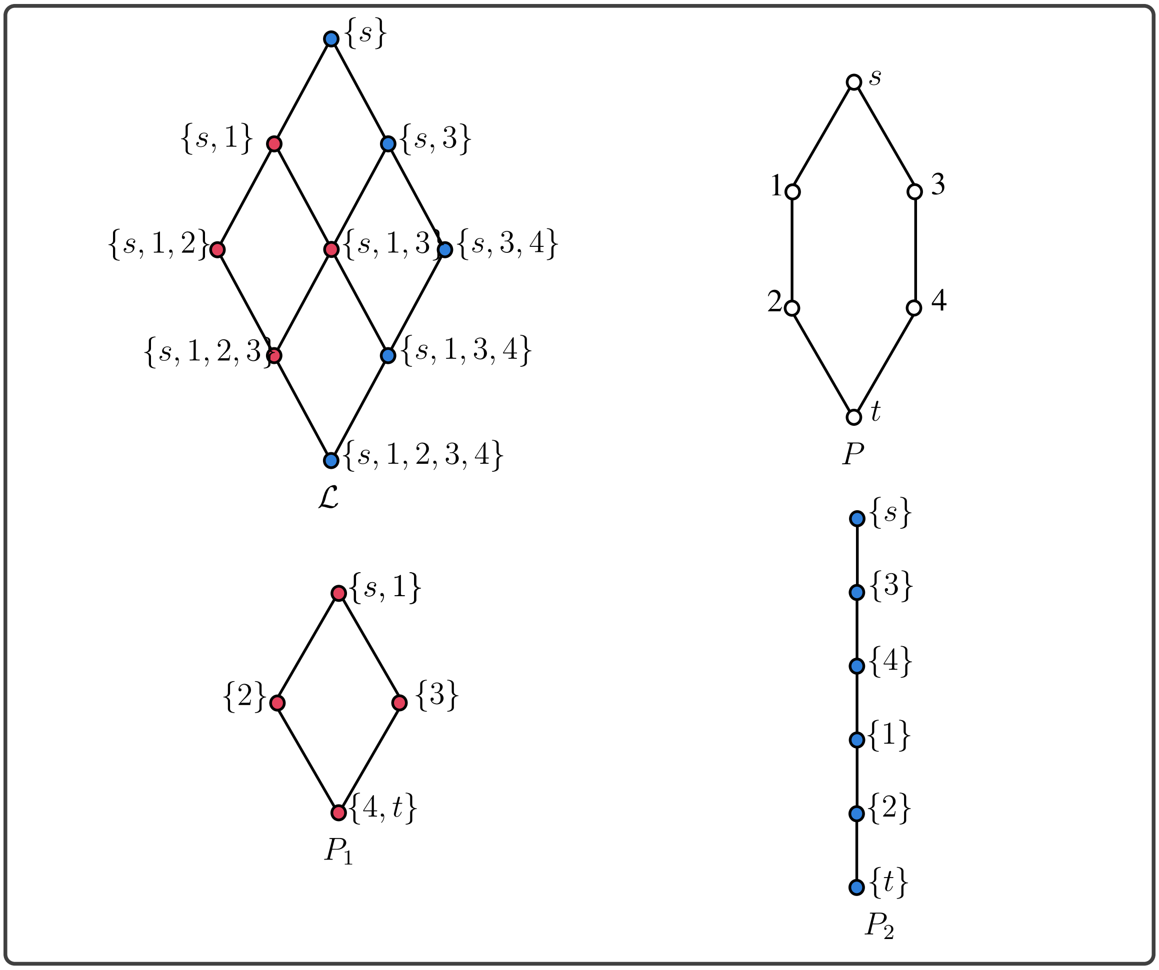

Figure 4.1: Two examples of compressions. Lattice $\mathcal{L} = L(P)$. P_1 and P_2 are compressions of P, and they generate the sublattices in $\mathcal{L}$, of red and blue elements, respectively.

meta-element. *Define the following precedence relations among the meta-elements: if x, y are elements of P 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 P_f be any partial order on the meta-elements such that the precedence relations of Q are a subset of the precedence relations of P_f. Then P_f will be called a* compression *of P. Let A_s and A_t denote the meta-elements of P_f containing s and t, respectively.*

For examples of compressions see Figure 4.1. Clearly, A_s precedes all other meta-elements in P_f and A_t succeeds all other meta-elements in P_f. Once again, by a *proper closed set of*

P_f we mean a closed set of P_f that contains A_s and does not contain A_t. Then the lattice formed by the set of all proper closed sets of P_f will be denoted by $L(P_f)$.

Our generalization of Birkhoff's Theorem is as follows:

Theorem 15. *There is a one-to-one correspondence between the compressions of P and the sublattices of $L(P)$. Furthermore, if a sublattice $\mathcal{L}'$ of $L(P)$ corresponds to compression P_f, then $\mathcal{L}'$ is isomorphic to $L(P_f)$.*

We will prove Theorem 15 in the context of stable matching lattices; this is w.l.o.g. since stable matching lattices are as general as finite distributive lattices. In this context, the proper elements of partial order P will be rotations, and meta-elements are called *meta-rotations*. Let $\mathcal{L} = L(P)$ be the corresponding stable matching lattice.

Clearly it suffices to show that:

- Given a compression P_f, $L(P_f)$ is isomorphic to a sublattice of $\mathcal{L}$.
- A sublattice $\mathcal{L}'$ is isomorphic to $L(P_f)$ for some compression P_f.

These two proofs are given in Sections 4.1.1 and 4.1.2, respectively.

4.1.1 $L(P_f)$ is isomorphic to a sublattice of $L(P)$

Let I be a closed subset of P_f; clearly I is a set of meta-rotations. Define $\mathrm{rot}(I)$ to be the union of all meta-rotations in I, i.e.,

$$\mathrm{rot}(I) = \{\rho \in A : A \text{ is a meta-rotation in } I\}.$$

We will define the process of *elimination of a meta-rotation* A of P_f to be the elimination of the rotations in A in an order consistent with partial order P. Furthermore, *elimination of meta-rotations in* I will mean starting from stable matching M_0 in lattice $\mathcal{L}$ and eliminating all meta-rotations in I in an order consistent with P_f. Observe that this is equivalent to starting from stable matching M_0 in $\mathcal{L}$ and eliminating all rotations in $\mathrm{rot}(I)$ in an order consistent with partial order P. This follows from Definition 1, since if there exist rotations x, y in P such that x is in meta-rotation X, y is in meta-rotation Y and x precedes y, then X must also precede Y. Hence, if the elimination of all rotations in $\mathrm{rot}(I)$ gives matching M_I, then elimination of all meta-rotations in I will also give the same matching.

Finally, to prove the statement in the title of this section, it suffices to observe that if I and J are two proper closed sets of the partial order P_f then

$$\mathrm{rot}(I \cup J) = \mathrm{rot}(I) \cup \mathrm{rot}(J) \quad \text{and} \quad \mathrm{rot}(I \cap J) = \mathrm{rot}(I) \cap \mathrm{rot}(J).$$

It follows that the set of matchings obtained by elimination of meta-rotations in a proper closed set of P_f are closed under the operations of meet and join and hence form a sublattice of $\mathcal{L}$.

4.1.2 $\mathcal{L}'$ is isomorphic to $L(P_f)$, for a compression P_f of P

We will obtain compression P_f of P in stages. First, we show how to partition the set of rotations of P to obtain the meta-rotations of P_f. We then find precedence relations among these meta-rotations to obtain P_f. Finally, we show $L(P_f) = \mathcal{L}'$.

Notice that $\mathcal{L}$ can be represented by its Hasse diagram $H(\mathcal{L})$. Each edge of $H(\mathcal{L})$ *contains* a (not necessarily unique) rotation of P. Then, by Lemma 1, for any two stable matchings $M_1, M_2 \in \mathcal{L}$ such that $M_1 \prec M_2$, all paths from M_1 to M_2 in $H(\mathcal{L})$ contain the same set of rotations.

Definition 2. *For $M_1, M_2 \in \mathcal{L}'$, M_2 is said to be an $\mathcal{L}'$-*direct successor *of M_1 iff $M_1 \prec M_2$ and there is no $M \in \mathcal{L}'$ such that $M_1 \prec M \prec M_2$. Let $M_1 \prec \ldots \prec M_k$ be a sequence of matchings in $\mathcal{L}'$ such that M_{i+1} is an $\mathcal{L}'$-direct successor of M_i for all $1 \leq i \leq k-1$. Then any path in $H(\mathcal{L})$ from M_1 to M_k containing M_i, for all $1 \leq i \leq k-1$, is called an*

$\mathcal{L}'$-path.

Let $M_{0'}$ and $M_{z'}$ denote the boy-optimal and girl-optimal matchings, respectively, in $\mathcal{L}'$. For $M_1, M_2 \in \mathcal{L}'$ with $M_1 \prec M_2$, let S_{M_1,M_2} denote the set of rotations contained on any $\mathcal{L}'$-path from M_1 to M_2. Further, let $S_{M_0,M_{0'}}$ and $S_{M_{z'},M_z}$ denote the set of rotations contained on any path from M_0 to $M_{0'}$ and $M_{z'}$ to M_z, respectively in $H(\mathcal{L})$. Define the following set whose elements are sets of rotations.

$$\mathcal{S} = \{S_{M_i,M_j} \mid M_j \text{ is an } \mathcal{L}'\text{-direct successor of } M_i, \text{ for every pair of matchings } M_i, M_j \text{ in } \mathcal{L}'\} \bigcup$$

$$\{S_{M_0,M_{0'}},\ S_{M_{z'},M_z}\}.$$

Lemma 22. *$\mathcal{S}$ is a partition of P.*

Proof. First, we show that any rotation must be in an element of $\mathcal{S}$. Consider a path p from M_0 to M_z in the $H(\mathcal{L})$ such that p goes from $M_{0'}$ to $M_{z'}$ via an $\mathcal{L}'$-path. Since p is a path from M_0 to M_z, all rotations of P are contained on p by Lemma 1. Hence, they all appear in the sets in $\mathcal{S}$.

Next assume that there are two pairs $(M_1, M_2) \neq (M_3, M_4)$ of $\mathcal{L}'$-direct successors such that $S_{M_1,M_2} \neq S_{M_3,M_4}$ and $X = S_{M_1,M_2} \cap S_{M_3,M_4} \neq \emptyset$. The set of rotations eliminated from M_0 to M_2 is

$$S_{M_0,M_2} = S_{M_0,M_1} \cup S_{M_1,M_2}.$$

Similarly,

$$S_{M_0,M_4} = S_{M_0,M_3} \cup S_{M_3,M_4}.$$

Therefore,

$$S_{M_0,M_2 \vee M_3} = S_{M_0,M_3} \cup S_{M_1,M_2} \cup S_{M_0,M_1}.$$

$$S_{M_0,M_1 \vee M_4} = S_{M_0,M_3} \cup S_{M_3,M_4} \cup S_{M_0,M_1}.$$

Let $M = (M_2 \vee M_3) \wedge (M_1 \vee M_4)$, we have

$$S_{M_0,M} = S_{M_0,M_3} \cup S_{M_0,M_1} \cup X.$$

Hence,

$$S_{M_0,M \wedge M_2} = S_{M_0,M_1} \cup X.$$

Since $X \subset S_{M_1,M_2}$ and $S_{M_1,M_2} \cap S_{M_0,M_1} = \emptyset$, $X \cap S_{M_0,M_1} = \emptyset$. Therefore,

$$S_{M_0,M_1} \subset S_{M_0,M \wedge M_2} \subset S_{M_0,M_2},$$

and hence M_2 is not a $\mathcal{L}'$-direct successor of M_1, leading to a contradiction. $\square$

We will denote $S_{M_0,M_{0'}}$ and $S_{M_{z'},M_z}$ by A_s and A_t, respectively. The elements of $\mathcal{S}$ will be the meta-rotations of P_f. Next, we need to define precedence relations among these meta-rotations to complete the construction of P_f. For a meta-rotation $A \in \mathcal{S}$, $A \neq A_t$, define the following subset of $\mathcal{L}'$:

$$\mathcal{M}^A = \{M \in \mathcal{L}' \text{ such that } A \subseteq S_{M_0,M}\}.$$

Lemma 23. *For each meta-rotation $A \in \mathcal{S}$, $A \neq A_t$, $\mathcal{M}^A$ forms a sublattice $\mathcal{L}^A$ of $\mathcal{L}'$.*

Proof. Take two matchings M_1, M_2 such that S_{M_0,M_1} and S_{M_0,M_2} are supersets of A. Then $S_{M_0,M_1 \wedge M_2} = S_{M_0,M_1} \cap S_{M_0,M_2}$ and $S_{M_0,M_1 \vee M_2} = S_{M_0,M_1} \cup S_{M_0,M_2}$ are also supersets of A. $\square$

Let M^A be the boy-optimal matching in the lattice $\mathcal{L}^A$. Let p be any $\mathcal{L}'$-path from $M_{0'}$ to M^A and let $\text{pre}(A)$ be the set of meta-rotations appearing before A on p.

Lemma 24. *The set* $\text{pre}(A)$ *does not depend on p. Furthermore, on any $\mathcal{L}'$-path from $M_{0'}$ containing A, each meta-rotation in* $\text{pre}(A)$ *appears before A.*

Proof. Since all paths from $M_{0'}$ to M^A give the same set of rotations, all $\mathcal{L}'$-paths from $M_{0'}$ to M^A give the same set of meta-rotations. Moreover, A must appear last in the any $\mathcal{L}'$-path from $M_{0'}$ to M^A; otherwise, there exists a matching in $\mathcal{L}^A$ preceding M^A, giving a contradiction. It follows that $\mathrm{pre}(A)$ does not depend on p.

Let q be an $\mathcal{L}'$-path from $M_{0'}$ that contains matchings $M', M \in \mathcal{L}'$, where M is an $\mathcal{L}'$-direct successor of M'. Let A denote the meta-rotation that is contained on edge (M', M). Suppose there is a meta-rotation $A' \in \mathrm{pre}(A)$ such that A' does not appear before A on q. Then $S_{M_0, M^A \wedge M} = S_{M_0, M^A} \cap S_{M_0, M}$ contains A but not A'. Therefore $M^A \wedge M$ is a matching in $\mathcal{L}^A$ preceding M^A, giving is a contradiction. Hence all matchings in $\mathrm{pre}(A)$ must appear before A on all such paths q. $\square$

Finally, add precedence relations from all meta-rotations in $\mathrm{pre}(A)$ to A, for each meta-rotation in $\mathcal{S} - \{A_t\}$. Also, add precedence relations from all meta-rotations in $\mathcal{S} - \{A_t\}$ to A_t. This completes the construction of P_f. Below we show that P_f is indeed a compression of P, but first we need to establish that this construction does yield a valid poset.

Lemma 25. *P_f satifies transitivity and anti-symmetry.*

Proof. First we prove that P_f satifies transitivity. Let A_1, A_2, A_3 be meta-rotations such that $A_1 \prec A_2$ and $A_2 \prec A_3$. We may assume that $A_3 \neq A_t$. Then $A_1 \in \mathrm{pre}(A_2)$ and $A_2 \in \mathrm{pre}(A_3)$. Since $A_1 \in \mathrm{pre}(A_2)$, $S_{M_0, M^{A_2}}$ is a superset of A_1. By Lemma 23, $M^{A_1} \prec M^{A_2}$. Similarly, $M^{A_2} \prec M^{A_3}$. Therefore $M^{A_1} \prec M^{A_3}$, and hence $A_1 \in \mathrm{pre}(A_3)$.

Next we prove that P_f satifies anti-symmetry. Assume that there exist meta-rotations A_1, A_2 such that $A_1 \prec A_2$ and $A_2 \prec A_1$. Clearly $A_1, A_2 \neq A_t$. Since $A_1 \prec A_2$, $A_1 \in \mathrm{pre}(A_2)$. Therefore, $S_{M_0, M^{A_2}}$ is a superset of A_1. It follows that $M^{A_1} \prec M^{A_2}$. Applying a similar argument we get $M^{A_2} \prec M^{A_1}$. Now, we get a contradiction, since A_1 and A_2 are different meta-rotations. $\square$

Lemma 26. *P_f is a compression of P.*

Proof. Let x, y be rotations in P such that $x \prec y$. Let X be the meta-rotation containing x and Y be the meta-rotation containing y. It suffices to show that $X \in \mathrm{pre}(Y)$. Let p be an $\mathcal{L}'$-path from M_0 to M^Y. Since $x \prec y$, x must appear before y in p. Hence, X also appears before Y in p. By Lemma 24, $X \in \mathrm{pre}(Y)$ as desired. $\square$

Finally, the next two lemmas prove that $L(P'_f) = \mathcal{L}'$.

Lemma 27. *Any matching in $L(P'_f)$ must be in $\mathcal{L}'$.*

Proof. For any proper closed subset I in P'_f, let M_I be the matching generated by eliminating meta-rotations in I. Let J be another proper closed subset in P'_f such that $J = I \setminus \{A\}$, where A is a maximal meta-rotation in I. Then M_J is a matching in $\mathcal{L}'$ by induction. Since I contains A, $S_{M_0,M_I} \supset A$. Therefore, $M^A \prec M_I$. It follows that $M_I = M_J \vee M^A \in \mathcal{L}'$. $\square$

Lemma 28. *Any matching in $\mathcal{L}'$ must be in $L(P'_f)$.*

Proof. Suppose there exists a matching M in $\mathcal{L}'$ such that $M \notin L(P'_f)$. Then it must be the case that $S_{M_0,M}$ cannot be partitioned into meta-rotations which form a closed subset of P_f. Now there are two cases.

First, suppose that $S_{M_0,M}$ can be partitioned into meta-rotations, but they do not form a closed subset of P_f. Let A be a meta-rotation such that $S_{M_0,M} \supset A$, and there exists $B \prec A$ such that $S_{M_0,M} \not\supset B$. By Lemma 23, $M \succ M^A$ and hence $S_{M_0,M}$ is a superset of all meta-rotations in $\mathrm{pre}(A)$, giving is a contradiction.

Next, suppose that $S_{M_0,M}$ cannot be partitioned into meta-rotations in P_f. Since the set of meta-rotations partitions P, there exists a meta-rotation X such that $Y = X \cap S_{M_0,M}$ is a non-empty subset of X. Let J be the set of meta-rotations preceding X in P_f.

$(M_J \vee M) \wedge M^X$ is the matching generated by meta-rotations in $J \cup Y$. Obviously, J is a closed subset in P_f. Therefore, $M_J \in L(P_f)$. By Lemma 27, $M_J \in \mathcal{L}'$. Since $M, M^X \in \mathcal{L}'$, $(M_J \vee M) \wedge M^X \in \mathcal{L}'$ as well. The set of rotations contained on a path from

M_J to $(M_J \vee M) \wedge M^X$ in $H(\mathcal{L})$ is exactly Y. Therefore, Y can not be a subset of any meta-rotation, contradicting the fact that $Y = X \cap S_{M_0,M}$ is a non-empty subset of X. $\square$

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

Let $\mathcal{L}'$ be the sublattice of $\mathcal{L}$ generated by P_f. We will say that the set of edges E *defines* $\mathcal{L}'$. It can be seen that each set E uniquely defines a sublattice $L(P_f)$; however, there may be multiple sets that define the same sublattice. Observe that given a compression P_f of P, a set E of edges defining $L(P_f)$ can easily be obtained. See Figure 4.2 for examples of sets of edges which define sublattices.

Proposition 16. *The two definitions of compression of a poset are equivalent.*

Proof. Let P_f be a compression of P obtained using the first definition. Clearly, for each meta-rotation in P_f, we can add edges to P 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. $\square$

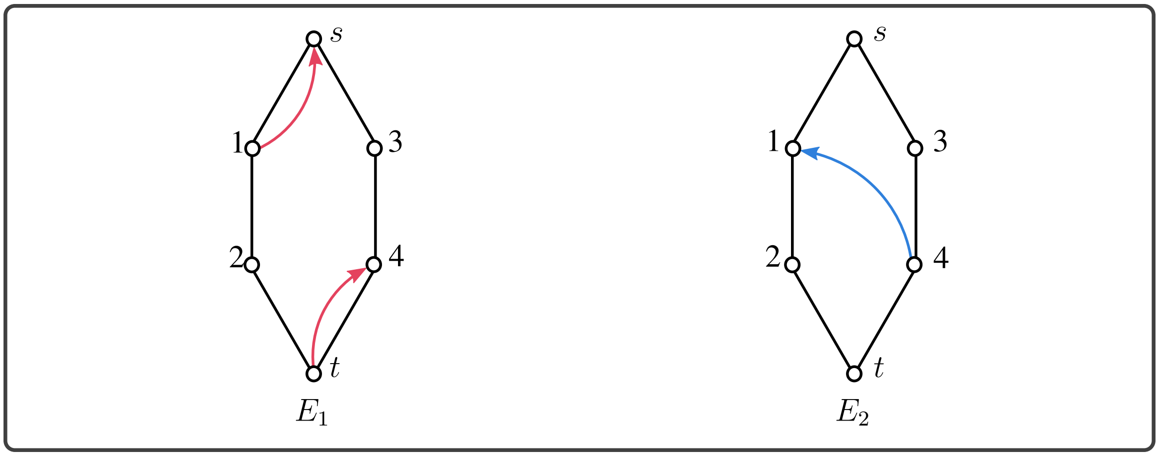

Figure 4.2: E_1 (red edges) and E_2 (blue edges) define the sublattices in Figure 4.1, of red and blue elements, respectively.

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 P. Then we say that:

- I *separates* an edge $uv \in E$ if $v \in I$ and $u \notin I$.
- I *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 29. *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 $uv \in E$.*

Proof. Let P_f be a compression corresponding to $\mathcal{L}'$. By Theorem 15, the matchings in $\mathcal{L}'$ are generated by eliminating rotations in closed subsets of P_f.

First, assume I separates $uv \in E$. Moreover, assume $M \in \mathcal{L}'$ for the sake of contradiction, and let I_f be the closed subset of P_f 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_f are identical. Therefore, $V \in I_f$ and $U \notin I_f$. Since $uv \in E$, there is an edge from U to V in H_f. Hence, I_f is not a closed subset of P_f.

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_f of P_f. 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 P_f. Assume otherwise, there exist meta-rotation $U \in I_f$ and $V \notin I_f$ such that there exists an edge from U to V in E_f. Therefore, there exists $u \in U$, $v \in V$ and $uv \in E$, which is a contradiction. $\square$

Remark 17. 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 P. 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 3. *W.r.t. an element v in a poset P, we define four useful subsets of P:*

$$I_v = \{r \in P : r \prec v\}$$

$$J_v = \{r \in P : r \preceq v\} = I_v \cup \{v\}$$

$$I'_v = \{r \in P : r \succ v\}$$

$$J'_v = \{r \in P : r \succeq v\} = I'_v \cup \{v\}$$

Notice that $I_v, J_v, P \setminus I'_v, P \setminus J'_v$ are all closed sets.

Lemma 30. *Both J_v and $P \setminus J'_v$ 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 17. Hence, J_v separates uv for all uv in E.

Similarly, since uv is in E, v cannot be in J'_u. Therefore, $P \setminus J'_v$ contains v but not u, and thus separates uv. $\square$

4.3 The Lattice Can be Partitioned into Two Sublattices

In this section we will prove the following theorem:

Theorem 18. *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.*

Again, we give a proof in the context of stable matchings. To prove the theorem, we 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 31. *There must exist a path from t to s composed of edges in E_1 and E_2.*

Proof. 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 $P \setminus R$. Without loss of generality, assume that $M \in \mathcal{L}_1$. By Lemma 29, $P \setminus R$ separates an edge $uv \in E_2$. Therefore, $u \in R$ and $v \in P \setminus R$. Since $uv \in E_2$, v is also reachable from t by a path of edges in E_1 and E_2. $\square$

Let Q be a path from t to s according to Lemma 31. 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. We will 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 32. *Let Q_i consist of edges in E_α (α = 1 or 2). Q_i can be replaced by an edge from r_{i-1} to r_i where $r_{i-1}r_i \in E_\alpha$.*

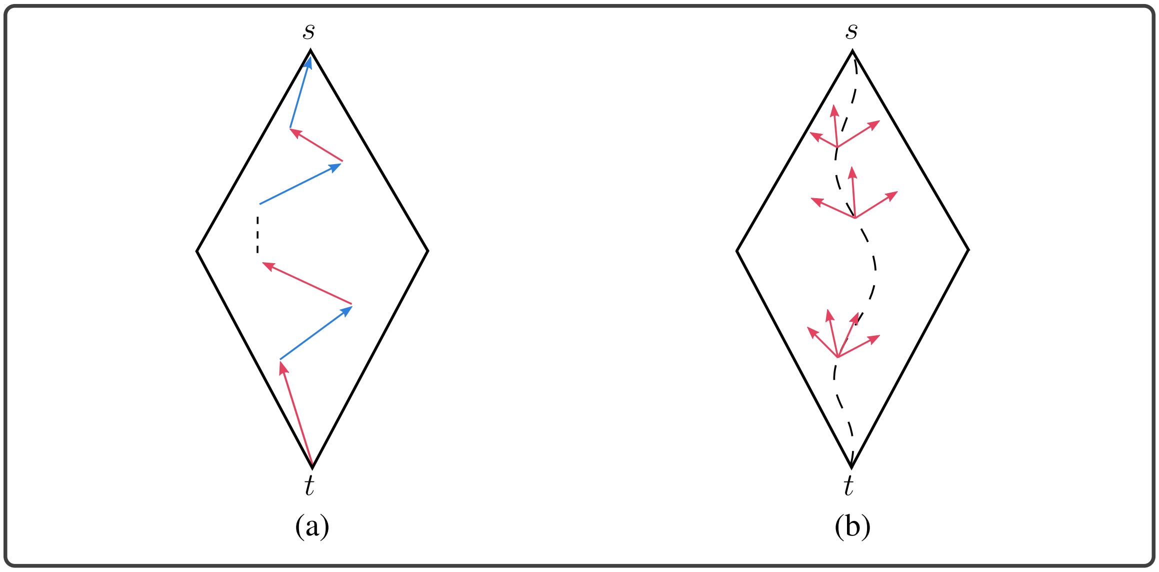

Figure 4.3: Examples of: (a) canonical path, and (b) bouquet.

Proof. 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. □

Lemma 33. *Edges in $E_1 \cup E_2$ but not in Q can be removed.*

Proof. 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 29, 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 29, 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. □

By Lemma 32 and Lemma 33, $r_0r_1, \ldots, r_{k-2}r_{k-1}, r_{k-1}r_k$ are all edges in E_1 and E_2 and they alternate between E_1 and E_2. Therefore, we have Theorem 18. An illustration of such a path is given in Figure 4.3(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 $0 \leq i \leq k$.*

4.4 The Lattice Can be Partitioned into a Sublattice and a Semi-Sublattice

Let $\mathcal{L}$ be a distributive lattice that can be partitioned into a sublattice $\mathcal{L}_1$ and a semi-sublattice $\mathcal{L}_2$. The next theorem, which generalizes Theorem 18, 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 P.*
2. *There is no path of length two consisting of edges in E.*
3. *For each $r \in T_E$, let*

$$F_r = \{v \in P : rv \in E\}.$$

   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 4.3(b). The black path, directed from s to t, is the chain mentioned in Theorem 15 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 P. 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 $P \setminus S$.

Lemma 34. *There is a unique maximal rotation in $T_E \cap S$.*

Proof. Suppose there are at least two maximal rotations $u_1, u_2, \dots u_k$ ($k \geq 2$) in $T_H \cap S$. Let $v_1, \dots v_k$ be the heads of edges containing $u_1, u_2, \dots u_k$. 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 30. 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 29.

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 semi-sublattice. $\square$

Denote by r the unique maximal rotation in $T_E \cap S$. Let

$$R_r = \{v \in P : \text{ there is a path from } r \text{ to } v \text{ using edges in } E\},$$

$$E_r = \{uv \in E : u, v \in R_r\},$$

$$G_r = \{R_r, E_r\}.$$

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 35. *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.*

Proof. 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.

□

Keep replacing edges according to Lemma 35 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 36. *Let*

$$X = \{v \in S : v \succeq x \textit{ for some } x \in R_r\}.$$

1. *$S \setminus X$ is a closed subset.*

2. *$S \setminus X$ contains u for each $u \in (T_E \cap S) \setminus R_r$.*

3. *$S \setminus X \cap R_r = \emptyset$.*

4. *$S \setminus X$ is a splitting set.*

Proof. The lemma follows from the claims given below:

Claim 1. $S \setminus X$ 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 2. $S \setminus X$ contains u for each $u \in (T_E \cap S) \setminus R_r$.

Proof. After replacing edges according to Lemma 35, 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 3. $(S \setminus X) \cap R_r = \emptyset$.

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

Claim 4. $S \setminus X$ 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 5. $S \setminus X$ 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 34. 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 17. ☐

☐

Lemma 37. *E_r can be replaced by the following set of edges:*

$$E'_r = \{rv : v \in R_r\}.$$

Proof. 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 34, $r \succ u$. Thus, I must also separate $rv \in E'_r$. $\square$

Proof of Theorem 20. To begin, let $S_1 = P$ and let r_1 be the unique maximal rotation according to Lemma 34. Then we can replace edges according to Lemma 35 and Lemma 37. After replacing, r_1 is the only tail vertex in G_{r_1}. By Lemma 36, 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 $r_1, \ldots, r_k$ be the rotations found in the above process. Since r_i is the unique maximal rotation in $T_E \cap S_i$ for all $1 \leq i \leq k$ and $S_1 \supset S_2 \supset \ldots \supset S_k$, we have $r_1 \succ r_2 \succ \ldots \succ r_k$. By Lemma 37, for each $1 \leq i \leq k$, 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, $r_1, \ldots, r_k$ 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 17. 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, $S_j \supset S_{j-1} \supset \ldots \supset S_i$, $R_{r_j} \subseteq S_j \setminus S_{j-1}$ 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. $\square$

Proposition 21. *There exists a sequence of rotations $r_1 \prec \ldots \prec r_k$ and a set F_{r_i} for each $1 \leq i \leq k$ 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(P):
Input: A poset P.
Output: A set E of edges defining $\mathcal{L}_1$.

1. Initialize: Let $S = P, E = \emptyset$.
2. If M_z is in $\mathcal{L}_1$: go to Step 3. Else: $r = t$, go to Step 5.
3. r = FINDNEXTTAIL(P, S).
4. If r is not NULL: Go to Step 5. Else: Go to Step 7.
5. F_r = FINDFLOWER(P, 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 4.4: Algorithm for finding a bouquet.

4.5 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 P 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 P. The idea of our algorihm, given in Figure 4.4, 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 P 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 $P \setminus \{t\}$ separates an edge in E according to Lemma 29. 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.

FINDNEXTTAIL(P, S):
Input: A poset P, 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:
   - $P \setminus I'_v$ generates a matching in $\mathcal{L}_1$.
   - $P \setminus J'_v$ 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 4.5: Subroutine for finding the next tail.

First we prove a simple observation.

Lemma 38. *Let v be a rotation in P. Let $S \subseteq P$ such that both S and $S \cup \{v\}$ are closed subsets. If S generates a matching in $\mathcal{L}_1$ and $S \cup \{u\}$ 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 $S \cup \{u\}$ 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 $S \cup \{u\}$ generates a matching in $\mathcal{L}_2$. By Lemma 29, S does not separate any edge in E, and $S \cup \{u\}$ 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 39. *Given a splitting set S,* FINDNEXTTAIL(P, S) *(Figure 4.5) returns the maximal tail vertex in S, or* NULL *if there is no tail vertex in S.*

Proof. 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 P. Therefore $P \setminus I'_r$ contains all tails in S. Hence, $P \setminus I'_r$ does not separate any edge whose tails are in S. Since S is a splitting set, $P \setminus I'_r$ does not separate any edge whose tails are in $P \setminus S$. Therefore, by Lemma 29, $P \setminus I'_r$ generates a matching in $\mathcal{L}_1$. By Lemma 30, $P \setminus J'_r$ must separate an edge in E, and hence generates a matching in $\mathcal{L}_2$ according to Lemma 29.

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

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 4.6: Subroutine for finding a flower.

By Lemma 38, any rotation in V must be the tail of an edge in E. Hence, they are all predecessors of r according to Theorem 20. □

Lemma 40. *Given a tail vertex r and a splitting set S containing r,* FINDFLOWER(P, S) *(Figure 4.6) correctly returns F_r.*

Proof. 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 P.

Claim 1. 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 I_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 2. 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 29, Y does not separate any edge in E. Since $r \notin Y$, Y must not contain any rotation in F_r. $\square$

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 3. $V = F_r$.

Proof. Let v be a rotation in V. By Lemma 38, 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. $\square$

$\square$

Theorem 22. FINDBOUQUET(P)*, given in Figure 4.4, returns a set of edges defining* $\mathcal{L}_1$.

Proof. From Lemmas 39 and 40, it suffices to show that S is udpated correctly in Step 6(b). To be precised, 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 36 by noticing that

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

$\square$

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 4.3.

4.6 Finding an Optimal Fully Robust Stable Matching

Consider the setting given in the Introduction, with D being the domain of all erroneous instances B under consideration. We show how to use the algorithm in Section 4.5 to find the poset generating all fully robust matchings w.r.t. D, and then use this poset to obtain a fully robust matching maximizing (or minimizing) any given weight function.

4.6.1 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 boy or one girl. Lemma 41 gives an example of a permutation so that $\mathcal{M}_{AB}$ is not a sublattice of $\mathcal{L}_A$, hence showing that the case studied in Section 4.3 does not suffice to solve the problem at hand. On the other hand, for all such instances B, Lemma 42 shows that $\mathcal{M}_{AB}$ forms a semi-sublattice of $\mathcal{L}_A$ and hence the case studied in Section 4.4 does suffice.

The next lemma pertains to the example given in Figure 4.7, in which the set of boys is $\mathcal{B} = \{a, b, c, d\}$ and the set of girls is $\mathcal{G} = \{1, 2, 3, 4\}$. Instance B is obtained from instance A by permuting girl 1's list.

Lemma 41. *$\mathcal{M}_{AB}$ is not a sublattice of $\mathcal{L}_A$.*

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
Girls' preferences in A					Girls' preferences in B					Boys' preferences in both instances				

Figure 4.7: An example in which $\mathcal{M}_{AB}$ is not a sublattice of $\mathcal{L}_A$.

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

In going from A to B, the positions of boys b and c are swapped in girl 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 $\mathcal{M}_{AB}$. However, $M_1 \wedge M_2$ is a stable matching under B, and therefore is it not in $\mathcal{M}_{AB}$. Hence, $\mathcal{M}_{AB}$ is not closed under the $\wedge$ operation. □

Lemma 42. *For any instance B obtained by permuting the preference list of one boy or one girl, $\mathcal{M}_{AB}$ forms a semi-sublattice of $\mathcal{L}_A$.*

Proof. Without loss of generality, assume that the preference list of a girl g is permuted. Let M_1 and M_2 be two matchings in $\mathcal{M}_{AB}$. Hence, neither of them are in $\mathcal{M}_B$. In other words, each has a blocking pair under instance B.

Let b be the partner of g in $M_1 \vee M_2$. Then b must also be matched to g in either M_1 or M_2 (or both). We may assume that b is matched to g 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 $M_1 \vee M_2$ under B. To begin, the girl y must be g since other preference lists remain unchanged. Since xg is a blocking pair of M_1 under B, $x >_g^B b$. Similarly, $g >_x g'$ where g' is the M_1-partner of x. Let g'' be the partner of x in $M_1 \vee M_2$. Then $g' \geq_x g''$. It follows that $g >_x g''$. Since $x >_g^B b$ and $g >_x g''$, xg must be a blocking pair of $M_1 \vee M_2$ under B. □

Proposition 23. *A set of edges defining the sublattice $\mathcal{L}'$, consisting of matchings in $\mathcal{M}_A \cap \mathcal{M}_B$, can be computed efficiently.*

Proof. We have that $\mathcal{L}'$ and $\mathcal{M}_{AB}$ partition $\mathcal{L}_A$, with $\mathcal{M}_{AB}$ being a semi-sublattice of $\mathcal{L}_A$, by Lemma 42. Therefore, FINDBOUQUET(P) finds a set of edges defining $\mathcal{L}'$ by Theorem 22.

By Lemma 3, the input P 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 P has $O(n^2)$ vertices (Lemma 3), any step of FINDBOUQUET takes polynomial time. □

4.6.2 Optimizing fully robust stable matchings

Finally, we will prove Theorem 2. 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 $1 \leq i \leq k$. Clearly, $\mathcal{L}' = \mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \ldots \cap \mathcal{M}_{B_k}$ is a sublattice of $\mathcal{L}_A$.

Lemma 43. $E = \bigcup_i E_i$ *defines* $\mathcal{L}'$.

Proof. By Lemma 29, 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 29. □

By Lemma 43, a compression P_f generating $\mathcal{L}'$ can be constructed from E as described in Section 4.2. By Proposition 23, we can compute each E_i, and hence, P_f efficiently. Clearly, P_f 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 P_f. This happens iff s and

t belong to different meta-rotations in P_f, an easy to check condition. Hence, we have Theorem 2.

We can use P_f to obtain a fully robust stable matching M maximizing $\sum_{bg \in M} w_{bg}$ by applying the algorithm of [18]. Specifically, let $H(P_f)$ be the Hasse diagram of P_f. Then each pair bg for $b \in \mathcal{B}$ and $g \in \mathcal{G}$ can be associated with two vertices u_{bg} and v_{bg} in $H(P_f)$ as follows:

- If there is a rotation r moving b to g, u_{bg} is the meta-rotation containing r. Otherwise, u_{bg} is the meta-rotation containing s.
- If there is a rotation r moving b from g, v_{bg} is the meta-rotation containing r. Otherwise, v_{bg} is the meta-rotation containing t.

By Lemma 2 and the definition of compression, $u_{bg} \prec v_{bg}$. Hence, there is a path from u_{bg} to v_{bg} in $H(P_f)$. We can then add weights to edges in $H(P_f)$, as stated in Chapter 2. Specifically, we start with weight 0 on all edges and increase weights of edges in a path from u_{bg} to v_{bg} by w_{bg} for all pairs bg. A fully robust stable matching maximizing $\sum_{bg \in M} w_{bg}$ can be obtained by finding a maximum weight ideal cut in the constructed graph. An efficient algorithm for the latter problem is given in [18].

CHAPTER 5

CONCLUSION

The structural and algorithmic results introduced in this thesis naturally lead to a number of new questions such as further extending the domain of error on the robust matching problem, improving the running time of our algorithms, extending to the stable roommate problem, etc.

Another interesting direction is to bound the number of stable matchings by partitioning the stable matching lattice into sublattices and applying the generalization of Birkhoff's Theorem.

Considering the deep and pristine structure of stable matching, it will not be surprising if many of these questions do get settled satisfactorily in due course of time. As stated above, some of our results, such as the generalization of Birkhoffs Theorem, transcend the setting of stable matching and should be applicable more widely.

REFERENCES

[1] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[2] D. Gusfield and R. W. Irving, *The stable marriage problem: structure and algorithms*. MIT press, 1989.

[3] R. W. Irving, "An efficient algorithm for the stable roommatesi problem," *Journal of Algorithms*, vol. 6, no. 4, pp. 577–595, 1985.

[4] R. W. Irving and P. Leather, "The complexity of counting stable marriages," *SIAM Journal on Computing*, vol. 15, no. 3, pp. 655–667, 1986.

[5] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[6] D. E. Knuth, *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*. American Mathematical Soc., 1997.

[7] D. Manlove, *Algorithmics of Matching Under Preferences*. World Scientific, 2013.

[8] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, New York, NY, 1986.

[9] J. H. V. Vate, "Linear programming brings marital bliss," *Operations Research Letters*, vol. 8, no. 3, pp. 147–153, 1989.

[10] U. G. Rothblum, "Characterization of stable matchings as extreme points of a polytope," *Mathematical Programming*, vol. 5, pp. 57–67, 1992.

[11] R. Irving, P.Leather, and D. Gusfield, "An efficient algorithm for the "optimal" stable marriage," *Journal of the ACM*, vol. 34, pp. 532–543, 1987.

[12] G. C. Calafiore and L. El Ghaoui, "On distributionally robust chance-constrained linear programs," *Jour. of Optimization Theory and Applications*, vol. 130, no. 1, Dec. 2006.

[13] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, ser. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[14] G. Birkhoff, "Rings of sets," *Duke Mathematical Journal*, vol. 3, no. 3, pp. 443–454, 1937.

[15] R. Stanley, *Enumerative combinatorics, vol. 1, wadsworth and brooks/cole, pacific grove, ca, 1986; second printing*, 1996.

[16] C. Blair, "Every finite distributive lattice is a set of stable matchings," *Journal of Combinatorial Theory, Series A*, vol. 37, no. 3, pp. 353–356, 1984.

[17] T. Mai and V. V. Vazirani, "Finding stable matchings that are robust to errors in the input," In arXiv, 2018.

[18] ——, "A natural generalization of stable matching solved via new insights into ideal cuts," In arXiv, 2018.