2016

Structure of the Stable Marriage and Stable Roommate Problems and Applications

Joe Hidakatsu
University of South Carolina

Structure of the Stable Marriage and Stable Roommate Problems and Applications

by

Joe Hidakatsu

Bachelor of Science
University of Michigan 2014

Bachelor of Science in Engineering
University of Michigan 2014

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Arts in

Mathematics

College of Arts and Sciences

University of South Carolina

2016

Accepted by:

Eva Czabarka, Director of Thesis

Laszlo Szekely, Reader

Lacy Ford, Senior Vice Provost and Dean of Graduate Studies

Acknowledgments

First, I thank my advisor, Eva Czabarka, for taking her time to help me with this thesis. She always kept both my short term and long term goals in mind. Next, I thank Laszlo Szekely for being my second reader, and George McNulty for answering various questions I had related to formatting. I also thank Ognian Trifonov for his clear responses to the many questions I had about deadlines. Next, I thank all of the faculty and graduate students within the mathematics department for providing a great workplace and learning environment over the last couple of years. I am also grateful for my family for their continued support throughout my life. Finally, I thank the SC smash community for providing a fun environment during my time away from my work. The SC smash group was a large part of my social life during my stay in South Carolina, and they certainly made my time here more enjoyable.

Abstract

The well-known Gale-Shapley algorithm is a solution to the stable marriage problem, but always results in the same stable marriage, regardless of how the algorithm is executed. Robert Irving and Paul Leather constructed the rotation poset, whose downward closed sets are in one-to-one correspondence with the set of stable marriage assignments. We discuss how to use the rotation poset to find the k-optimal matching, and prove that a k-optimal matching is the same as a minimum regret matching for high enough k. Finally, Dan Gusfield defines the rotation poset for the stable roommate problem, and uses it to efficiently enumerate all stable roommate assignments.

Table of Contents

List of Figures

Chapter 1

Introduction

Consider that we have a group of men and women, and we are tasked with matching pairs of men and women in a "good" way. That is, we wish to match them in such a way that the pairs are unlikely to break up. Each man ranks all the women as an ordered list, according to their preference, and vice versa. For a matching to be good, there should be no choice of man and woman who would both prefer each other to their current match. Otherwise, that man and woman would agree to drop their current match in favor of each other, resulting in a breakup. We call a matching without these pairs a stable matching. This problem is called the stable matching problem. David Gale and Lloyd Shapley proved in 1962 that there always exists a stable matching regardless of how each man and woman ranks the opposite sex. They also constructed an algorithm which finds a stable matching. Gale and Shapley were awarded the 2012 Nobel Prize in Economics for their work in this area. One application in economics involves distributing resources among people in a "good" way. Gale and Shapley's algorithm is also currently used to match medical students to residency programs in the National Resident Matching Program (*How the Matching Algorithm Works* 2016). The Gale-Shapley algorithm also provides an "optimal" solution to one of the two groups involved. For example, in the application just mentioned, the algorithm results in a matching that is optimal for the students. This means that each student is matched with his or her best possible match out of all stable matchings which may exist.

However, not all stable matchings are necessarily good matchings. Consider a scenario in which each man is matched to his first choice and each woman is matched to her last choice. Such a matching is considered to be stable, since no man would prefer a different partner, but from the women's point of view, the situation is not ideal. Thus, a matching which favors only one of the two groups is not always desirable, even if stable. We examine a mathematical structure which allows us to obtain a good grasp on all of the possible stable matchings. This structure leads to being able to calculate the matching with the best "average happiness" efficiently. This structure, discovered by Robert W. Irving and Paul Leather, is mathematically more pleasing (because of its size and structure) than a set of rankings which do not give any insight to what the stable matchings are.

A generalization to the stable matching problem is called the stable roommate problem. Instead of having two groups being matched to each other, there is only one group and we want to pair up people within that group. The application in the name of the problem is to match roommate pairs in a group of students. Dan Gusfield generalizes the structure results of Irving and Leather to the stable roommate problem. Gusfield uses this structure to provide a way to list all possible stable matchings in a relatively efficient manner. However, depending on the number of participants in the matching process, the number of possible stable matchings can be an extremely large number, so finding all stable matchings may take a long time for the simple reason that many stable matchings may exist. Since the stable roommate problem and its structure are generalizations of the stable marriage problem, Gusfield's method can also enumerate all of the stable marriages in a given problem. Unfortunately, a stable roommate assignment does not always exist. Research has been done in finding a suitable assignment in the case that no stable assignments exists. For example, Abraham, Biró, and Manlove (2006) discuss the problem of finding a matching with the least amount of pairs violating the definition of stability. This research, however, will not be discussed in this thesis.

Overall, the structure results of Irving, Leather, and Gusfield can lead to the ability to find the "right" stable matching for the application in mind, noting that not all stable matchings are appropriate. For example, the situation discussed above which favors one particular group, or possibly a stable matching in which all but one person has a match they are happy with, but that one person gets paired with his or her last choice.

Chapter 2

The Structure of the Stable Marriage Problem

2.1 Basic Definitions and the Gale-Shapley Algorithm

We assume the reader has basic knowledge of graph theoretic definitions. Such definitions can be found in (Diestel 2010) or any other graph theory text. We will begin with a structure result of Irving and Leather (1986) on the stable matching problem, a well-known problem in graph theory.

Definition 2.1. An *instance* of the stable matching problem (SMP) is given by two sets M and W of finite cardinality along with a total ordering $<_m$ of some set $W_m \subseteq W$ for every $m \in M$, and a total ordering $<_w$ of some set $M_w \subseteq M$ for every $w \in W$. We denote this instance of SMP by $(M, W, (<_v)_{v \in M \cup W})$.

We think of the set M as a set of men, the set W as a set of women, and the total orders as a ranking of the members of the opposite gender. Notice that not all members of the opposite sex must be ranked. We often think of these ranking as lists, with the most preferred people at the front of the lists (the smallest elements in the ordering), and will call them *preference lists*. If a person a does not rank a person b, then this means that a does not want to be matched with b. If $x <_v y$, then v prefers x to y. For a set of preference lists $\mathcal{L}$ and $v \in M \cup W$, we will use $\text{first}_{\mathcal{L}}(v)$, $\text{second}_{\mathcal{L}}(v)$, and $\text{last}_{\mathcal{L}}(v)$ to denote the first, second, and last person on the list of v, respectively (where first is most preferred and smallest in the ordering). We omit the subscript $\mathcal{L}$ when unambiguous.

Definition 2.2. Let $(M, W, (<_v)_{v \in M \cup W})$ be an instance of SMP. Then, let G be a bipartite graph with bipartition (M, W), with the edges of G containing the allowed pairs in a matching. We call G the *associated graph* or *underlying graph* of the instance of SMP . A *matching* A of G is a set of edges in G such that for all $e_1, e_2 \in A$, either $e_1 = e_2$, or e_1 is not incident to e_2 in G. For $v \in M \cup W$, let v_A denote the vertex that v is matched to in A (if that vertex exists). If G is an associated graph for an instance of SMP, then we impose another restriction on the definition of matching: for all $v_1v_2 \in A$, v_1 is on the list of v_2 and v_2 is on the list of v_1.

Definition 2.3. For an instance $(M, W, (<_v)_{v \in M \cup W})$ of SMP with associated graph G, a matching $A \in E(G)$ is *stable* if there does not exist an edge $mw \in E(G) \setminus A$ in which the following two conditions hold:

1. m is unmatched in A, or $w <_m m_A$
2. w is unmatched in A, or $m <_w w_A$

In terms of men and women, a stable matching is a matching in which there is no man-woman pair allowed by the graph that both prefer each other to their current partner. The following notation will be used throughout this thesis.

Definition 2.4. For an instance $(M, W, (<_v)_{v \in M \cup W})$ of SMP, $m \in M$, $w \in W$, let $r_m(w)$ be the *rank* of w in m's preference list, and $r_w(m)$ be the rank of m in w's preference list, where rank is the position in the preference list. For example, $r_m(\text{first}(m)) = 1$.

It is well known that for any instance of SMP, a stable matching exists. In the case that $|M| = |W|$ and $G = K_{n,n}$, the Gale-Shapley algorithm gives a stable matching that is man-optimal. That is, the Gale-Shapley algorithm returns a matching such that for each $m \in M$ matched with $w \in W$, there does not exist a matching in which m is matched with a vertex w' where $w' <_m w$. This algorithm, however, only gives one specific type of stable matching, and does not give much insight into the other stable matchings which may exist. This motivates a structure result on the set of all stable matchings. Pittel (1989) shows that the expected number of stable matchings is asymptotic to $e^{-1} n \ln n$, and Balinski and Ratier (1997) show that the maximum number of stable matchings is exponential. Thus, having a structure on the stable matchings rather than an enumerated list, is desired.

For the remainder of the chapter, unless specifically stated otherwise, assume that $(M, W, (<_v)_{v \in M \cup W})$ is the relevant instance of SMP with associated graph $K_{n,n}$, and all preference lists are full. That is, $|M| = |W| = n$, and all pairs are allowed in a matching. First, we must present the Gale-Shapley algorithm.

Algorithm 2.1. Each man and woman begins as not engaged.

1. Choose a man m who is not engaged. This man m proposes to the most preferred woman w on m's list who m has not yet proposed to.

2. If w is not engaged, or if w is currently engaged to someone less preferable to m, then w tentatively accepts the proposal from m. We say m and w are *engaged*. If w accepts m's proposal and w was previously engaged to m', then w and m' are no longer engaged.

3. Repeat steps 1 and 2 until all men are either engaged or have proposed to all women.

Note the following obvious properties of this algorithm.

Proposition 2.1.

1. Each man $m \in M$ proposes to the women on m's list in order, starting with the most preferred woman, until the algorithm terminates. So, as the algorithm progresses, the woman that m is engaged to can only get worse for m.

2. For each $w \in W$, as the algorithm progresses, the man that w is engaged to can only get better for w.

Theorem 2.5. *(Gale and Shapley 1962) The Gale-Shapley algorithm terminates and results in a perfect stable matching A.*

Proof. First, we show that the algorithm terminates, and results in a perfect matching. Suppose that the algorithm terminates, and there is a man $m \in M$ who is not engaged at the end of the algorithm. This means that every woman rejected m. For a woman to reject m, she needs to be engaged, which means that every woman is engaged. Therefore, by the Pigeonhole Principal, every man is engaged, contradicting that m is not engaged. So, if the algorithm terminates, it results in a perfect matching A. The algorithm terminates simply because no man proposes to the same woman twice, and so the termination condition in step 3 is clearly achieved.

We must now show that A is stable. Suppose for the sake of contradiction that there exists $m \in M, w \in W, mw \notin A$ such that $w <_m m_A$ and $m <_w w_A$. Then, by Proposition 2.1 (1), m must have proposed to w before proposing to m_A. So, since m will not propose to the next person unless he is rejected, we see that w rejected m. So, by Proposition 2.1 (2), $w_A <_w m$, contradicting our assumption. Therefore, A is stable. $\square$

Consider the example shown in Figure 2.1. This example will be used as we continue to work towards the structure result.

1: 3 1 5 7 4 2 8 6	1: 4 3 8 1 2 5 7 6
2: 6 1 3 4 8 7 5 2	2: 3 7 5 8 6 4 1 2
3: 7 4 3 6 5 1 2 8	3: 7 5 8 3 6 2 1 4
4: 5 3 8 2 6 1 4 7	4: 6 4 2 7 3 1 5 8
5: 4 1 2 8 7 3 6 5	5: 8 7 1 5 6 4 3 2
6: 6 2 5 7 8 4 3 1	6: 5 4 7 6 2 8 3 1
7: 7 8 1 6 2 3 4 5	7: 1 4 5 6 2 8 3 7
8: 2 6 7 1 8 3 4 5	8: 2 5 4 3 7 8 1 6
male preference lists	**female preference lists**

Figure 2.1 (Irving, Leather, and Gusfield 1987) Male and female preference lists (total orders).

Execution of the Gale-Shapley algorithm on this example provides the following stable matching in pairs from $M \times W$:

$$\{(1,3),(2,1),(3,7),(4,5),(5,4),(6,6),(7,8),(8,2)\}.$$

See Appendix A for details on how the algorithm results in these pairs. The appendix also discusses the creation of shortlists, described in the next section, but can be read the first time by ignoring the discussion of removing people from lists. We will need the following useful results. Propositions 2.2 and 2.3 were likely first proved by Gale and Shapley, and proofs can be found in the references in parentheses.

Proposition 2.2. (Gusfield and Irving 1989) In any stable marriage, no man receives a better choice than the one that he receives as a result of the Gale-Shapley algorithm. Thus, all executions of the Gale-Shapley algorithm result in the same stable matching, which we call the male-optimal matching.

Proof. Let A be a matching resulting from the Gale-Shapley algorithm. For $m \in M$, let $\alpha(m)$ denote the most preferred match for m out of all matches that appear in some stable matching. Suppose for the sake of contradiction that in A, there is some $m \in M$ such that $m_A \neq \alpha(m)$. Then, by Proposition 2.1, in the execution of the Gale-Shapley algorithm, there is some man m who is rejected by his optimal partner over all possible stable matchings, $\alpha(m)$. Let m be the first such man, during the execution of the algorithm, who is rejected by $\alpha(m)$, and suppose that $\alpha(m)$ rejects m because $\alpha(m)$ is engaged to m'. Also, let O be a stable matching in which m is matched with $\alpha(m)$.

First, $r_{\alpha(m)}(m') < r_{\alpha(m)}(m)$ by our assumption. Moreover, $r_{m'}(\alpha(m)) < r_{m'}(\alpha(m'))$ because m was the first man rejected by his optimal pair, so m' has not yet been rejected by $\alpha(m')$ when he is engaged to $\alpha(m)$. Then, by definition of α, we have that $r_{m'}(\alpha(m)) < r_{m'}(\alpha(m')) \leq r_{m'}(m'_O)$. So, since $r_{m'}(\alpha(m)) < r_{m'}(m'_O)$ and $r_{\alpha(m)}(m') <$

$r_{\alpha(m)}(m)$, we contradict the stability of O. So, for all $m \in M$, $m_A = \alpha(m)$, which indicates that A is male-optimal. □

Proposition 2.3. (McVitie and Wilson 1971) In any stable marriage, no woman receives a poorer choice than the one she receives in the male-optimal solution.

Proof. Let O be the male-optimal matching obtained by the Gale-Shapley algorithm. Suppose, for the sake of contradiction that woman w is matched with a man m in a stable marriage A such that $r_w(m) > r_w(w_O)$. By definition of male-optimal, we know that $r_{w_O}(w) < r_{w_O}((w_O)_A)$. This contradicts the stability of A. Note that $w \neq (w_O)_A$, since $mw \in A$ and $m \neq w_O$. □

2.2 Shortlists

Now, we will present shortlists. The creation of shortlists is the first step in the algorithm of Irving and Leather (1986) that gives a partial order related to the set of all stable matchings. Consider the male-optimal solution given by the Gale-Shapley algorithm. During the Gale-Shapley algorithm, when $w \in W$ receives a proposal from m, remove all $m' \in M$ such that $m' >_w m$ from the list of w. Moreover, if $m' \in M$ was removed from the list of w in this process, then remove w from the list of m'. The resulting lists after execution of the Gale-Shapley algorithm are called the *male-optimal shortlists*, presented by Irving and Leather (1986). Figure 2.2 shows the shortlists in our example from Figure 2.1. See Appendix A for details on the construction of these shortlists.

The following statement is obvious.

Lemma 2.1. For all $w \in W$, while creating the male-optimal shortlists by running the Gale-Shapley algorithm, people can only be removed from the *end* of w's list. In particular, men are removed from w's list when w is matched to someone more preferable, and the Gale-Shapley match of w is never removed.

1: 3 1 5 7 4	1: 4 3 8 1 2
2: 1 3 4 8 7	2: 3 7 5 8
3: 7 4 3 1 2 8	3: 7 5 8 3 6 2 1
4: 5 8 6 1 4 7	4: 6 4 2 7 3 1 5
5: 4 2 8 7 3 6 5	5: 8 7 1 5 6 4
6: 6 5 7 4 3	6: 5 4 7 6
7: 8 6 2 3 4 5	7: 1 4 5 6 2 8 3
8: 2 7 1 3 5	8: 2 5 4 3 7
male shortlists	**female shortlists**

Figure 2.2 (Irving, Leather, and Gusfield 1987) Male and female male-optimal shortlists

Proposition 2.4. (Irving and Leather 1986) If x is not on y's shortlist, then there does not exist a stable matching in which x is matched to y.

Proof. First, assume $x \in M$ and $y \in W$, and let S be the matching resulting from the Gale-Shapley algorithm. Note that since y did not reject y_S, we know that y_S is on y's shortlist. Now, for sake of contradiction, assume that $xy \in A$, for a stable matching A. Since x is not on y's shortlist, $r_y(y_S) < r_y(x)$ by Lemma 2.1. This contradicts Proposition 2.3. So, x and y cannot be matched in a stable matching. We cover the case of when $x \in W$ and $y \in M$ by noting that a woman w is removed from a man m's list only when m is removed from w's list. □

Note, however that the converse is not true: that if A is a perfect matching in $K_{n,n}$ such that x is on y's shortlist and y is on x's shortlist whenever $xy \in A$, then A is stable. For example, if we begin with the male-optimal stable matching in our example $\{(1,3),(2,1),(3,7),(4,5),(5,4),(6,6),(7,8),(8,2)\}$ and "switch" 4 and 5 by matching man 4 with female 4 and man 5 with female 5, we obtain the matching $\{(1,3),(2,1),(3,7),(4,4),(5,5),(6,6),(7,8),(8,2)\}$. This matching is not stable, since male 4 prefers female 1, and female 1 prefers male 4 to their current match. From now on, we use first, second, and last to refer to the corresponding elements of the male-optimal shortlists unless otherwise specified.

Proposition 2.5. (Irving and Leather 1986) For $m \in M$, $w = \text{first}(m)$ if and only if $m = \text{last}(w)$. Moreover, m is matched with w in the male-optimal solution.

Proof. Let O be the male-optimal matching. By the construction of the shortlists of W, the last element of each woman's shortlist must be her match in the male-optimal solution. Now, suppose that there exists $w \in W$ on the shortlist of $m \in M$ such that $r_m(w) < r_m(m_O)$. Then, m proposed to and got rejected by w in the Gale-Shapley algorithm. Now, w did not remove m from her shortlist. So, $r_w(m) \leq r_w(w_O)$. However, w rejected m's proposal, so we have a contradiction. Thus, the shortlist of every man does not contain any woman better than m_O. $\square$

The results above show that the shortlists are a shorter version of the original total orders, but do not lose any information about the set of stable matchings. Note that the same process can be done by switching the roles of the men and women, to create female-optimal shortlists. Figure 2.3 shows the female-optimal shortlists in our example. These shortlists are created by taking the preference lists in Figure 2.1, reversing the roles of men and women, and running the Gale-Shapley algorithm. These shortlists are not shown by Irving and Leather, but are easily verifiable, since the first elements of the women's original lists form a permutation.

1: 3 1 5 7	1: 4 3 8 1 2 5 7
2: 6 1 3 4 8	2: 3 7 5 8 6 4
3: 7 4 3 6 5 1 2	3: 7 5 8 3 2 1 4
4: 5 3 8 2 6 1	4: 6 2 3 5 8
5: 4 1 2 8 7 3 6	5: 8 1 6 4 3
6: 6 2 5 7 8 4	6: 5 4 7 6 2 8 3
7: 7 8 1 6 2 3	7: 1 5 6 8 3 7
8: 2 6 7 1 8 3 4 5	8: 2 5 4 7 8 6
male shortlists	female shortlists

Figure 2.3 Male and female
female-optimal shortlists

2.3 Rotations

We now discuss rotations, introduced by Irving and Leather (1986), an object which allows us to walk along different stable matchings starting from the male-optimal solution. Assume for this section that the lists and orderings in consideration are for the shortlists when not specified.

Definition 2.6. A *rotation* with respect to a set of shortlists $\mathcal{L}$ is a sequence $\rho = (m_0, w_0), \ldots, (m_{k-1}, w_{k-1})$, $k > 1$, where $m_i \in M$ and $w_i \in W$ such that for every $0 \leq i < k$: $w_i = \text{first}(m_i)$, and $w_{i+1} = \text{second}(m_i)$ (where $i+1$ is mod k). These rotations are said to be *exposed* in the given shortlists.

Definition 2.7. Consider the matching where all pairs in the rotation ρ "rotate": m_i is matched with w_{i+1}, and any pairs not in the rotation are matched with their same partners. We say the men and women in ρ *rotate* their partners.

Irving and Leather (1986) show that a matching after rotating is also stable, which will be shown after the following definition. First, consider our example with shortlists shown in Figure 2.2. The three exposed rotations in the male-optimal shortlists are $\rho_1 = (1,3), (2,1)$, $\rho_2 = (3,7), (5,4), (8,2)$, and $\rho_3 = (4,5), (7,8), (6,6)$ (see Appendix A).

Definition 2.8. Given the rotation $\rho = (m_0, w_0), \ldots, (m_{k-1}, w_{k-1})$, for each $0 \leq i < k$, if each man x which follows m_{i-1} ($i-1$ is mod k) is removed from the shortlist of w_i, along with removal of the corresponding women in the shortlists of M, then the rotation is *eliminated*, and ρ is the *eliminating rotation* for any pair (m, w) such that w was deleted from the list of m, and m was deleted from the list of w. This results in new lists, which we still call shortlists.

Elimination of ρ_2 gives the shortlists in Figure 2.4. See Appendix A for details.

Note that after a rotation is eliminated, each man is matched with the first elements of the man's list, and each woman is matched with the last element of the

1: 3 1 5 7	1: 4 3 8 1 2
2: 1 3 4 8 7	2: 3 7 5
3: 4 3 1 2 8	3: 7 5 8 3 6 2 1
4: 5 8 6 1 4 7	4: 6 4 2 7 3
5: 2 8 7 3 6 5	5: 8 7 1 5 6 4
6: 6 5 7 4 3	6: 5 4 7 6
7: 8 6 2 3 4 5	7: 1 4 5 6 2 8
8: 7 1 3 5	8: 2 5 4 3 7
male shortlists	female shortlists

Figure 2.4 Male and female shortlists after elimination of ρ_2.

woman's list in the matching after rotating. Eliminating rotations can be done consecutively to construct different stable matchings, as long as rotations exist in the shortlists. We prove this in the same way as Irving and Leather (1986) below.

Lemma 2.2. (Irving and Leather 1986) If a set of shortlists $\mathcal{L}$ is obtained after a sequence of rotations, then the following two properties hold:

1. $w = \text{first}_{\mathcal{L}}(m)$ if and only if $m = \text{last}_{\mathcal{L}}(w)$ $(m \in M, w \in W)$

2. $w \in W$ is not in the list of $m \in M$ if and only if $r_w(m) > r_w(\text{last}_{\mathcal{L}}(w))$

Proof. If $\mathcal{L}$ is the original male-optimal shortlists, then the first property is Proposition 2.5, and the second property comes from the construction of the shortlists. So, using induction on the number of rotations eliminated, assume that the properties hold for $\mathcal{L}'$ (created by eliminating a sequence of rotations), and that $\mathcal{L}$ is obtained by eliminating a rotation from $\mathcal{L}'$.

To prove Property 1, first assume that m is not in the rotation. Then, $w = \text{first}_{\mathcal{L}}(m)$ if and only if $w = \text{first}_{\mathcal{L}'}(m)$, by our assumption. Then, $w = \text{first}_{\mathcal{L}'}(m)$ if and only if $m = \text{last}_{\mathcal{L}'}(w)$, by induction. Then, since w cannot be part of the rotation, we have that $m = \text{last}_{\mathcal{L}'}(w)$ if and only if $m = \text{last}_{\mathcal{L}}(w)$. Now, if m is part of the rotation, then it is easy to see that Property 1 holds from the definition of rotation elimination.

Now, for property 2, w is not in the list of m in $\mathcal{L}$ if and only if w was not in the list of m in $\mathcal{L}'$ or, w was eliminated from the list of m during the rotation elimination. This is true if and only if $r_w(m) > r_w(\text{last}_{\mathcal{L}'}(w))$ (induction), or $r_w(\text{last}_{\mathcal{L}}(w)) < r_w(m) \leq r_w(\text{last}_{\mathcal{L}'}(w))$. This is true if and only if $r_w(m) > r_w(\text{last}_{\mathcal{L}}(w))$, since $\text{last}_{\mathcal{L}'}(w) \geq_w \text{last}_{\mathcal{L}}(w)$. □

Theorem 2.9. *(Irving and Leather 1986) Eliminating a rotation from a set of shortlists results in a stable matching when each man is matched with the first woman on the man's shortlist.*

Proof. Assume a rotation R is eliminated from a set of shortlists $\mathcal{L}$ to produce a new set of shortlists $\mathcal{L}'$. Since we are proving a result about $\mathcal{L}'$, if lists are not specified, then first, second, and last refer to $\mathcal{L}'$. Suppose that for $m_1, m_2 \in M$, $\text{first}(m_1) = \text{first}(m_2)$. Then, by Lemma 2.2 (1), there is a $w \in W$ such that $\text{last}(w) = m_1 = m_2$. So, $m_1 = m_2$. Thus, if we show that no list is empty, we have a valid matching. It suffices to show that if no lists in $\mathcal{L}$ are empty, then there are no empty lists in $\mathcal{L}'$. For $m \in M$, the only way to remove $\text{first}_{\mathcal{L}}(m)$ from the list of m is to eliminate a rotation containing the pair $(m, \text{first}_{\mathcal{L}}(m))$. So, if R does not contain the pair $(m, \text{first}_{\mathcal{L}}(m))$, then m's list in $\mathcal{L}'$ is not empty. Otherwise, R contains the pair $(m, \text{first}_{\mathcal{L}}(m))$, and $\text{first}_{\mathcal{L}'}(m) = \text{second}_{\mathcal{L}}(m)$, so m's list in $\mathcal{L}'$ is nonempty in this case as well. Since $m \in M$ was arbitrary, all men's lists in $\mathcal{L}'$ are not empty. Thus, we have a perfect matching. Note that this also means that all women's lists are nonempty.

We must now show the matching is stable. Suppose $m \in M$ prefers $w \in W$ to his current match $\text{first}(m)$. Then, since $\text{first}(m)$ is the most preferred person on the list of m, w is not on the list of m. So, m is not on the list of w. So, by Property 2 of Lemma 2.2, $r_w(m) > r_w(\text{last}(w))$. Since $\text{last}(w)$ is the partner of w, we have shown that there cannot exist a pair preferring each other to their current partner, so indeed we have a stable matching. □

We list some other important results without proof below.

Theorem 2.10. *(Irving and Leather 1986) Every stable matching can be obtained by eliminating a sequence of rotations.*

Note that after eliminating a rotation, the resulting matching is worse for men and better for women (in terms of preference). Thus, Theorem 2.9 and Theorem 2.10 give the following Corollary.

Corollary 2.1. Eliminating all rotations from the male-optimal shortlists results in the female-optimal solution when pairing each man with the first woman on his list.

Proposition 2.6. (Irving and Leather 1986) Every pair (m, w) can appear in at most one rotation over all possible shortlists ((m, w) may appear in a rotation on more than one set of shortlists, but the rotation that (m, w) is a part of is always the same).

Proposition 2.7. (Irving and Leather 1986) If (m, w) belongs to a rotation, then in a set of shortlists obtained by a sequence of rotation eliminations, w is absent from m's list if and only if the rotation containing (m, w) has been eliminated.

Proof. Suppose that (m, w) belongs to a rotation R and w is absent from m's list. Suppose for the sake of contradiction that R has not been eliminated. Then, at some point, w must have switched partners to last(w), and $r_w(\text{last}(w)) < r_w(m)$. Assume that the shortlists of interest are the shortlists immediately after the aforementioned step. Then, since the first element of any man's list can only change when eliminating a rotation, we also see that $r_m(\text{first}(m)) < r_m(w)$. The stable matching resulting from the current set of shortlists pairs m with first(m) and w with last(w). Call this stable matching A. By Theorem 2.9, there also exists a stable matching B such that $mw \in B$. So, m and w both prefer their match in A over their match in B. Let M_A and W_A (respectively M_B and W_B) denote the sets of men and women who have more preferred partners in A than in B (respectively B than in A). Then, $m \in M_A$ and $w \in W_A$. Every man in M_A has an A-partner in W_B, because B is stable. Additionally, every woman in W_B has a B-partner in M_A, because A is stable. Therefore, $|M_A| \leq |W_B|$ and $|W_B| \leq |M_A|$, and so $|M_A| = |W_B|$. So, there are no B-pairings between M_A and W_A. This is a contradiction, since $mw \in B$. $\square$

It is not difficult to show from these results that for any stable matching A, there is a unique set of rotations R which must be eliminated in order to obtain shortlists corresponding to the stable matching A. Moreover, the order of elimination in R does not matter, provided that the rotations can be eliminated in that order. Consider the relation $\prec$ on the set of all rotations R, where $\pi \prec \rho$ if and only if π must be eliminated before ρ is exposed. Note that $\prec$ is a strict partial order. Irreflexivity holds, since any rotation ρ does not need to be eliminated before eliminating ρ. Transitivity and asymmetry clearly hold as well. The reflexive closure $\preceq$ given by $\pi \preceq \rho$ if and only if $\pi \prec \rho$ or $\pi = \rho$ is a non-strict partial order (the standard definition of partial order). A downward closed set $S \subseteq R$ is a set such that for all $\pi \in S$, if $\rho \in R$ and $\rho \prec \pi$, then $\rho \in S$. Then, the downward closed sets in $(R, \preceq)$ are in one-to-one correspondence with the set of stable matchings, which is shown in (Irving and Leather 1986). In particular, the downward closed sets indicate which set of rotations to eliminate. Constructing this poset gives rise to many applications.

Definition 2.11. The poset corresponding to an instance I of SMP, denoted $(R_I, \preceq_I)$, is its *rotation poset*. The subscript I will be omitted when unambiguous.

Irving and Leather also show that the construction of the rotation poset can be done in $O(n^2)$ time. Table 2.1 is a list of all rotations in our example, and Figure 2.5 is the Hasse diagram of the rotation poset. The calculations were done by Irving and Leather. The table of rotations also includes weights, which are defined in Chapter 4.

Table 2.1 List of rotations and weights

Rotation Name	Rotation	Weight
ρ_1	(1,3), (2,1)	0
ρ_2	(3,7), (5,4), (8,2)	1
ρ_3	(4,5), (7,8), (6,6)	2
ρ_4	(1,1), (6,5), (8,7)	-2
ρ_5	(2,3), (3,4)	-2
ρ_6	(4,8), (7,6), (5,2)	1
ρ_7	(3,3), (8,1)	1
ρ_8	(2,4), (5,8), (6,7)	0
ρ_9	(1,5), (5,7), (8,3)	-1
ρ_{10}	(3,1), (7,2), (5,3), (4,6)	0

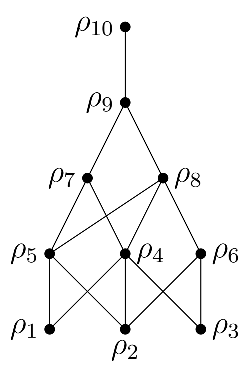


Figure 2.5 Hasse diagram of the rotation poset

Chapter 3

The General Case of the Stable Marriage Problem

There are many variations of the stable marriage problem. Manlove et al. (2002) shows several of these variations. We will focus on three of the most basic generalizations here, and they will be described below.

1. The case that the underlying graph is a not necessarily complete bipartite graph

2. The case that the preference lists are incomplete

3. The case that $|M| \neq |W|$

We will first explain what we mean by each case and then show without difficulty that case 1 and case 3 can both be modeled by case 2. What is meant by the last case is clear. The number of men and women are not the same, so the best we can hope for is to match all members of the smaller set. For the first case, we add another parameter to the SMP problem as a graph G such that G is bipartite with partition classes M and W. That is, we have an instance $(M, W, (<_v)_{v \in M \cup W}, G)$. Now, a stable matching in this setting is still a matching $A \subseteq E(G)$ such that there does not exist $mw \in E(G) \setminus A$ such that m prefers w to his partner in A (or m is unmatched) and w prefers m to her partner in A (or w is unmatched). The graph G is a structure which restricts the possible pairings in a matching. As an example, consider a man m and a woman w who live too far away from each other for a match to be pratical. Then, mw should not be an edge in G. Now, case 2 is when some men and women do not list certain people on their preference list, because they refuse to be matched with these people. It is clear that the first case can be modeled by the second case by simply removing all of the nonadjacent pairs in G in the preference lists, and adding edges to form a complete bipartite graph. The third case is also easily modeled by the second case by simply adding people with empty preference lists of the appropriate gender until $|M| = |W|$. Therefore, it is enough to consider case 2 in these three generalizations. Note that if w is on m's preference list and m is not on w'w preference list (or vice versa), then there can be no matching that matches m and w, so we may simply remove w from m's preference list without changing the possible matchings. Thus, we will assume in this case that m is on w's preference list if and only if w is on m's preference list.

In fact, we can always take a case 2 instance of SMP and transform it into the case of Chapter 2 where $|M| = |W|$, $G = K_{n,n}$, and all preference lists are full. This idea is made precise in Theorem 3.1 below. For the purpose of this theorem, a person y is on the preference list of x if and only if xy is an edge in the underlying graph of the instance of SMP. This allows for ease of notation, and from the discussion above, we do not lose any generality.

Theorem 3.1. *Let $I = (M, W, (<_v)_{v \in M \cup W}, G)$ be an instance of SMP with incomplete preference lists, where G contains precisely the edges given by the preference lists. Then, create a (standard) instance of SMP S with underlying graph $K_{n,n}$ by adding to the end of each incomplete preference list all missing members of the opposite sex, in any order. Then, the following statements are true:*

1. *If A_S is a stable matching in S, then $A_I := A_S \cap E(G)$ is a stable matching in I.*

2. *If A_I is a stable matching in I, then there exists a stable matching A_S in S such that $A_I = A_S \cap E(G)$.*

Proof. (1) Let $mw \in E(G) \setminus A_I$ where $m \in M$ and $w \in W$. We will show that mw does not violate the condition of stability of A_I.

Case 1: $mm_{A_S}, w_{A_S}w \in E(G)$. Then, $m_{A_I} = m_{A_S}$ and $w_{A_I} = w_{A_S}$, so from the stability of A_S, it cannot be the case that both $w <_m m_{A_I}$ and $m <_w w_{A_I}$, so the condition is not violated.

Case 2: $mm_{A_S} \in E(G)$, $w_{A_S}w \notin E(G)$. Then, $m_{A_I} = m_{A_S}$, and w is unmatched in A_I. Suppose, for sake of contradiction, that $w <_m m_{A_I}$. Note that we must have $m <_w w_{A_S}$ since $w_{A_S}w \notin E(G)$ and $mw \in E(G)$, so this contradicts the stability of A_S. Therefore, $w >_m m_{A_I}$. So, mw does not violate the condition of stability. The case that $mm_{A_S} \notin E(G)$ and $w_{A_S}w \in E(G)$ is symmetric.

Case 3: $mm_{A_S} \notin E(G)$, $w_{A_S}w \notin E(G)$. Then, $w <_m m_{A_S}$ and $m <_w w_{A_S}$. This contradicts the stability of A_S. So in fact, case 3 cannot happen.

(2) Let A_I be a stable matching of I, and let $U \subseteq V(G)$ be the set of vertices unmatched in A_I. Then, let B be a stable matching of U (with smaller preference lists which do not include $V(G) \setminus U$). We claim that $A_S := A_I \cup B$ is a stable matching. Let $mw \in E(K_{n,n}) \setminus A_S$ where $m \in M$ and $w \in W$. We wish to show that mw does not violate the condition of stability of A_S. First, assume that $mw \in E(G)$.

Case 1: $mm_{A_S}, w_{A_S}w \in E(G)$. Then, $m_{A_I} = m_{A_S}$ and $w_{A_I} = w_{A_S}$, so from the stability of A_I, it cannot be the case that both $w <_m m_{A_S}$ and $m <_w w_{A_S}$, so the condition is not violated.

Case 2: $mm_{A_S} \in E(G)$, $w_{A_S}w \notin E(G)$. Then, $m <_w w_{A_S}$. Suppose then that $w <_m m_{A_S} = m_{A_I}$. Since A_I is stable and w is unmatched in A_I, this means $m_{A_I} <_m w$, which is a contradiction, so $w \geq_m m_{A_S}$. Thus, mw does not violate stability in A_S. The case that $mm_{A_S} \notin E(G)$ and $w_{A_S}w \in E(G)$ is symmetric.

Case 3: $mm_{A_S}, w_{A_S}w \notin E(G)$. Then, m and w are unmatched in A_I, which contradicts the stability of A_I, so case 3 cannot happen.

Now, assume that $mw \notin E(G)$. It is only possible for $w <_m m_{A_S}$ if $mm_{A_S} \notin$

$E(G)$, so assume that $mm_{A_S}, ww_{A_S} \notin E(G)$. This means that m, w, m_{A_S}, w_{A_S} are all unmatched in A_I. Thus, by stability of B, mw can not violate the stability of A_S. $\square$

Theorem 3.1 essentially states that it is sufficient to only consider the case of Chapter 2. When an edge in $E(K_{n,n}) \setminus E(G)$ is assigned to a stable matching of S, we simply ignore that edge in the stable matching of I. This can lead to duplicate stable matchings of I when unmatched vertices in A_I are matched in different ways in A_S.

Now, we give an example of the third case. First, we do not extend the shortlists in order to show why Theorem 3.1 is necessary. The Gale-Shapley algorithm given in Chapter 2 will still result in a stable matching in the general case. For example, suppose $|M| = 3$ and $|W| = 4$, and consider the preference lists shown in Figure 3.1.

1: 2 3 1 4	1: 1 2 3
2: 1 3 2 4	2: 2 3 1
3: 1 2 3 4	3: 3 2 1
male preference lists	4: 1 2 3
	female preference lists

Figure 3.1 Male and female preference lists for an example with $|M| \neq |W|$

The Gale-Shapley algorithm executes to give the following matching (in $M \times W$): $A_1 := \{(1, 3), (2, 1), (3, 2)\}$. The shortlists are shown in Figure 3.2.

1: 3 1 4	1: 1 2
2: 1 3 2 4	2: 2 3
3: 2 3 4	3: 3 2 1
male shortlists	4: 1 2 3
	female shortlists

Figure 3.2 Male and female male-optimal shortlists for an example with $|M| \neq |W|$

The statement that every woman is matched with the last man on her shortlist does not hold here, since woman 4 is not matched with anyone. To solve this, we first convert to case 2 by adding an extra man with an empty preference list. Then, we may use Theorem 3.1 by adding to the end of each list all members of the opposite sex who are not in the original preference lists. Figure 3.3 shows the extended lists in our example. This leads to matchings with male 4 matched to female 4, which after intersecting with the original edges (defined by the preference lists), results in female 4 being unmatched.

1: 2 3 1 4	1: 1 2 3 4
2: 1 3 2 4	2: 2 3 1 4
3: 1 2 3 4	3: 3 2 1 4
4: 1 2 3 4	4: 1 2 3 4
male preference lists	female preference lists

Figure 3.3 Extending the preference lists to include man 4

Another stable matching exists: $A_2 := \{(1,1),(2,3),(3,2)\}$, which can be obtained by eliminating the rotation $(1,3),(2,1)$ from the male-optimal shortlists. Note that A_1 is clearly more beneficial to men than A_2, and A_1 is male-optimal if we extend the definition of male-optimal. That is, a matching is male-optimal if every man that can be matched in a stable matching is matched to his best possible partner among all stable matchings. In fact, we prove a known and useful result that all stable matchings cover the same set of vertices.

Proposition 3.1. Let $I = (M, W, (<_v)_{v \in M \cup W}, G)$ be an instance of SMP with possibly incomplete preference lists, where $xy \in E(G)$ if and only if x and y are on each other's lists. Then, if A_1 and A_2 are stable matchings, A_1 and A_2 cover the same set of vertices of G.

Proof. Let A_1 and A_2 be two stable matchings of G. Suppose that these matchings do not cover the same vertices. Then, without loss of generality, suppose there exists x_0 that is matched in A_1, but unmatched in A_2. Then, let x_1 be such that $x_0x_1 \in A_1$. Now, since x_0 is unmatched in A_2, x_1 must be matched in A_2, so let x_2 be such that $x_1x_2 \in A_2$. Then, $x_0x_1x_2$ is an alternating path between edges of A_1 and A_2. Continue this process to form an alternating path $x_0x_1 \ldots x_k$, where x_k is only matched in one of A_1 or A_2. Note that we will not result in a cycle, since x_0 is unmatched in A_2.

Now, $x_{k-2} <_{x_{k-1}} x_k$, because x_k is unmatched in one of the matchings, and x_{k-1} could match with x_k instead of x_{k-2}. If $k-2=0$, then this is a contradiction, since $x_k <_{x_{k-1}} x_{k-2}$, as x_{k-2} is unmatched in A_2. If $k>2$, then $x_{k-3} <_{x_{k-2}} x_{k-1}$, because if $x_{k-1} <_{x_{k-2}} x_{k-3}$, then $k-2$ would switch to $k-1$ in one of the matchings. We continue this process until we arrive at the conclusion that $x_0 <_{x_1} x_2$. This is a contradiction, since x_0 is unmatched in A_2. Therefore, we conclude that A_1 and A_2 must cover the same vertices. □

One other generalization that is worth mentioning is the many-to-one stable matching problem. We have two groups A and B such that for each $b \in B$, b can be matched with up to k_b people in group A. This generalization is being currently used to match medical students to residency programs in the National Resident Matching Program (*How the Matching Algorithm Works* 2016). In this case, A is the set of medical students, and B is the set of hospitals. The elements of A and B still rank the members of the other group as in the original problem. To define stability, we think of each element b as a list of k_b slots. Then, a matching is stable if there does not exist $a \in A$ and a slot s who both prefer each other to their current match. This problem can be specialized to an instance of SMP by simply adding k_b copies of b for all $b \in B$. These copies will all have the same lists as b did, and for each $a \in A$ that had b on a's list, b is replaced by the list of k_b copies of b in any order (consecutively and in the same position that b was). The proof of this generalization is straightforward and will not be shown.

Chapter 4

Applications of the Stable Marriage Structure

We begin with an application of the rotation poset. For this chapter, assume that $(M, W, (<_v)_{v \in M \cup W}, K_{n,n})$ is the relevant instance of SMP with full preference lists, and P is its rotation poset, unless stated otherwise. Consider the task of finding the "best" stable matching. There are many ways to try to define best, and two definitions which appear in (Manlove et al. 2002) are below.

Definition 4.1. Let A be a stable matching. Then, for $v \in V(G)$, we say $c_A(v) := r_v(v_A)$ is the *cost* of A for v. We denote $w(A) = \sum_{v \in V(G)} c_A(v)$ to be the *weight* of A, and $r(A) = \max_{v \in V(G)} c_A(v)$ to be the *regret* of A. An *egalitarian* matching is a matching whose weight is minimized, and a *minimum regret* matching is a matching whose regret is minimized.

The weight is just the sum of the cost of each person's match, so it is the average cost multiplied by $|V(G)|$. Thus, the weight is a measure of average happiness. Regret is a measure of the worst match in the matching. These definitions seem to be the most natural way to weigh a stable matching, but there is nothing inherit about these definitions being the "correct" way to measure a matching. Gusfield (1987) describes a way to find a minimum regret matching in $O(n^2)$ time (where $n = |M|$). Irving, Leather, and Gusfield (1987) present how to use the rotation poset to find an egalitarian matching in $O(n^4)$ time, and we will present the idea here. The following definition appears in (Irving, Leather, and Gusfield 1987).

Definition 4.2. Let $\rho = (m_0, w_0), \ldots, (m_{k-1}, w_{k-1})$ be a rotation. The *weight*, $w(\rho)$, of the rotation is

$$w(\rho) = \sum_{i=0}^{k-1}(r_{m_i}(w_{i+1}) - r_{m_i}(w_i)) + \sum_{i=0}^{k-1}(r_{w_i}(m_{i-1}) - r_{w_i}(m_i)),$$

where $i-1$ and $i+1$ are taken mod k.

Table 2.1 provides the weights of all the rotations in our running example.

Proposition 4.1. (Irving, Leather, and Gusfield 1987) Let A be a stable matching, and let R be the set of rotations corresponding to A. Then, $w(A) = w(O) + \sum_{\rho \in R} w(\rho)$, where O is the male-optimal solution.

Proof. It suffices to show that if S is a stable matching and S' is a stable matching resulting from eliminating a rotation $\rho = (m_0, w_0), \ldots, (m_{k-1}, w_{k-1})$ from S, then $w(S') = w(S) + w(\rho)$. For any edge $mw \in S$ such that (m, w) is not in ρ, we have that $mw \in S'$, so this pair does not affect any change in weight from S to S'. Now, for every $0 \leq i < k$, m_i is matched with w_i in S, and m_i is matched with w_{i+1} in S'. Also, w_i is matched with m_i in S, and w_i is matched with m_{i-1} in S'. Thus:

$$\begin{aligned} w(S') &= w(S) + \sum_{i=0}^{k-1} r_{m_i}(w_{i+1}) + r_{w_i}(m_{i-1}) - (r_{m_i}(w_i) + r_{w_i}(m_i)) \\ &= w(S) + w(\rho) \end{aligned}$$

□

Thus, finding an egalitarian matching is equivalent to finding the closed set of minimum weight in P. Intuitively, the male-optimal and female-optimal solutions should be "bad" matchings compared to other stable matchings. The male-optimal solution in our example has weight 55. It is apparent from the results stated previously that the female-optimal solution is a result of eliminating all rotations. Thus, we can calculate the weight of the female-optimal solution by adding the weight of the male-optimal solution and the weight of all rotations. We calculate the same weight of 55 for the female-optimal solution. However, by the egalitarian metric, the male-optimal and female-optimal solutions are not the worst: consider that eliminating rotations

ρ_1, ρ_2, and ρ_3 result in a stable matching with weight 58. On the other side of the spectrum, the closed subsets of minimum weight have weight -1, and $\{\rho_1, \rho_2, \rho_5\}$ is one such example, shown in (Irving, Leather, and Gusfield 1987). Eliminating these rotations results in an egalitarian matching of weight 57.

Though finding an egalitarian matching for an instance of SMP can be done in polynomial time, Feder (1992) shows that finding an egalitarian matching for the roommate problem (see the next chapter) is NP-complete.

Now, let us consider a more arbitrary metric on the set of stable matchings.

Definition 4.3. Let A be a stable matching and let k be a positive real number. Then, for $v \in V(G)$, we say $c_A(k, v) := r_v(v_A)^k$ is the *k-cost* of A for v. We denote $w(k, A) = \sum_{v \in V(G)} c_A(k, v)$ to be the *k-weight* of A. A *k-optimal* matching is a matching whose k-weight is minimized.

A k-optimal matching is just a matching in which the sum of the k'th powers of the ranks are minimized. In particular, when $k > 1$, there is a greater penalty for matching a person to somebody low on his or her preference list. Thus, when k is higher, a k-optimal matching is more similar to a minimum regret matching than to an egalitarian matching. When $k < 1$, there is a greater reward for matching a person to somebody low on his or her list. That is, there is a larger penalty for somebody changing from their first choice to their second choice than there is for someone changing from their penultimate choice to their last choice. We can find a k-optimal matching in the same way as we found an egalitarian matching.

Definition 4.4. Let $\rho = (m_0, w_0), \ldots, (m_{k-1}, w_{k-1})$ be a rotation. The *k-weight*, $w(k, \rho)$ of the rotation is

$$w(k, \rho) = \sum_{i=0}^{k-1} (r_{m_i}(w_{i+1})^k - r_{m_i}(w_i)^k) + \sum_{i=0}^{k-1} (r_{w_i}(m_{i-1})^k - r_{w_i}(m_i)^k),$$

We obtain a proposition analogous to Proposition 4.1. It is stated without proof. The proof is the same as the proof of Proposition 4.1.

Proposition 4.2. Let A be a stable matching, and let R be the set of rotations corresponding to A. Then, $w(k, A) = w(k, O) + \sum_{\rho \in R} w(k, \rho)$, where O is the male-optimal solution.

Thus, using the same algorithmic idea, we can find a k-optimal matching in polynomial time. The notion of high k being similar to minimum regret is made precise below. First, we need a lemma.

Lemma 4.1. For any $n \geq 2$, there exists $k' \geq 1$ such that for all integers a and b with $1 \leq b < a \leq n$, and for all $k \geq k'$, $a^k - b^k > (2n-1)(b^k - 1)$. In particular, $k' = \log(2n+1)/\log\left(\frac{n}{n-1}\right)$ is a k' which satisfies the property.

Proof. We strengthen the inequality to $a^k - b^k > 2nb^k$. Dividing both sides by b^k, this inequality becomes $\left(\frac{a}{b}\right)^k > 2n+1$. Given that $a > b$ and a and b are integers satisfying $1 \leq b < a \leq n$, it is easy to see that a/b is minimized when $a/b = n/(n-1)$. So, the inequality is satisfied if $\left(\frac{n}{n-1}\right)^k > 2n+1$. So, $k' = \log(2n+1)/\log\left(\frac{n}{n-1}\right)$ satisfies the result. $\square$

Proposition 4.3. For an instance of SMP, there exists $k' \geq 1$ such that for all $k \geq k'$, any k-optimal matching is a minimum regret matching. In particular, if $n \geq 2$ (where $n = |M|(= |W|)$), then $k' = \log(2n+1)/\log\left(\frac{n}{n-1}\right)$ is a k' which satisfies the property.

Proof. Let $n = |M|$. We assume $n \geq 2$, since SMP is trivial when $n = 1$. Let k' be as in Lemma 4.1 and suppose $k \geq k'$. Let A be a k-optimal matching, and suppose for sake of contradiction that A is not a minimum regret matching. Then, there exists another stable matching B such that $w(k, B) \geq w(k, A)$ and $r(A) > r(B)$. Suppose that $r(A) = a$ and $r(B) = b$ Then, $a > b$. Let $v \in M \cup W$ such that $r_v(v_A) = a$. Note that $c_A(k, v) = a^k$. For all $x \in M \cup W$, obviously, $c_A(k, x) \geq 1$, and $c_B(k, x) \leq b^k$. So, we obtain the following inequality.

$$\begin{aligned} w(k,B) - w(k,A) &= \sum_{x \in M \cup W, x \neq v} (c_B(k,x) - c_A(k,x)) + (c_B(k,v) - a^k) \\ &\leq (2n-1)(b^k - 1) + (b^k - a^k) \end{aligned}$$

So, by Lemma 4.1, $w(k,B) - w(k,A) < 0$, which is a contradiction. Therefore, A must be a minimum regret matching. □

So, by using an appropriately large enough k, we may still use the rotation poset to find a minimum regret matching. That is, if $n = |M| = |W|$, then we can set $k = \log(2n+1)/\log\left(\frac{n}{n-1}\right)$ and find a k-optimal stable matching.

We assumed in this chapter that $|M| = |W|$. Now, assume that we have incomplete preference lists, as in Chapter 3. Proposition 3.1 shows that all stable matchings cover the same set of vertices. Suppose that this set of vertices is $M' \cup W'$, where $M' \subseteq M$ and $W' \subseteq W$. Then, it makes sense in this case to define a k-optimal matching as a k-optimal matching on the instance of SMP with sets M' and W'. This optimizes the happiness of the people who can be matched, while ignoring the people who cannot hope to have a match.

Chapter 5

Generalization to the Roommate Problem

The stable roommate problem is similar to the stable marriage problem, except that there is no bipartition: any pair of vertices may be matched.

Definition 5.1. Let F be a set of finite cardinality. Then, $(F, (<_f)_{f \in F})$ is an *instance* of the stable roommate problem (SRP), where each $<_f$ is a total ordering of some set $F_f \subseteq F \setminus \{f\}$. Additionally, G, a spanning subgraph of $K_{|F|}$, is the *associated graph* or *underlying graph* for the instance of SRP, where $V(G) = F$. This graph G gives the allowed pairs in a matching.

We think of the total orderings as rankings of possible roommate candidates, and use the term *preference lists* as in the previous chapters. Similar to the discussion in Chapter 3, we may assume that $G = K_{|F|}$, $|F|$ is even, and all preference lists are full. We will assume these conditions for the rest of the discussion on the stable roommate problem. A stable roommate assignment has exactly the same definition as a stable marriage.

Definition 5.2. For an instance $(F, (<_f)_{f \in F})$ of SMP with associated graph G, a matching $A \in E(G)$ is *stable* if there does not exist an edge $f_1 f_2 \in G \setminus A$ in which the following two conditions hold:

1. f_1 is unmatched in A, or $f_2 <_{f_1} (f_1)_A$
2. f_2 is unmatched in A, or $f_1 <_{f_2} (f_2)_A$

Unlike the stable marriage problem, an instance of SRP may not contain a stable assignment. As an example, consider a set of four people $\{1, 2, 3, 4\}$ such that person

4 is the last choice for all other people, and person i most prefers person $i+1$ (mod 3) for $i \in \{1,2,3\}$. Then, in any perfect matching, there exists $i \in \{1,2,3\}$ who is matched to 4, and the other two people are matched to each other. In this case, $i-1$ (mod 3) would prefer i, and i would prefer $i-1$ over 4, so the matching cannot be stable.

We will illustrate results with the example used by Gusfield (1988) in Figure 5.1. Additionally, when discussing the roommate problem in general, we assume that $(F, (<_f)_{f\in F})$ is the instance of SRP unless otherwise stated.

1	7	2	6	8	5	3	4
2	4	6	5	3	8	1	7
3	5	2	1	7	4	6	8
4	1	7	3	6	5	8	2
5	7	1	8	4	6	2	3
6	7	3	8	4	5	1	2
7	2	8	4	3	5	6	1
8	4	2	3	5	6	7	1

Figure 5.1 (Gusfield 1988) Preference lists in the running example

In order to present the generalization of rotations, we must first use the algorithm in (Irving 1985) which finds a stable roommate assignment, if one exists, and otherwise, states that no stable assignment exists. Terminology used by Gusfield (1988) will be used. We will use first, second, and last as in the previous chapters. We will sometimes refer to the first element of a list as the *head* of that list. We will also use the notation $r_f(g)$ to denote the rank of g on f's list, as in the previous chapters.

Definition 5.3. At any point in the algorithm, a person f is *semi-engaged* to first(f) if and only if last(first$(f)) = f$. Any person who is not semi-engaged is *free*.

Algorithm 5.1. (Phase 1): Repeat the following steps:

1. If there is an empty list, end the algorithm, as there is no stable assignment.

2. If everyone is semi-engaged, then end phase 1.

3. Choose any free person f, and for each person k who is ranked below f on first(f)'s list, remove k from first(f)'s list and remove first(f) from k's list. Then, f becomes semi-engaged to first(f).

Definition 5.4. The set of lists after the execution of phase 1 is called the *phase 1 table*.

The phase 1 table in our example is shown in Figure 5.2. See Appendix A for details on the construction.

1	2	6	5	3	4		
2	6	5	3	8	1		
3	5	2	1	7	4	6	
4	1	7	3	6	5	8	
5	7	1	8	4	6	2	3
6	3	8	4	5	1	2	
7	8	4	3	5			
8	4	2	5	6	7		

Figure 5.2 (Gusfield 1988)
Phase 1 table in the running example

Irving (1985) proves that in the phase 1 table, if j is missing from i's list, then there are no stable assignments which pair i to j. In this sense, the phase 1 table is analogous to the male-optimal shortlists of the stable marriage problem. We present the proof here.

Proposition 5.1. (Irving 1985) In the phase 1 table, if j is missing from i's list, then there are no stable assignments which pair i to j.

Proof. We will prove this by induction on the number of iterations of phase 1. If no iterations occur, then the result is trivial. Now, suppose after $n-1$ ($n \geq 1$) iterations, if j is missing from i's list, then there are no stable assignments which pair i to j.

Now, suppose that we have run n iterations of phase 1, and that j is missing from i's list. If this occured before the n'th iteration, we are done by induction, so assume that j is removed from i's list in the n'th iteration. We may assume without loss of generality that i is removed from j's list before j is removed from i's list. We may assume then that i is removed from j's list when a person k becomes semi-engaged to j, where $j = \text{first}(k)$ and $k <_j i$. Suppose there is a stable assignment A such that $ij \in A$. Then, by induction, since $j = \text{first}(k)$ and anyone not on j's list after $n-1$ iterations cannot be matched with j, we know that $j <_k k_A$. So, we have a contradiction to the stability of A. □

Since there are no stable assignments when the phase 1 table has an empty list, we will always assume that the phase 1 table has no empty lists. Next, we must present rotations. The following definitions should look similar to the definitions used in the stable marriage problem. If $e_i \in F$, then $h_i := \text{first}(e_i)$ and $s_i := \text{second}(e_i)$.

Definition 5.5. An *exposed rotation* R in a table T is an ordered subset of people $E = (e_0, e_1, \ldots, e_{r-1})$ such that $s_i = h_{i+1}$ for all $0 \leq i < r$, where $i+1$ is taken mod r. This is often written $R = (E, H, S)$, where H is the set of head entries of E in the corresponding order of E, and S is the set of second entries in the same order. Though H determines S, we use both H and S for ease of notation.

Definition 5.6. Let $R_1 = (E_1, H_1, S_1)$ and $R_2 = (E_2, H_2, S_2)$ be rotations, and let $E = ((e_1)_0, (e_1)_1, \ldots, (e_1)_{r-1})$ and $E_2 = ((e_2)_0, (e_2)_1, \ldots, (e_2)_{r-1})$. Then, R_1 and R_2 are equivalent if there exists an integer k ($0 \leq k \leq r-1$) such that $(e_1)_i = (e_2)_{i+k}$, $(h_1)_i = (h_2)_{i+k}$, and $(s_1)_i = (s_2)_{i+k}$ for all $0 \leq i < r$, where addition is taken mod r. This is an equivalence relation, and so we say that $R_1 = R_2$.

Definition 5.7. An exposed rotation $R = (E, H, S)$ is *eliminated* in the following manner: for every $s_i \in S$, every entry below e_i in s_i's list is removed, and then s_i is removed from k's list for each k who was removed from s_i's list.

Figure 5.3 shows the result of a rotation eliminated from the phase 1 table in the running example. The rotation elminated is $R = (E, H, S)$, where $E = \{1, 2, 3\}$, $H = \{2, 6, 5\}$, and $S = \{6, 5, 2\}$. See Appendix A for details. Next, we present the second and final phase of the algorithm.

1	6	5	3	4		
2	5	3				
3	2	1	7	4	6	
4	1	7	3	6	5	8
5	7	1	8	4	6	2
6	3	8	4	5	1	
7	8	4	3	5		
8	4	5	6	7		

Figure 5.3 (Gusfield 1988) The result of eliminating the rotation R from the phase 1 table

Algorithm 5.2. (Phase 2): Repeat the following steps:

1. If there is an empty list, end the algorithm, as there is no stable assignment.

2. If every person has precisely one entry on their list, end the algorithm. Matching each person with the the entry on their list results in a stable assignment.

3. Otherwise, eliminate a rotation (one does exist).

It is easy to see that during the second phase of the algorithm, all people remain semi-engaged to somebody at every step. We state this without proof.

Lemma 5.1. (Irving 1985) During phase 2, at every step, all people who have non-empty lists remain semi-engaged to somebody. That is, for all $f_1 \in F$, $\text{first}(f_1) = f_2$ if and only if $\text{last}(f_2) = f_1$.

From this lemma, we easily obtain the following.

Lemma 5.2. If no lists are empty, then for a table after any step of phase 2, the first elements of all the lists form a permutation of F.

Proof. Since no list is empty, it suffices to show that there does not exist $g, f_1, f_2 \in F$, $f_1 \neq f_2$ such that $\text{first}(f_1) = \text{first}(f_2) = g$. Suppose that there do exist such elements g, f_1, f_2. Then, $\text{last}(g) = f_1 = f_2$, which contradicts that $f_1 \neq f_2$. $\square$

Irving (1985) shows that this algorithm take $O(n^2)$ time. Henceforth, we shall refer to this algorithm as the *roommate finding algorithm* or RFA. We construct a decision tree for RFA based on the available choices of rotation eliminations at each step. Every vertex of the tree represents a table, with the root of the tree being the phase 1 table. Let this tree be D. Different paths on D may lead to the same assignment. Gusfield (1988) gives a nice figure of D for our running example shown in Figure 5.4.

Definition 5.8. If $R = (E, H, S)$ is a rotation, then we define $R^d := (S, E, E^r)$ to be the *dual* of R, where E^r is a cyclic rotation of E such that the first element of E^r is the second element of E (note $H^r = S$). Note that $R^{d^d} = R$. If R and R^d are both rotations in D, then they are called a *dual pair* of rotations. If R is a rotation in D and R^d is not a rotation in D, then R is called a *singleton* rotation. Using the rotation names in Figure 5.4, (R_4, R_5) and (R_2, R_6) are dual pairs, while R_1 and R_3 are singleton rotations. This is sated in (Gusfield 1988).

The following are important results for our purposes.

Proposition 5.2. (Gusfield 1988) All stable roommate assignments can be reached by an execution of the algorithm along some path of D.

Proposition 5.3. (Gusfield 1988) Every path from the root of D to a leaf contains every singleton rotation and exactly one of each dual pair of rotations.

Proposition 5.4. (Gusfield 1988) Two different paths of D containing the same set of rotations leads to the same table.

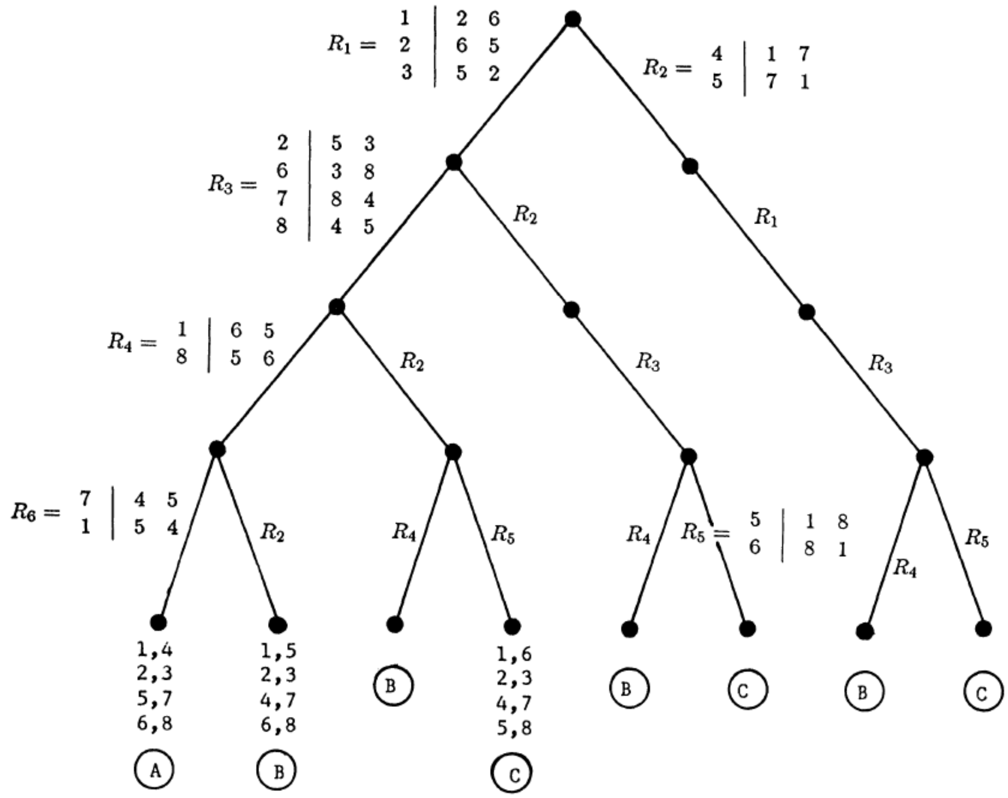


Figure 5.4 (Gusfield 1988) The decision tree D. There are three stable assignments: A, B, and C. In the rotations, the three columns represent E, H, and S, in that order.

These three results give many of the necessary steps in proving the main structure theorem below. We present a couple of the key lemmas in proving the correctness of RFA. These lemmas appear in (Gusfield 1988), but were first proved in (Irving 1985).

Lemma 5.3. (Irving 1985) If T is a table (during phase 2) where no list is empty, and at least one person has more than one entry, then there is a rotation exposed in T.

Proof. Suppose without loss of generality that a person e_0 has more than one entry. That is, h_0 and s_0 exist. Lemma 5.2 shows that the head entries are a permutation of F. Thus, there exists a person e_1 such that s_0 is the head of e_1. We will show that e_1 must have at least 2 people on his or her list. Suppose not. Then, $\text{first}(e_1) =$

$\mathrm{last}(e_1) = s_0$, and so by Lemma 5.1, $\mathrm{first}(s_0) = \mathrm{last}(s_0) = e_1$ (e_1 and s_0 are both semi-engaged to each other). Therefore, e_1 is the only person on s_0's list. This contradicts that e_0 must be on s_0's list. Thus, the triples $(e_0, h_0, s_0), (e_1, h_1, s_1)$ exist, where $h_1 = s_0$. We similarly build this sequence $(e_0, h_0, s_0), (e_1, h_1, s_1), \ldots, (e_k, h_k, s_k)$ until $h_k = h_i$ for some $i < k$. This must happen, since $|F|$ is finite. Then, our exposed rotation is $(e_i, h_i, s_i), \ldots, (e_{k-1}, h_{k-1}, s_{k-1})$. □

Definition 5.9. We say a stable assignment A is *contained* in a table T if for all $ij \in A$, j is on i's list and i is on j's list in T.

Lemma 5.4. (Irving 1985) Let $R = (E, H, S)$ be an exposed rotation in a table T, and let A be a stable assignment contained in T. Then, if $e_i \in E$ and (e_1, h_1) is a pair in A, then (e_i, h_i) must also be a pair in A for all $e_i \in E$.

Proof. We assume that (e_1, h_1) is a pair in A, and we will show that (e_0, h_0) is also a pair in A. This argument will extend to show that all (e_i, h_i) must be pairs in A. Suppose for the sake of contradiction that (e_0, h_0) is not a pair in A. Then, $s_0 <_{e_0} (e_0)_A$, since $s_0 = h_1$ is matched with e_1. Next, since the head of e_1 is h_1, we have $\mathrm{last}(h_1) = e_1$. So, since $h_1 = s_0$, we have $\mathrm{last}(s_0) = e_1$. Moreover, since s_0 is on e_0's list, e_0 is on s_0's list. So, $e_0 <_{s_0} e_1$. This contradicts the stability of A. □

The significance of this lemma is that if, for example, we are interested in finding a stable assignment in which e_i is not paired with h_i, we can eliminate a rotation which contains (e_i, h_i, s_i) without losing stable assignments of interest. The last key lemma is that if there is a rotation $R = (E, H, S)$ and a stable assignment in T which pairs e_i and h_i for some $e_i \in E$, then there is also a stable assignment in T which does not pair e_i and h_i. Therefore, by Lemma 5.4, eliminating a rotation does not remove all possible stable assignments from the table. We construct a poset similar to the rotation poset of the stable marriage problem. To do this, we use the following result.

Lemma 5.5. (Gusfield 1988) Suppose p is a person who must be removed from the list of q for a rotation R to be exposed. Then, there exists a unique rotation R' such that R' appears before R on every path that contains R, and R' is the only rotation in which elimination removes p from q's list and that R' appears before R on every path that contains R.

Definition 5.10. If p must be removed from the list of q for a rotation R to be exposed, and if R' is the unique rotation of Lemma 5.5, then R' *explicitly precedes* R.

The transitive closure of explicit precedence is clearly a partial order, and we again call this partial order the *rotation poset* of the instance of SRP. In the next chapter, we describe how to construct the rotation poset efficiently. Figure 5.5 shows the rotation poset of our running example. Calculations were done by Gusfield (1988).

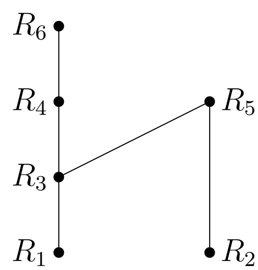


Figure 5.5 Hasse diagram of the rotation poset for the roommate problem example

Now, we state the main structure theorem for the roommate problem.

Theorem 5.11. *(Gusfield 1988) There is a one-to-one correspondence between stable roommate assignments and the downward closed sets in the rotation poset which contain every singleton rotation and exactly one of each dual pair.*

In fact, the set of rotations in the downward closed set describes which rotations need to be eliminated to produce the stable assignment.

5.1 Relationship Between the Posets of SMP and SRP

In roughly the same way as in Theorem 3.1, an instance of SMP can be extended to an instance of SRP. Consider the example of Figure 3.3. Figure 5.6 shows the preference lists of the generalizion of this example to the roommate problem by adding all members of the same sex to the end of each person's list (women 1, 2, 3, and 4 are relabled to be people 5, 6, 7, and 8, respectively).

1: 6 7 5 8 2 3 4
2: 5 7 6 8 1 3 4
3: 5 6 7 8 1 2 4
4: 5 6 7 8 1 2 3
5: 1 2 3 4 6 7 8
6: 2 3 1 4 5 7 8
7: 3 2 1 4 5 6 8
8: 1 2 3 4 5 6 7

Figure 5.6
Generalization of the example in Figure 3.3 to the roommate problem

Figure 5.7 shows the male-optimal shortlists of the SMP example, and the phase 1 table of the corresponding instance of SRP.

male shortlists	female shortlists	phase 1 table
1: 7 5 8	5: 1 2	1: 7 5
2: 5 7 6 8	6: 2 3	2: 5 7 6
3: 6 7 8	7: 3 2 1	3: 6 7
4: 8	8: 1 2 3 4	4: 8
		5: 1 2
		6: 2 3
		7: 3 2 1
		8: 4

Figure 5.7 Male-optimal shortlists and phase 1 table of the generalization.

In this section, suppose $I_{SM} := (M, W, (<_v)_{v \in M \cup W})$ is an instance of SMP, and I_{SR} is the corresponding instance of SRP in which all members of the same sex are placed on the end of each preference list. We also assume that the associated graph for I_{SM} is a complete bipartite graph, and the associated graph for I_{SR} is a complete graph. We first formalize the equivalence of I_{SM} and I_{SR}.

Proposition 5.5. A matching A is stable in I_{SM} if and only if A is stable in I_{SR}.

Proof. Let A be a stable matching in I_{SM}. Suppose that A is not stable in I_{SR}. Then, there exists $x, y \in M \cup W$ such that $r_x(y) < r_x(x_A)$ and $r_y(x) < r_y(y_A)$. Since A is stable in I_{SM}, it must be that x and y are of the same sex. Without loss of generality, suppose $x, y \in M$. Then, since $r_x(y) < r_x(x_A)$, this indicates that $x_A \in M$. This contradicts the fact that A is a matching in I_{SM}. So, A must be stable in I_{SR}.

Now, suppose that A is stable in I_{SR}. First, we must show that A is even a matching in I_{SM}: that there are no same-sex pairs in A. Suppose for the sake of contradiction that $m_1, m_2 \in M$ (without loss of generality) and $m_1 m_2 \in A$. Then, by the Pigeonhole Principle, there exists $w_1, w_2 \in W$ with $w_1 w_2 \in A$. Then, clearly, m_1 and w_1 both prefer each other to their current partner, contradicting the stability of A. We must now show that A is stable in I_{SM}. This is clear, since any pair violating the definition of stability in I_{SM} must also be present in I_{SR}. □

Next, we present a theorem which we will use to prove a few results in (Gusfield 1988). The intersection of the male-optimal and female-optimal shortlists means for each person $x \in M \cup W$, to intersect both of x's lists, while maintaining the original ordering. Note that during construction, the relative ordering does not change in any case.

Theorem 5.12. *The phase 1 table of I_{SR} is the intersection of the male-optimal and female-optimal shortlists of I_{SM}.*

Proof. Let x be in the list of $m \in M$ in the phase 1 table. We first show that $x \in W$. Suppose in contrast that $x \in M$. Then, since x is in the list of m, the last person on m's list $b := \text{last}(m)$ is a male, and b is semi-engaged to m. By Lemma 5.1, the head of b's list is m. So, all women were removed from the list of b, which indicates that all women removed b from their list. Since b is semi-engaged to a male, this can only happen if $\text{last}(z)$ is better than b for all women z. This is not possible by the Pigeonhole Principle, as the last elements are a permutation of all people (this is clear from Lemma 5.1 and Lemma 5.2) and b is a male. Thus, $x \in W$. By symmetry, we know that women only have men on their phase 1 table lists.

Note that running part of phase 1 when all free people f chosen are men (so that every man is semi-engaged to a woman) results precisely in the male-optimal shortlists (with the men at the end of each male's list). Similarly, running part of phase 1 when all free people f chosen are woman results in the female-optimal shortlists. Therefore, if y is on the list of z in the phase 1 table, then y is on the list of z in both the male-optimal and female-optimal shortlists.

Now, suppose without loss of generality that $w \in W$ is in the list of $m \in M$ in both the male-optimal and female-optimal shortlists. We wish to show that w is in the list of m in the phase 1 table. Suppose not. Then, w can be removed from the list of m in one of two ways.

Case 1: A free person $m' \in M$ was chosen during phase 1 such that $m' <_w m$, and so w removes m from her list ($\text{first}(m') = w$ at this point). If, in the male-optimal Gale-Shapley algorithm, m' proposes to w, then w would remove m from her list, which contradicts our assumption. So, suppose that m' does not propose to w. Then, in the male-optimal solution, m' is matched with someone w' such that $w' <_{m'} w$. So, w' must be in the list of m' in the phase 1 table. However, since $\text{first}(m') = w$ at some point, this is a contradiction, because $w' <_{m'} w$.

Case 2: A free person $w' \in W$ was chosen during phase 1 such that $w' <_m w$, and so m removes w from his list ($\text{first}(w') = m$ at this point). A symmetric argument to case 1 using the female-optimal algorithm proves this case.

Therefore, w is in the list of m in the phase 1 table. $\square$

One advantage of the phase 1 table then, is that the male-optimal and female-optimal solutions are both immediate. The male-optimal solution is the result of pairing each man with the first woman on his row in the phase 1 table, and the female-optimal solution is the result of pairing each woman with the first man on her row in the phase 1 table. So, we see from Figure 5.7 that the male-optimal solution has pairs $(1, 7), (2, 5), (3, 6), (4, 8)$, and the female-optimal solution has pairs $(1, 5), (2, 6), (3, 7), (4, 8)$.

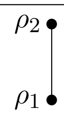
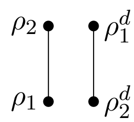
The rotation posets of SMP and SRP are very similar, and Gusfield (1988) gives some statements about this relationship without proof. We will present the proofs here. If $(m_1, w_1), \ldots, (m_k, w_k)$ is a rotation in I_{SM}, and $(e_1, h_1, s_1), \ldots, (e_k, h_k, s_k)$ is a rotation in I_{SR}, then we say these two rotations are equal if $m_i = e_i$ and $w_i = h_i$ for all $1 \leq i \leq k$. We will use the same symbol to denote two rotations that are equal in I_{SM} and I_{SR}. Let I_{SM} be the SMP instance of our most recent example. I_{SM} has two rotations: $\rho_1 = (1, 7), (2, 5)$ and $\rho_2 = (2, 7), (3, 6)$. Figure 5.1 shows the rotation posets for I_{SM} and I_{SR}, as well as the rotations needed to be eliminated to create each stable assignment.

First, we present a lemma which helps us prove the results in (Gusfield 1988).

Lemma 5.6. Every rotation in I_{SM} is a rotation in I_{SR}.

Proof. Let T be the phase 1 table of I_{SR} and SL be the male-optimal shortlists of I_{SM}. Let $R = (m_0, w_0), \ldots, (m_{k-1}, w_{k-1})$ be an exposed rotation in SL. For all $0 \leq i < k$, we have that m_i can be matched with w_i or w_{i+1} in some stable matching (since in SMP, the head of a male's shortlist is always part of a stable matching). So, w_i and w_{i+1} are on m_i's list in T, since y is not on x's row of T only when xy is not in any stable matching. Moreover, by Theorem 5.12, it must be that the head of m_i

Table 5.1 The rotation posets of I_{SM} and I_{SR}, and the set of rotations eliminated to create each stable assignment. The top row shows the three stable assignments.

Problem Instance	Rotation Poset	(1,7), (2,5), (3,6), (4,8)	(1,5), (2,7), (3,6), (4,8)	(1,5), (2,6), (3,7), (4,8)
I_{SM}		$\emptyset$	$\{\rho_1\}$	$\{\rho_1, \rho_2\}$
I_{SR}		$\{\rho_2^d, \rho_1^d\}$	$\{\rho_1, \rho_2^d\}$	$\{\rho_1, \rho_2\}$

is w_i and the second person on m_i's list is w_{i+1}. Therefore, since this holds for all i, we see that R is exposed in T. Therefore, the lemma holds for the rotations exposed in SL.

Next, by Theorem 5.12, the lists of males in T can be obtained from the lists of males in SL by removing people from the end of lists. This is because T can be obtained by running the male-optimal Gale-Shapley algorithm followed by the female-optimal Gale-Shapley algorithm (see the proof of Theorem 5.12). So, since in the female-optimal algorithm, people can only be removed from the end of the list of a male, we see that the male lists of T are obtained from removing people from the end of the lists in SL. Thus, for the lists of males, the effect of eliminating a rotation R in I_{SM} is the same as eliminating R in I_{SR}, but without the extra people at the end of the lists that appear in SL. These extra people will never be part of a rotation (since they cannot be matched with the person who owns the list), and so all rotations in I_{SM} are rotations in I_{SR}. □

Now, let the set of rotations of I_{SM} be $\mathcal{R}_{SM}$. Let I_W be the instance of SMP after switching the roles of M and W, and let $\mathcal{R}_W$ be the set of all rotations in I_W. Let $(\mathcal{R}_{SR}, \preceq)$ be the rotation poset of I_{SR}.

Lemma 5.7. $\mathcal{R}_{SM}$ and $\mathcal{R}_W$ both contain all singleton rotations and one of each dual pair of rotations in $\mathcal{R}_{SR}$.

Proof. Lemma 5.6 shows that all rotations in I_{SM} are rotations in I_{SR}. The proof of the lemma shows that we can eliminate $\mathcal{R}_{SM}$ in I_{SR}, with the head of each man being the match in the result from eliminating $\mathcal{R}_{SM}$ in I_{SM}. By Corollary 2.1, this matching is the female-optimal solution in I_{SM}. By Theorem 5.12, since the last person of a male's lists in the female-optimal shortlists is the male's female-optimal match, we see that eliminating $\mathcal{R}_{SM}$ in I_{SR} results in each man having only one woman on his list. Consequently, each woman only has one man on her list. So, by Theorem 5.11, since we have a complete run of RFA, $\mathcal{R}_{SM}$ contains all singletons and one of each dual pair of rotations. By symmetry, $\mathcal{R}_W$ contains all singletons and one of each dual pair of rotations. $\square$

Theorem 5.13. *$\mathcal{R}_{SM}$ and $\mathcal{R}_W$ partition $\mathcal{R}_{SR}$, and there is no relation between elements of $\mathcal{R}_{SM}$ and $\mathcal{R}_W$. Moreover, if $R \in \mathcal{R}_{SM}$, then $R^d \in \mathcal{R}_W$, and if $R \in \mathcal{R}_W$, then $R^d \in \mathcal{R}_{SM}$. Finally, $\mathcal{R}_{SR}$ does not contain any singleton rotations.*

Proof. The rotations of $\mathcal{R}_{SM}$ all involve men's lists and the elements of $\mathcal{R}_W$ all involve women's lists, so clearly $\mathcal{R}_{SM} \cap \mathcal{R}_W = \emptyset$. So, by Lemma 5.7, $\mathcal{R}_{SR}$ does not have any singleton rotations. Moreover, no element of $\mathcal{R}_{SM}$ can be related to an element of $\mathcal{R}_W$, since by construction, it is clear that any predecessors of elements in $\mathcal{R}_{SM}$ are in $\mathcal{R}_{SM}$, and any predecessors of $\mathcal{R}_W$ are in $\mathcal{R}_W$. By Lemma 5.6, $\mathcal{R}_{SM} \cup \mathcal{R}_W \subseteq \mathcal{R}_{SR}$, so we must show that $\mathcal{R}_{SR} \subseteq \mathcal{R}_{SM} \cup \mathcal{R}_W$. Let $R \in \mathcal{R}_{SM}$. Then, since R involves men's lists of women, R^d involves women's lists of men. So, since $\mathcal{R}_W$ contains one of each dual pair and only involves women's lists, $R^d \in \mathcal{R}_W$. Similarly, if $R \in \mathcal{R}_W$, then $R^d \in \mathcal{R}_{SM}$. So, since $\mathcal{R}_{SM}$ and $\mathcal{R}_W$ both contain one of each dual pair, we see that $\mathcal{R}_{SR} \subseteq \mathcal{R}_{SM} \cup \mathcal{R}_W$. $\square$

Proposition 5.6. Let C be a downward closed set in $\mathcal{R}_{SM}$, and let $C' = \mathcal{R}_{SM} \setminus C$. Also let C'^d be the duals of the elements in C'. Then, eliminating $C \cup C'^d$ in I_{SR} results in exactly the same stable matching as eliminating C in I_{SM}.

Proof. Let C be a downward closed set in $\mathcal{R}_{SM}$. Then, eliminating C in I_{SM} results in some stable marriage A. Now, the only way to move the head $\text{first}(f)$ of an entry f during phase 2 of SRP (assuming a stable matching exists) is by eliminating a rotation which contains the pair $(f, \text{first}(f))$. Therefore, there exists a set of rotations Z such that eliminating Z in I_{SR} results in the stable assignment A, and $Z \cap \mathcal{R}_{SM} = C$. Since Z must contain exactly one of each dual pair of rotations, Theorem 5.13 shows that $Z \setminus C = C^d$. So, $Z = C \cup C'^d$. $\square$

The converse also holds:

Proposition 5.7. Let C be a downward closed set in $\mathcal{R}_{SR}$ which contains exactly one of every dual pair of rotations. Then, eliminating C in I_{SR} results in exactly the same stable matching as eliminating $C \cap \mathcal{R}_{SM}$ in I_{SM}.

Proof. The statement is clear, since the only way to move the head $\text{first}(f)$ of an entry f during phase 2 of SRP is by eliminating a rotation which contains the pair $(f, \text{first}(f))$. $\square$

Thus, when generalizing an instance of SMP to SRP, we obtain two disjoint posets, one corresponding to the poset of the male-optimal SMP instance and one corresponding to the female-optimal instance of SMP, and it is enough to ignore one of the two posets when looking at the set of stable matchings.

Chapter 6

Applications of the Stable Roommate Structure

In this chapter, we present how to use the structure of the stable roommate problem to efficiently enumerate all stable assignments, presented by Gusfield (1988). This is done in $O(n^3 \log n + kn^2)$ time, where k is the number of stable assignments, and $n = |F|$. Though the number of stable assignments can be exponential, we see that enumeration only takes a polynomial time factor of the number of stable assignments. First, we will present the algorithm. The proof of its correctness can be found in (Gusfield 1988). We will discuss their time bound in detail here. For a rotation R and rotation poset Π, denote $\Pi(R)$ to be $\{R' | R' \preceq_\Pi R\}$.

As stated in the last chapter, it is sufficient to know the set of rotations eliminated to produce a stable assignment. That is, if the same set of rotations are eliminated in a different order, the same stable assignment is produced. We construct a binary tree $\mathcal{B}$. For the sake of discussion, we think of the root as the top of the tree, and children are said to be below their parent. As in the decision tree D, each vertex of the tree corresponds to a table after elimination of a set of rotations. We construct the tree as follows. We begin with a node representing the phase 1 table. Then, for each node x, if a rotation is exposed in that table, choose any exposed rotation R. That node has possibly 2 children: one from eliminating R from x, and one from eliminating all rotations in $\Pi(R^d)$ from x. We call the first child the *left* child, and the second child the *right* child. If R is a singleton, R only has a left child. Gusfield (1988) shows that the set of leaves after this tree is produced is the set of stable assignments, with no repetitions. Gusfield calls this the *dual enumeration method*. Figure 6.1 shows the binary tree $\mathcal{B}$ in our running example. Again, calculations were done by Gusfield (1988).

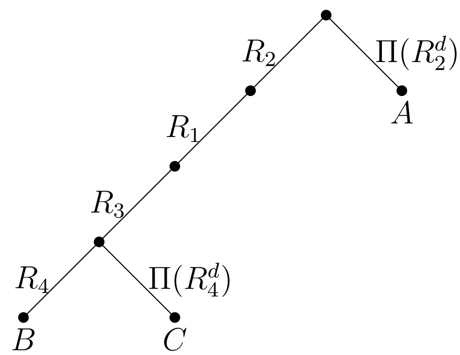


Figure 6.1 A binary tree $\mathcal{B}$ in our running example. A, B, and C are the three stable matchings shown in Figure 5.4

6.1 Time Complexity of Building the Rotation Poset

First, the "construction" of Π takes $O(n^3 \log n)$ time. In fact, it is the Hasse Diagram of Π which will be constructed. Since RFA runs in $O(n^2)$ time, and there can be at most double the amount of rotations that exist on any particular path of D (by taking duals), there are $O(n^2)$ rotations in Π. Now, consider a person p. We claim that if rotation R moves p's head before rotation R' does on a path P, then R preceds R' in Π. Note that Lemma 5.4 implies that any pair (e, h) can appear in at most one rotation. Thus, our claim is clear, by Lemma 5.5. So, we run the algorithm once to create a path P in D, and for every person p, we create a chain C_p of rotations that move the head of p while moving along P. Gusfield (1988) shows that in Π, only singleton rotations can precede singleton rotations. This is because all singleton rotations occur in every path of P, so a rotation R preceding a singleton must be on every path of P. This would imply that R^d is on no path of P. That is, R is a singleton. So, for each person p, C_p is a chain in which all singletons in the chain are consecutively below all dual rotations. Consider that path along R_1, R_2, R_3, R_4 (in order) in the tree D in our running example. Table 6.1 shows the chains C_p using this path. Notice that the singleton rotations R_1 and R_3 always precede any dual rotations.

Table 6.1 The chains C_p in our example

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
R_4 •	R_3 •						R_4 •
R_1 •	R_1 •	R_1 •	R_2 •	R_2 •	R_3 •	R_3 •	R_3 •

Lemma 6.1. (Gusfield 1988) Determining if a rotation R on P is a singleton or dual rotation takes $O(n^2)$ time.

Proof. Let T be the table such that R represents an edge from T to someone below T along P. Eliminate rotations from T until either R^d is exposed or R is the only rotation exposed. In the first case, R is clearly a dual rotation. In the latter case, we claim R is a singleton rotation. Suppose not. Then, T has 2 children in $\mathcal{B}$ (by examination of the proof of the correctness of the dual enumeration method, we may assume that T is in $\mathcal{B}$ in this case). The right child must contain R, since R is the only rotation exposed in T. This is a contradiction by the correctness of the dual elimination method, since this would result in eliminating both R and R^d. So, R must be a singleton rotation. This process of checking if R is a singleton or dual takes time $O(n^2)$, since RFA takes $O(n^2)$ time. □

So, we use binary search on each path C_p to determine where the first dual rotation in the chain is. By Lemma 6.1, we can find this point in $O(n^2 \log n)$ time per chain, since each C_p has length at most $n - 1$ (one rotation for each move of the head of p). So, since there are n chains, we can find all rotations in $O(n^3 \log n)$ time: We run RFA in $O(n^2)$ time, constructing the chains C_p during the process (we can add elements to the chains as we remove people from the heads of the tables), and then determine if each rotation is a singleton or dual in $O(n^3 \log n)$ time. We then list all singleton rotations and dual pairs in $O(n^2)$ time.

In the dual enumeration method, it is sufficient to have the Hasse diagram HD of Π. For the sake of the next lemma, we view HD and Π as directed graphs: there is an edge from u to v if and only if $u \prec v$.

Lemma 6.2. (Gusfield 1988), (Gusfield 1987) There exists a graph HD^* such that $HD \subseteq HD^* \subseteq \Pi$ and after finding all rotations, HD^* can be constructed in $O(n^2)$ time. Thus, it must be that HD^* has $O(n^2)$ edges.

6.2 Time Complexity of the Dual Enumeration Method

After HD^* is constructed, what remains is the dual enumeration method.

Theorem 6.1. *(Gusfield 1988) After construction of HD^*, each stable assignment can be found in $O(n^2)$ time per assignment.*

Proof. The edge from a parent to a left (right) child is called a left (right) edge. A vertex x of $\mathcal{B}$ is defined to be a left vertex if x is the root or if x is a left child. If x is a leaf and left child, then there is a unique maximal path P_x starting at x consisting of only left edges, and all of these P_x's are disjoint. The uniqueness is clear (one cannot go down another left node after going up a left node). Suppose there is a vertex v such that two of these maximal paths P_x and P_y go through v. Since the paths are maximal and only use left edges, everything below v on P_x and P_y must be the same. Everything above v must also be the same on P_x and P_y for the same reason that the paths themselves are unique. So, $P_x = P_y$.

If we add the time necessary to "run" each edge of $\mathcal{B}$ (eliminating the corresponding rotations), we arrive at the time for the dual enumeration method. Clearly, since

RFA runs in $O(n^2)$ time, each P_x runs in $O(n^2)$ time. Thus, we charge $O(n^2)$ time for each assignment which corresponds to a left child. We are then left with the time for each right edge. We claim that each right edge e either ends in a leaf or ends at the top of a path P_x for some x. Assume $e = yz$ where z is y's child. Suppose that e does not end in a leaf. If z is the top of a P_x path, then we are done. If not, then since z is not a leaf, z has a left child, so z must be on a path P_x for some x. However, since yz is a right edge, z must be the top of P_x, giving a contradiction. So, if a right edge ends in a leaf, we charge the time to the assignment the leaf corresponds to, and if a right edge ends at the top of a path P_x, then we charge the time to the assignment that x corresponds to. So, the time needed to run the right edges is $O(kt)$, where k is the number of stable assignments, and t is the time necessary to eliminate a set of rotations $\Pi(R^d)$.

We claim that $t = O(n^2)$. First, finding all the rotations in $\Pi(R^d)$ takes $O(n^2)$ time by simply searching backwards through the tree HD^*. We eliminate the rotations in $\Pi(R^d)$ in the following way. For each person $p \in F$, let p' be the highest person on p's list such that $p' = e_i$ and $p = s_i$ for some rotation $R = (E, H, S)$ in $\Pi(R^d)$. If such a person does not exist, then set $p' = 0$, where 0 is just a symbol. Then, to eliminate all rotations in $\Pi(R^d)$, for each $p \in F$, remove all people below p' on p's list, and for each person q removed from p's list, remove p from q's list. Do nothing if $p = 0$. First, p' can be found in $O(n^2)$ time since we can simply check each candidate for p' to see if $p = s_i$ in constant time (where $e_i = p'$), and keep track of the best candidates. There are $O(n^2)$ checks to make. Then, the removals clearly can be done in $O(n^2)$ time, since the entire size of the preference lists is $O(n^2)$. So, the total time for right edges is $O(kn^2)$. Adding this with the left edges, we still obtain $O(kn^2)$ time for the enumeration of all stable assignments. $\square$

Taking $O(n^3 \log n)$ time to find all rotations, $O(n^2)$ time to construct HD^*, and then taking $O(kn^2)$ time to enumerate all stable assignments gives a total of

$O(n^3 \log n + kn^2)$ time for the dual enumeration method. Figure 6.2 shows some binary tree $\mathcal{B}$ resulting from the dual enumeration method. There is no specific set of preference lists in mind. Just assume that $\mathcal{B}$ was constructed. Each left edge is dashed and labeled with the maximal path it belongs to, and each right edge is labeled with the corresponding assignment that time was charged to.

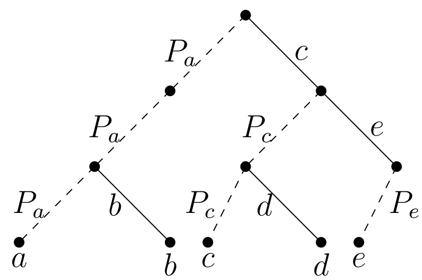


Figure 6.2 Labeled edges for some binary tree $\mathcal{B}$. Dashed edges are left edges.

Bibliography

Abraham, David J., Péter Biró, and David F. Manlove (2006). "Approximation and Online Algorithms: Third International Workshop, WAOA 2005, Palma de Mallorca, Spain, October 6-7, 2005, Revised Papers". In: ed. by Thomas Erlebach and Giuseppe Persinao. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. "Almost Stable" Matchings in the Roommates Problem, pp. 1–14. DOI: 10.1007/11671411_1.

Balinski, Michel and Guillaume Ratier (1997). "On stable marriages and graphs, and strategy and polytopes". In: *SIAM Rev.* 39.4, pp. 575–604. DOI: 10.1137/S0036144595294515.

Diestel, Reinhard (2010). *Graph theory*. Fourth. Vol. 173. Graduate Texts in Mathematics. Springer, Heidelberg. DOI: 10.1007/978-3-642-14279-6.

Feder, Tomás (1992). "A new fixed point approach for stable networks and stable marriages". In: *J. Comput. System Sci.* 45.2. Twenty-first Symposium on the Theory of Computing (Seattle, WA, 1989), pp. 233–284. DOI: 10.1016/0022-0000(92)90048-N.

Gusfield, Dan (1987). "Three fast algorithms for four problems in stable marriage". In: *SIAM J. Comput.* 16.1, pp. 111–128. DOI: 10.1137/0216010.

— (1988). "The structure of the stable roommate problem: efficient representation and enumeration of all stable assignments". In: *SIAM J. Comput.* 17.4, pp. 742–769. DOI: 10.1137/0217048.

Gusfield, Dan and Robert W. Irving (1989). *The stable marriage problem: structure and algorithms*. Foundations of Computing Series. MIT Press, Cambridge, MA, p. 11.

How the Matching Algorithm Works (Feb. 2016). URL: http://www.nrmp.org/match-process/match-algorithm/.

Irving, Robert W. (1985). "An efficient algorithm for the "stable roommates" problem". In: *J. Algorithms* 6.4, pp. 577–595. DOI: 10.1016/0196-6774(85)90033-1.

Irving, Robert W. and Paul Leather (1986). "The complexity of counting stable marriages". In: *SIAM J. Comput.* 15.3, pp. 655–667. DOI: 10.1137/0215048.

Irving, Robert W., Paul Leather, and Dan Gusfield (1987). "An efficient algorithm for the "optimal" stable marriage". In: *J. Assoc. Comput. Mach.* 34.3, pp. 532–543. DOI: 10.1145/28869.28871.

Manlove, David F. et al. (2002). "Hard variants of stable marriage". In: *Theoret. Comput. Sci.* 276.1-2, pp. 261–279. DOI: 10.1016/S0304-3975(01)00206-7.

McVitie, D. G. and L. B. Wilson (1971). "The stable marriage problem". In: *Comm. ACM* 14, pp. 486–490.

Pittel, Boris (1989). "The average number of stable matchings". In: *SIAM J. Discrete Math.* 2.4, pp. 530–549. DOI: 10.1137/0402048.

Appendix A

Execution of Algorithms

We will present the execution of the Gale-Shapley algorithm and the construction of the shortlists for the example in Chapter 2. The original preference lists are shown in Figure A.1.

1: 3 1 5 7 4 2 8 6	1: 4 3 8 1 2 5 7 6
2: 6 1 3 4 8 7 5 2	2: 3 7 5 8 6 4 1 2
3: 7 4 3 6 5 1 2 8	3: 7 5 8 3 6 2 1 4
4: 5 3 8 2 6 1 4 7	4: 6 4 2 7 3 1 5 8
5: 4 1 2 8 7 3 6 5	5: 8 7 1 5 6 4 3 2
6: 6 2 5 7 8 4 3 1	6: 5 4 7 6 2 8 3 1
7: 7 8 1 6 2 3 4 5	7: 1 4 5 6 2 8 3 7
8: 2 6 7 1 8 3 4 5	8: 2 5 4 3 7 8 1 6
male preference lists	**female preference lists**

Figure A.1 (Irving, Leather, and Gusfield 1987) Male and female preference lists (total orders).

Since no man is engaged in the beginning, we may begin with any man. We choose man 1, and man 1 proposes to woman 3, and now man 1 is engaged with woman 3. Then, woman 3 removes all men below man 1 on her list. In this case, she removes just man 4 from her list. We must also remove woman 3 from man 4's list. So, after one round, we obtain the lists in Figure A.2.

To continue, man 2 proposes to woman 6, and now man 2 is engaged with woman 6. Woman 6 then removes man 8, 3, and 1 from her list, and men 8, 3, and 1 remove woman 6 from their lists. Then, we may have man 6 propose to woman 6. Since woman 6 prefers man 6 to man 2, she accepts the proposal and man 2 is no longer

male lists	female lists
1: 3 1 5 7 4 2 8 6	1: 4 3 8 1 2 5 7 6
2: 6 1 3 4 8 7 5 2	2: 3 7 5 8 6 4 1 2
3: 7 4 3 6 5 1 2 8	3: 7 5 8 3 6 2 1
4: 5 8 2 6 1 4 7	4: 6 4 2 7 3 1 5 8
5: 4 1 2 8 7 3 6 5	5: 8 7 1 5 6 4 3 2
6: 6 2 5 7 8 4 3 1	6: 5 4 7 6 2 8 3 1
7: 7 8 1 6 2 3 4 5	7: 1 4 5 6 2 8 3 7
8: 2 6 7 1 8 3 4 5	8: 2 5 4 3 7 8 1 6

Figure A.2 Male and female lists after one round of the Gale-Shapley algorithm

engaged with woman 6. Instead, woman 6 is now engaged to man 6. We then remove man 2 from the list of woman 6, and consequently remove woman 6 from the list of man 2. The resulting lists are shown in Figure A.3. So far, the engaged pairs in $M \times W$ are (1,3) and (6,6).

male lists	female lists
1: 3 1 5 7 4 2 8	1: 4 3 8 1 2 5 7 6
2: 1 3 4 8 7 5 2	2: 3 7 5 8 6 4 1 2
3: 7 4 3 5 1 2 8	3: 7 5 8 3 6 2 1
4: 5 8 2 6 1 4 7	4: 6 4 2 7 3 1 5 8
5: 4 1 2 8 7 3 6 5	5: 8 7 1 5 6 4 3 2
6: 6 2 5 7 8 4 3 1	6: 5 4 7 6
7: 7 8 1 6 2 3 4 5	7: 1 4 5 6 2 8 3 7
8: 2 7 1 8 3 4 5	8: 2 5 4 3 7 8 1 6

Figure A.3 Male and female lists after three rounds of the Gale-Shapley algorithm

Next, man 3 will propose to woman 7, and since woman 7 is not engaged, man 3 becomes engaged to woman 7. Then, woman 7 removes man 7 from her list and man 7 removes woman 7 from his list. Figure A.4 shows the resulting lists.

Then, we may have man 7 propose to woman 7. Since woman 7 is engaged to a man better than man 7 (man 3), man 7's proposal is rejected. In fact, any time a man proposes to somebody no longer on his list, he will be rejected, so we may

1: 3 1 5 7 4 2 8	1: 4 3 8 1 2 5 7 6
2: 1 3 4 8 7 5 2	2: 3 7 5 8 6 4 1 2
3: 7 4 3 5 1 2 8	3: 7 5 8 3 6 2 1
4: 5 8 2 6 1 4 7	4: 6 4 2 7 3 1 5 8
5: 4 1 2 8 7 3 6 5	5: 8 7 1 5 6 4 3 2
6: 6 2 5 7 8 4 3 1	6: 5 4 7 6
7: 8 1 6 2 3 4 5	7: 1 4 5 6 2 8 3
8: 2 7 1 8 3 4 5	8: 2 5 4 3 7 8 1 6
male lists	female lists

Figure A.4 Male and female lists after four rounds of the Gale-Shapley algorithm

skip these steps when constructing shortlists. Thus, we need not refer back to the original preference lists while creating shortlists. It is enough to have a man who is not engaged just propose to the first woman on his shortened list. We have now gone over cases of all situations in the construction of shortlists. The reader may finish the algorithm and check the result with Figure A.5. Recall that the algorithm terminates when all men are engaged. The resulting stable matching is the matching that comes from pairing each man with the first woman on his shortlist (see Chapter 2).

1: 3 1 5 7 4	**1: 4 3 8 1 2**
2: 1 3 4 8 7	**2: 3 7 5 8**
3: 7 4 3 1 2 8	**3: 7 5 8 3 6 2 1**
4: 5 8 6 1 4 7	**4: 6 4 2 7 3 1 5**
5: 4 2 8 7 3 6 5	**5: 8 7 1 5 6 4**
6: 6 5 7 4 3	**6: 5 4 7 6**
7: 8 6 2 3 4 5	**7: 1 4 5 6 2 8 3**
8: 2 7 1 3 5	**8: 2 5 4 3 7**
male shortlists	**female shortlists**

Figure A.5 (Irving, Leather, and Gusfield 1987) Male and female male-optimal shortlists

Now, $\rho_2 = (3,7),(5,4),(8,2)$ is a rotation in the male-optimal shortlists. Notice that first(3) $=7$, first(5) $=4$, first(8) $=2$, second(3) $=4$, second(5) $=2$, and second(8) $=7$. To eliminate ρ_2, we may begin with woman 7. Woman 7 removes each man who follows man 8 on her shortlist. So, woman 7 removes man 3 from her shortlist and man 3 removes woman 7 from his shortlist. We perform the same process with woman 4 and woman 2 to complete the process of eliminating ρ_2.

Consider the roommate problem with preference lists given in Figure A.6. We will explain the construction of the phase 1 table for this instance of SRP.

1	7	2	6	8	5	3	4
2	4	6	5	3	8	1	7
3	5	2	1	7	4	6	8
4	1	7	3	6	5	8	2
5	7	1	8	4	6	2	3
6	7	3	8	4	5	1	2
7	2	8	4	3	5	6	1
8	4	2	3	5	6	7	1

Figure A.6 (Gusfield 1988)
Preference lists in the running example

Since the construction of the phase 1 table is analogous to the construction of shortlists for an instance of SMP, we will use the term *propose* in the obvious way. That is, when we choose a free person f in the algorithm, we say f proposes to first(f). Notice that 1 is semi-engaged to 7, because last(first(1)) = last(7) = 1. We may choose any free person, such as person 5. Person 5 proposes to 7, so person 7 removes the people below person 5 on his or her list. Specifically, 7 removes 6 and 1 from his or her list. Then, 6 and 1 must remove 7 from their lists. Figure A.7 shows the resulting lists.

Note that at this point, 1 is no longer semi-engaged to 7. Person 1 becomes free, so he or she may propose to person 2. Phase 1 of the algorithm will continue until every person is semi-engaged to the head of their list. The resulting phase 1 table is shown in Figure A.8.

Now, if $E = (1, 2, 3)$, $H = (2, 6, 5)$, and $S = (6, 5, 2)$, then $R = (E, H, S)$ is

1: 2 6 8 5 3 4
2: 4 6 5 3 8 1 7
3: 5 2 1 7 4 6 8
4: 1 7 3 6 5 8 2
5: 7 1 8 4 6 2 3
6: 3 8 4 5 1 2
7: 2 8 4 3 5
8: 4 2 3 5 6 7 1

Figure A.7 Lists after one round of phase 1

1	2	6	5	3	4		
2	6	5	3	8	1		
3	5	2	1	7	4	6	
4	1	7	3	6	5	8	
5	7	1	8	4	6	2	3
6	3	8	4	5	1	2	
7	8	4	3	5			
8	4	2	5	6	7		

Figure A.8 (Gusfield 1988) Phase 1 table in the running example

a rotation exposed in the phase 1 table. Notice that first(1) = 2, first(2) = 6, first(3) = 5, second(1) = 6, second(2) = 5, and second(3) = 2. To eliminate this rotation, we may begin with the fact that second(1) = 6, so that everyone below person 1 on person 6's list will be removed. So, person 2 is removed from person 6's list, and 6 is removed from 2's list. To continue, second(2) = 5, so everyone below 2 on 5's list is removed. That is, 3 is removed from 5's list and 5 is removed from 3's list. Lastly, second(3) = 2, so we remove 8 and 1 on 2's list (since they are below 3). We must also remove 2 from the list of 8 and the list of 1. This completes the elimination of R. The resulting table is shown in Figure A.9.

1	6	5	3	4		
2	5	3				
3	2	1	7	4	6	
4	1	7	3	6	5	8
5	7	1	8	4	6	2
6	3	8	4	5	1	
7	8	4	3	5		
8	4	5	6	7		

Figure A.9 (Gusfield 1988) The result of eliminating the rotation R from the phase 1 table