UNIVERSITY OF CALIFORNIA

Los Angeles

Distributed Almost Stable Matchings

A dissertation submitted in partial
satisfaction of the requirements for the
degree Doctor of Philosophy in Mathematics

by

William Bailey Rosenbaum

2016

ABSTRACT OF THE DISSERTATION

Distributed Almost Stable Matchings

by

William Bailey Rosenbaum

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2016

Professor Rafail Ostrovsky, Chair

The Stable Marriage Problem (SMP) is concerned with the follow scenario: suppose we have two disjoint sets of agents—for example prospective students and colleges, medical residents and hospitals, or potential romantic partners—who wish to be matched into pairs. Each agent has preferences in the form of a ranking of her potential matches. How should we match agents based on their preferences?

We say that a matching is ***stable*** if no unmatched pair of agents mutually prefer each other to their assigned partners. In their seminal work on the SMP, Gale and Shapley [11] prove that a stable matching exists for any preferences. They further describe an efficient algorithm for finding a stable matching. In this dissertation, we consider the computational complexity of the SMP in the distributed setting, and the complexity of finding "almost stable" matchings. Highlights include (1) communication lower bounds for finding stable and almost stable matchings, (2) a distributed algorithm which finds an almost stable matching in polylog time, and (3) hardness of approximation results for three dimensional analogues of the SMP.

The dissertation of William Bailey Rosenbaum is approved.

Marek Biskup

Yiannis N Moschovakis

Alexandr Sherstov

Rafail Ostrovsky, Committee Chair

University of California, Los Angeles

2016

For Alivia—my stable partner

Contents

Acknowledgments

First and foremost, I thank my advisor Rafi Ostrovsky. Thank you for pushing me to improve my results and find the narrative in my arguments. I am awed and inspired by your keen insight and intuition. Your guidance has helped make me a better mathematician.

I thank my doctoral committee members Marek Biskup, Yiannis Moschovakis, and Alexander Sherstov for their encouragement and support.

I thank my collaborators, Yannai Gonczarowski and Noam Nisan who generously agreed to join forces when I learned we had been working independently on the same problem. The material in Chapter 2 is based on our joint work [13]. I am especially grateful to Yannai for sharing his wisdom about the stable marriage problem, and LaTeX dark magic.

I have had the immense privilege of learning from some wonderful teachers throughout my life. I am especially thankful of (in roughly chronological order) Sherm Williamson, Linda Hill, Tom Wieting, Joe Roberts, Jerry Shurman, Benny Sudakov, and Alexander Sherstov. Thank you for your encouragement and your dedication to the art of teaching.

I thank Olga Radko for helping me to become a better teacher. Thank you to all of my students at UCLA—you are a perpetual reminder of why I started studying math in the first place.

Thank you to my fellow mathematicians at UCLA, especially Sam Miner for being a wonderful hiking companion, pop culture connoisseur, and all-around great guy; Zach Norwood, Sudesh Kalyanswamy, and Andy Soffer for being spectacular office mates; Pietro Carolino and Prabhanjan Ananth for sharing my excitement for TCS; and Jackie Lang for her tireless efforts to make the math department a better place.

I thank the UCLA math department staff—especially Maggie Albert, Maida Bassili, Martha Contreras, and Leticia Dominguez—for helping me navigate my path through UCLA.

I am eternally grateful to my parents, David and Margaret Rosenbaum, and my sister Liz Juusola. It has only been through your love and support that I have been able to flourish. Thank you for always putting my education first. I also thank the family I have gained since starting at UCLA: Patricia Novak, Nick Zappas, and Geri Novak. I cannot even express how helpful you have been these past few years. Thanks to Marika and Simona Zappas for being the middle and youngest sisters, respectively.

Thank you Alivia Zappas, for everything. I cannot imagine having done all of this without you. This one's for you.

Finally, I thank Finnegan Zappenbaum for reminding that sometimes the best thing is to go for a walk.

Curriculum Vitae

Education

- **University of California**, Los Angeles, CA

  M.A. in Mathematics, Spring 2011

- **Reed College**, Portland, OR

  B.A. in Mathematics, Spring 2009

  Advisor: Jerry Shurman

  Thesis: *Analysis on Circles: A Modern View of Fourier Series*

Published Work

- *Fast Distributed Almost Stable Matchings* (with Rafail Ostrovsky), ACM Symposium on Principles of Distributed Computing (PODC), 2015.

- *A Stable Marriage Requires Communication* (with Yannai Gonczarowski, Noam Nisan and Rafail Ostrovsky), ACM-SIAM Symposium on Discrete Algorithms (SODA), 2015.

- *It's Not Easy Being Three: The Approximability of Three-Dimensional Stable Matching Problems* (with Rafail Ostrovsky), International Workshop on Matching Under Preferences (MATCH-UP), 2015.

- *Analysis On Circles: A Modern View of Fourier Series.* Undergraduate Thesis, Reed College, 2009.

Awards & Funding

- **Teaching Assistant Consultantship**

  Department of Mathematics, UCLA, Fall 2015

- **Student Travel Grant**

  Association of Computing Machinery, PODC, 2015

- **Graduate Student Instructorship**

  Department of Mathematics, UCLA, 2014

- **Robert Sorgenfrey Distinguished Teaching Award**

  Department of Mathematics, UCLA, 2013

- **Phi Beta Kappa**

  Reed College, 2009

- **Commendation for Academic Achievement**

  Reed College, 2004–2005, 2006–2007, 2007–2008, 2008–2009

Chapter 1

Introduction

In this chapter we introduce the Stable Marriage Problem (SMP) which is the focus of this dissertation. The SMP was originally studied by Gale and Shapley in their seminal work, *College Admissions and the Stability of Marriage* in 1962 [11]. Since then, the SMP and its many variants—which we will refer to as ***stable matching problems***—have been the focus of hundreds of journal articles and several books which span the fields of economics, discrete mathematics, and computer science. The broad appeal of stable matching problems is succinctly described by Kurt Melhorn in his forward to [31]:

> Matching under preferences is a topic of great practical importance, deep mathematical structure, and elegant algorithmics. The great practical importance stems from the numerous applications such as the assignment of students to universities, families to housing, kidney transplant patients to donors, and so on. In these applications, the participants have preferences over the outcomes and the goal is to find an assignment that optimizes the satisfaction of the participants. The importance of the area was recently clearly demonstrated to the

world-at-large by the award of the 2012 Nobel Prize in Economics to Alvin Roth and Lloyd Shapley, partly for their work on the stable marriage problem.

While stable matching problems are well studied, there remain many fascinating open questions of both theoretical and practical significance. This dissertation provides new results concerning the computational complexity of several variants of the SMP. We are especially interested in distributed models of computation, where the input to a problem is scattered across several parties. While this model seems to be the natural context for many stable matching problems arising in nature, it has only been formally addressed in the literature quite recently.

The remainder of the chapter is organized as follows: Section 1.1 gives a formal definition of the stable marriage problem, and describes the classical results of Gale and Shapley. It further describes an important variant of the SMP with incomplete preferences. Section 1.2 gives results concerning the computational complexity of finding a stable matching. In Section 1.3 we show that the SMP has a natural interpretation as a statement about graphs. We describe a computational model which allows us to consider the SMP as a distributed problem. We further prove a computational lower bound which motivates the use of "almost stable" matchings in the distributed setting. Section 1.4 formalizes several relaxations of the stable marriage problem to "almost stable" matchings. Finally, Section 1.5 introduces three dimensional variants of the SMP. A thorough introduction to the SMP can be found in Gusfield and Irving's wonderful book [15], while a comprehensive survey of recent work (through 2013) appears in Manlove's book [31].

1.1 The Stable Marriage Problem

The stable marriage problem (SMP) was introduced by Gale and Shapley in [11]. Their original motivation was to describe a mechanism for assigning prospective students to colleges. To model the situation, each student has preferences in the form of a ranking of their most favored colleges. Similarly, the colleges rank the prospective students (perhaps by academic performance, standardized test scores, strength of recommendations, etc). Ideally, an assignment of students to colleges should have the property that no student or college has an incentive to deviate. Specifically, we wish to avoid the following scenario:

> Alice is assigned to go to USC, but she prefers UCLA to USC. Further, UCLA prefers Alice to a student Bob who is assigned to UCLA.

In such a scenario, Alice and UCLA would both be better off if UCLA expels Bob, and Alice leaves USC to enroll at UCLA. When this happens, we say that Alice and UCLA form a ***blocking*** or ***unstable*** pair, as they have a mutual incentive to depart from their assignments. An assignment without blocking pairs is known as a ***stable matching***. With these definitions, a few natural questions arise: Under what conditions (i.e., for which preferences) does a stable matching exist? If a stable matching exists, can we find one efficiently?

Gale and Shapley proved that, remarkably, stable matchings exist for every possible set of preferences for the students and the colleges. Further, a stable matching can be efficiently found by applying a simple algorithm.

While Gale and Shapley's results apply to the problem of assigning students to colleges, their seminal work focuses on the conceptually simpler (though perhaps disquietingly heteronormative) problem of forming romantic partnerships between sets of men and women. Each man ranks all of the women, and each woman ranks all of the men. They seek a one-to-one matching between the men and women where again no pair has an incentive to deviate. Concretely, if Alice and Bob are matched together and Alice prefers Carl to Bob, Carl should be matched with a woman he prefers to Alice. Otherwise, Alice and Carl have an incentive to jilt their partners and elope. The problem of finding such a stable matching is ***the stable marriage problem*** (***SMP***).

The stable marriage problem is simpler than the college assignment problem because it avoids the complication of colleges admitting multiple students. However, Gale and Shapley's solution to the SMP (described in the sequel) can easily be adapted to the more general college assignment problem.

1.1.1 The basic problem, formally

Let M and W denote disjoint sets of ***men*** and ***women***, respectively. We refer to an element in $M \cup W$ by the gender-neutral ***agent***. For simplicity, we assume that $|M| = |W| = n$. Each man $m \in M$ has ***preferences*** in the form of a total order $\prec_m$ over the set W of women. If $w, w' \in W$ and $w \prec_m w'$, we say that m ***prefers*** w to w'. Symmetrically, each woman $w \in W$ holds a ranking $\prec_w$ over the set M of men. We denote the set of preferences for all men and women by $\mathcal{P}$.

Definition 1.1. An ***instance*** of the stable marriage problem (SMP) is a triple $(M, W, \mathcal{P})$ consisting of a set M of men, a set W of women, and a set $\mathcal{P} = \{\prec_a \mid a \in M \cup W\}$ of preferences for each man and woman.

We use μ to denote a ***matching*** between the men and women. That is, μ is a set of disjoint pairs $\mu \subseteq M \times W$ such that each man and woman appears in exactly one pair. We refer to the pairs $(m, w) \in \mu$ as ***partners***. Given a matching μ, we denote m's partner in μ by $\mu(m)$.

Similarly, w's partner in μ is $\mu(w)$. Thus

$$(m,w) \in \mu \iff \mu(m) = w \iff \mu(w) = m.$$

In the case where some m does not have a partner (i.e., μ is only a partial matching) then we will write $\mu(m) = \varnothing$.

Definition 1.2 (Blocking pair, stable matching)**.** Suppose $(M, W, \mathcal{P})$ is an instance of the SMP and μ is a matching. Given a pair $(m, w) \in M \times W$ we say that (m, w) is ***blocking*** if

$$w \prec_m \mu(m) \quad \text{and} \quad m \prec_w \mu(w).$$

That is, (m, w) is blocking if m and w mutually prefer each other to their partners in μ. If μ contains no blocking pairs, we say that μ is ***stable***.

The name "blocking pair" is due to Gale and Shapley [11]. For a blocking pair (m, w), both m and w would both benefit if they abandoned their respective partners in μ and instead formed a new pair (leaving $\mu(m)$ and $\mu(w)$ partner-less).

Our primary goal is, given an SMP instance $(M, W, \mathcal{P})$ to find a matching μ which is stable. Under what conditions does a stable matching exists? Is there an efficient way of finding a stable matching? Following the work of Gale and Shapley, we give remarkably elegant solutions to both of these questions in the sequel.

1.1.2 The Gale-Shapley algorithm

In their seminal work on the SMP, Gale and Shapley prove the following remarkable result.

Theorem 1.3 (Gale & Shapley [11])**.** Let $(M, W, \mathcal{P})$ be any instance of the SMP. Then there exists a stable matching μ.

To prove the theorem, Gale and Shapley describe an efficient algorithm which finds a stable matching. Gale and Shapley's algorithm is often referred to as a ***deferred acceptance*** algorithm. The idea is that in the beginning, each man proposes to his most favored woman. If a woman receives multiple proposals, she rejects all but her most favored suitors, and defers judgment on her best proposal so far. When a man is rejected, he proposes to his next most favored woman. The algorithm continues with rounds of proposals and rejections until every woman has received at least one proposal. At this point, each man is the suitor to exactly one woman, and the women collectively accept their deferred proposals. A formal description of the Gale-Shapley deferred acceptance algorithm is given in Algorithm 1. Theorem 1.3 is proved by showing the correctness of the Gale-Shapley algorithm.

Algorithm 1 GaleShapley$(M, W, \mathcal{P})$

1: $\mu \leftarrow \varnothing$
2: each $m \in M$ sets $W_m \leftarrow W$
3: **while** there exists $m \in M$ such that $\mu(m) = \varnothing$ and $W_m \neq \varnothing$ **do**
4:     $w \leftarrow m$'s most preferred woman in W_m
5:     **if** $\mu(w) = \varnothing$ **then**
6:         $\mu \leftarrow \mu \cup \{(m, w)\}$
7:     **else if** $\mu(w) \prec_w m$ **then**
8:         $W_{\mu(w)} \leftarrow W_{\mu(w)} \setminus \{w\}$
9:         $\mu \leftarrow \mu \setminus \{(\mu(w), w)\} \cup \{(m, w)\}$
10:     **else**
11:         $W_m \leftarrow W_m \setminus \{w\}$
12:     **end if**
13: **end while**
14: **return** μ

To prove the correctness of Algorithm 1, we first prove the following lemmas.

Lemma 1.4. Suppose $w \in W$ and $\mu(w) \neq \varnothing$ after some iteration of the while loop in Algorithm 1. Then we have $\mu(w) \neq \varnothing$ in every subsequent iteration.

Proof. This is an immediate consequence of lines 7–12 in Algorithm 1. Indeed, if $\mu(w) = m$, then either $(m, w) \in \mu$ for the remainder of the algorithm, or a new pair of the form (m', w) is added to μ. □

Lemma 1.5 (Woman monotonicity)**.** Suppose $\mu(w) = m$ in some iteration of the loop in Algorithm 1. If $\mu(w) = m'$ in some later iteration we have $m' \preceq_w m$. That is, w's matches can only improve in each iteration.

Proof. Note that w can only reject m in lines 8–9, which only get executed if w prefers her new suitor m' to m. □

Proof of Theorem 1.3. We first argue that the output of Algorithm 1 is indeed a (perfect) matching. To this end, note that when the main loop terminates, every man $m \in M$ satisfies either $\mu(m) \neq \varnothing$ or $W_m = \varnothing$. We claim that for all men m, $\mu(m) \neq \varnothing$. To see this, note that if some $W_m = \varnothing$ then m was rejected by every $w \in W$. But a woman w can only reject m if she already has $\mu(w) \neq \varnothing$. Thus, by Lemma 1.4, if m is rejected by all $w \in W$, all women w must have $\mu(w) \neq \varnothing$ while $\mu(m) = \varnothing$. Since $|M| = |W| = n$, this cannot occur.

We next argue that the output μ is stable. To see this, suppose to the contrary that (m, w) is a blocking pair. In particular $w \prec_m \mu(m)$. Since m proposes to women in order of his preference, this implies that in some iteration of the loop in Algorithm 1, m must have proposed to w, and w must have rejected m in a subsequent iteration. Let m' denote w's partner immediately after she rejected m. Then by Lemma 1.5, we must have

$$\mu(w) \preceq_w m' \prec_w m.$$

This contradicts the assumption that (m, w) was blocking, hence μ does not contain any blocking pairs. □

1.1.3 Optimal stable matchings

Theorem 1.3 states that stable matchings always exist, and the Gale-Shapley algorithm gives an explicit means of finding a stable matching for any instance. In general there may be many stable matchings.

Exercise 1.6. For $n = 2$ (that is, $|M| = |W| = 2$) find preferences $\mathcal{P}$ that admit two stable matchings μ_0 and μ_1. Using the construction for $n = 2$, show that for every k, there exists an instance with $2k$ men and women which has at least 2^k stable matchings. Thus, the number of stable matchings can grow exponentially in the instance size n.

Given an instance $(M, W, \mathcal{P})$ of the SMP, we denote the set of all stable matchings by $\mathcal{M}$. For $m \in M$, $w \in W$, we say that m and w are ***stable partners*** if there exists $\mu \in \mathcal{M}$ with $(m, w) \in \mu$.

The set $\mathcal{M}$ is a partially ordered set (poset) when endowed with following order.

Definition 1.7. Suppose $\mu, \mu' \in \mathcal{M}$ and $m \in M$. Then we say that m ***(weakly) prefers*** μ to μ' and write $\mu \preceq_m \mu'$ if $\mu(m) \preceq_m \mu'(m)$. That is, m (weakly) prefers his partner in μ to μ'. We say that the men ***weakly prefer*** μ to μ' and write $\mu \preceq_M \mu'$ if for all $m \in M$, $\mu \preceq_m \mu'$. That is, *all* men weakly prefer μ to μ'.

Dually, we can define women preference on $\mathcal{M}$, thus giving the poset $(\mathcal{M}, \preceq_W)$. We remark that the posets $(\mathcal{M}, \preceq_M)$ and $(\mathcal{M}, \preceq_W)$ have the structure of a distributive lattice (see, for example, [29, 15]).

Definition 1.8. Fix an instance $(M, W, \mathcal{P})$ of the SMP. We say that $\mu_0 \in \mathcal{M}$ is ***male optimal*** if for all $\mu \in \mathcal{M}$ we have $\mu_0 \preceq_M \mu$. Similarly, $\mu_z \in \mathcal{M}$ is ***male pessimal*** if for all $\mu \in \mathcal{M}$ we have $\mu \preceq_M \mu_z$. Female optimality and pessimality are defined analogously.

Observe that that optimal and pessimal matchings must be unique.

Theorem 1.9 (Gale and Shapley [11])**.** Let $(M, W, \mathcal{P})$ be an instance of the SMP, and μ_0 the matching output by the Gale-Shapley algorithm. Then μ_0 is the male optimal stable matching.

Proof. Suppose to the contrary that there exists $\mu \in \mathcal{M}$ with $\mu \neq \mu_0$ such that $\mu_0 \not\preceq_M \mu$. That is, there exists $m \in M$ who strictly prefers his partner w in μ to his partner w' in μ_0. Since m prefers w to w', and he proposes to women in preference order in the Gale-Shapley algorithm, he must have been rejected by w during an execution of the algorithm. Suppose that when w rejected m, this was the first time in the execution that a man was rejected by a stable partner. Suppose further that w rejected m because she received a proposal from m' (whom she prefers to m). Since w's rejection of m was the first rejection of a stable partner, m' does not have any stable partners whom he prefers to w. Thus, m' must (strictly) prefer w to his partner $\mu(m')$ in μ. However, since w rejected m for m', we must also have $m' \prec_w m$. Thus, (m', w) form a blocking pair in μ, contradicting its stability. □

The following theorem gives the relationship between the partial orderings of $\mathcal{M}$ by men's preferences and women's preferences (i.e., $(\mathcal{M}, \preceq_M)$ and $(\mathcal{M}, \preceq_W)$).

Theorem 1.10 (McVitie and Wilson [32])**.** Let $(M, W, \mathcal{P})$ be an instance of the SMP and $\mu, \mu' \in \mathcal{M}$. Suppose $\mu \preceq_M \mu'$. Then $\mu' \preceq_M \mu$. In particular, if μ_0 is male optimal, then it is female pessimal, and if μ_z is female optimal, then it is male pessimal.

Using Theorems 1.9 and 1.10, we can efficiently identify instance $(M, W, \mathcal{P})$ with unique stable matchings. Specifically, we run the Gale-Shapley algorithm with the men proposing to women to find μ_0, the male optimal/female pessimal matching. Applying the algorithm with the roles of the genders reversed—that is, women propose to men—yields the female optimal/male pessimal matching μ_z. Then $(M, W, \mathcal{P})$ admits a unique stable matching if and only if $\mu_0 = \mu_z$.

Corollary 1.11. Let $(M, W, \mathcal{P})$ be an instance of the SMP. Suppose μ_z is the male pessimal matching and μ_0 the female pessimal matching. Then $\mu_0 = \mu_z$ if and only if $(M, W, \mathcal{P})$ admits a unique stable matching.

1.1.4 Incomplete preferences

Previously, we assumed that the agents' preferences $\mathcal{P}$ were ***complete*** in the sense that every man $m \in M$ ranks every woman $w \in W$, and symmetrically every woman w ranks every man m. In nature, however, this assumption seems unnatural. Indeed, when applying to colleges, one does not typically apply to *every* college, but only to a chosen subset of schools. Here we relax the notion of preferences so that they may be ***incomplete***.

Definition 1.12 (Incomplete preferences)**.** Let M and W be disjoint sets of agents. ***Incomplete preferences*** $\mathcal{P}$ consist of a subset $W_m \subseteq W$—m's ***acceptable partners***—for each $m \in M$ as well as a ranking $\prec_m$ over W_m. Similarly, each $w \in W$ holds $M_w \subseteq M$ and ranking $\prec_w$ over M_w. We assume preferences are ***symmetric*** in the sense that if $w \in W_m$, then $m \in M_w$. If $w \notin W_m$ (and symmetrically $m \notin M_w$) then we say that m and w are ***unacceptable partners***.

The notions of blocking/unstable pair and stable matching are easily generalized to incomplete preferences, where we need only consider acceptable partners to be blocking pairs. In this case, a stable matching μ may not be a perfect matching (i.e., some $m \in M$, $w \in W$ may have $\mu(m), \mu(w) = \varnothing$). Nonetheless, the results of the previous subsection easily generalize to incomplete preferences.

Theorem 1.13. Suppose M and W are disjoint sets and $\mathcal{P}$ incomplete preferences. Then there exists a stable matching μ.

The proof of Theorem 1.3 is easily modified to prove Theorem 1.13. Further, the Gale-Shapley algorithm (Algorithm 1) still finds a stable matching for incomplete preferences, and its run-time is linear in the number of acceptable partners $\sum_{m \in M} |M_m|$, rather than n^2.

Since stable matchings for incomplete preferences are not generally perfect matchings, it is natural to ask which agents are matched in stable matchings. Is it possible to have stable matchings μ and μ' such that $\mu(m) = \varnothing$, but $\mu'(m) \neq \varnothing$? Roth [42] proved that this situation never occurs in what is known as the "rural hospitals theorem."

Theorem 1.14 (Roth, rural hospitals [42])**.** Fix an instance of the SMP with preferences $\mathcal{P}$. For a stable matching μ, let M_μ (respectively W_μ) denote the set of $m \in M$ with $\mu(m) \neq \varnothing$ (respectively $w \in W$ with $\mu(w) \neq \varnothing$). Then for all stable matchings μ, μ' we have $M_\mu = M_{\mu'}$ and $W_\mu = W_{\mu'}$. That is, the sets of matched men and matched women are the same for all stable matchings with preferences $\mathcal{P}$.

1.2 Computational Complexity of the SMP

The Gale-Shapley algorithm always terminates in time $O(n^2)$. To see this, note that there are at most $n(n-1)$ proposals by the men, as each of the n men can propose at most $n-1$ times. This bound is asymptotically tight, as one can construct preferences on which the Gale-Shapley algorithm requires $n^2 - O(n)$ proposals before it terminates.

Exercise 1.15. Construct preferences $\mathcal{P}$ such that in the (unique) stable matching every all but one man is matched with one of his two least favored woman.

Interestingly, if the preferences for the men and women are chosen uniformly at random, the expected run-time of the Gale-Shapley algorithm is $\Theta(n \log n)$ [46].

In 1976, Knuth [29] asked, *"Does there exist an algorithm to find a stable matching where the number of operations increases less quickly than n^2 in the worst case?"* A related question was put forward in 1987 by Gusfield [14], who asked whether even *verifying* the stability of a proposed matching can be done any faster. As the input size is quadratic in n, these questions only make sense in models that do not require sequentially reading the whole input, but rather provide some kind of random access to the preferences of the participants.

A partial answer to both questions was given by Ng and Hirschberg [33], who considered a model that allows two types of unit-cost queries to the preferences of the participants: "what is woman w's ranking of man m?" (and, dually, "what is man m's ranking of woman w?") and "which man does woman w rank at place k?" (and, dually, "which woman does man m rank at place k?").

Theorem 1.16 (Ng & Hirschberg [33])**.** Any algorithm which finds a stable matching using only queries of the form "what is woman w's ranking of man m?" and "which man does woman w rank at place k?" requires $\Omega(n^2)$ queries to the men and women in the worst case.

Chou and Lu [6] later showed that even if one is allowed to separately query each of the $\log n$ bits of the answer to queries such as "which man does woman w rank at place k?" (and its dual query), $\Theta(n^2 \log n)$ such Boolean queries are still required in order to deterministically find a stable matching.

These results of [33] and [6] still leave two questions open. The first is whether some more powerful computational model may allow for faster algorithms. While most "natural" algorithms for the SMP do fit into the computational models described above, there may be others that do not. The second question concerns randomized algorithms: can a randomized algorithm do significantly better than deterministic ones? This question is especially fitting for this problem as the *expected* running time is known to be small.[1] In Chapter 2, we employ communication complexity to give negative answers to both questions, as well as several other related questions.

[1]In particular, this would be the case if the expected running time could be made small for *any* distribution on preferences, rather than just the uniform one.

1.3 Distributed Stable Matchings

1.3.1 SMP as a problem on graphs

The stable marriage problem can be viewed as a graph theoretic problem. Recall that a ***graph*** $G = (V, E)$ consists of a set V of ***vertices*** and $E \subseteq V \times V$ of ***edges***. Given an instance $(M, W, \mathcal{P})$ of the SMP, we associate the graph G with vertex set $V = M \cup W$ and edge set $E = \{(m, w) \mid m \in M, w \in W_m\}$ of acceptable partners. Thus G is a bipartite graph, as all edges have one endpoint in M and the other in W. Throughout this section, we will use ***agent*** and ***vertex*** interchangeably. Also, ***acceptable partner*** is synonymous with ***neighbor***.

In many contexts, this graph theoretic view of the SMP is a natural one. For example, in the case of romantic partnerships, there may be an underlying social network in which acceptable partners are acquaintances; in a computer network or content delivery system, the network topology determines the acceptable partners. In the graph model, we can view each vertex/agent as ranking their acceptable partners, or equivalently as ranking their incident edges.

In the graph theoretic view of the SMP, finding a stable matching has a natural interpretation as a distributed problem. Specifically, we assume that each vertex initially has only a local view, consisting of a ranking of its neighbors. In order to find a stable matching, the vertex must communicate with its neighbors and perform local computations. The Gale-Shapley algorithm can be interpreted as a distributed algorithm in this model. The algorithm proceeds in rounds where alternatingly the men and women send messages. In the first round, each $m \in M$ sends the message PROPOSE to its most favored $w \in W_m$. In the following round, each $w \in W$ receiving proposals responds by sending the message DEFER to her most favored suitor, and REJECT to the others. Rejected men PROPOSE to their next most favored w, and the process continues until no more proposals are made. In this view of the Gale-Shapley algorithm, each vertex communicates only with its neighbors. Thus, the vertices are able to find a stable matching by performing local computations based on the simple messages received from their neighbors.

1.3.2 Computational model

Here we formalize the computational model in which we will describe the distributed algorithms for finding stable matchings. We use the CONGEST model formalized by Peleg [39]. In this distributed computational model, each vertex $v \in M \cup W$ represents a processor. Given preferences $\mathcal{P}$, the communication links between the players are given by the set of edges E in the ***communication graph*** G. Communication is performed in synchronous rounds. Each communication round occurs in three stages: first, each processor receives messages (if any) sent from its neighbors in G during the previous round. Next, each processor performs local calculations based on its internal state and any received messages. We make no restrictions on the complexity of local computations. Finally, each processor sends short ($O(\log n)$ bit) messages to its neighbors in G—the processor may send distinct messages to distinct neighbors. In the CONGEST model, complexity is measured by the number of communication rounds needed to solve a problem.

When computation begins, each vertex v has initial ***state***, which in the context of the SMP will consist of v's neighbors/acceptable partners, and v's ranking of its neighbors. When computation terminates, each vertex announces an ***output***. For the SMP, that output will consist that vertex's (stable) partner. In this context, we refer to the problem of finding a stable matching as the ***distributed stable marriage problem*** or ***DSMP***.

1.3.3 "Easy" lower bounds for distributed stable matchings

For a graph $G = (V, E)$, and $v, w \in V$, we use $\mathrm{dist}_G(v, w)$ to denote the graph distance—i.e., the length of the shortest path—between v and w. In the CONGEST model, computation is restricted by locality and communication. Locality refers the topology of the communication network G: vertices only communicate with their neighbors in the graph. If a distributed algorithm terminates after t rounds, then the output of each vertex v is a function only of v's initial state, and the initial states of w satisfying $\mathrm{dist}_G(v, w) \leq t$. Thus, if the output of a vertex v depends on the input of a vertex w which is far away from v, then many communication rounds will be required in order for v to produce the correct output. The following lemma is a consequence of the description of the CONGEST model of computation.

Lemma 1.17. Let $G = (V, E)$ be a communication graph and $v, w \in V$. Suppose that for some computational problem P, the correct output of v depends on the initial state of w, and that some (correct) algorithm for P in the CONGEST model halts in t rounds. Then $t \geq \mathrm{dist}_G(v, w)$.

Using Lemma 1.17, we can prove general lower bounds for solving the SMP in the distributed model.

Theorem 1.18. Any distributed algorithm which solves the distributed stable marriage problem with incomplete preferences requires $\Omega(n)$ communication rounds in the worst case.

Proof. Consider an instance of the SMP with

$$M = \{m_1, m_2, \ldots, m_n\} \quad \text{and} \quad W = \{w_1, w_2, \ldots, w_n\}.$$

Each vertex has at most two neighbors, and the communication graph is a path. Specifically, each m_i has acceptable partners w_{i-1} and w_i, while each w_j has acceptable partners m_j and

m_{j+1} (where m_1 and w_n each have a unique acceptable partner). Each m_i prefers w_{i-1} to w_i, while w_j with $j \neq 1$ prefers m_j to m_{j+1}. The vertex w_1 may prefer m_1 or m_2.

We note that if w_1 prefers m_1, then the unique stable matching for this instance is

$$\mu_1 = \{(m_1, w_1), (m_2, w_2), \ldots, (m_n, w_n)\}.$$

On the other hand, if w_1 prefers m_2, then the unique stable matching is

$$\mu_2 = \{(m_2, w_1), (m_3, w_2), \ldots, (m_n, w_{n-1})\}.$$

In particular, for any algorithm which solves the DSMP, the output of w_n depends on the input of w_1. Since $\mathrm{dist}_G(w_1, w_n) = 2n - 2$, the desired result follows from Lemma 1.17. □

Remark 1.19. The lower bound in Theorem 1.18 shows that we cannot hope to solve the DSMP in time smaller than the diameter of the communication graph G. However, even when the diameter of G is small ($O(1)$), there is no known general purpose distributed algorithm which improves on the trivial upper bound of $O(n)$ rounds, where each vertex simply announces its preferences to all of its neighbors, and each vertex locally performs the Gale-Shapley algorithm.

Open Question 1.20. In the CONGEST model for complete preferences, is it possible to find a stable matching in $o(n)$ rounds? Note that for complete preferences, the communication graph G is the complete bipartite graph $K_{n,n}$, which has diameter 2.

Remark 1.21. We remark that there are no known $\omega(1)$ round lower bounds for any problem in the CONGEST model on $K_{n,n}$. The work of Drucker, Kuhn, and Oshman [8] shows that such lower bounds are likely difficult to obtain, as they would imply previously unknown circuit lower bounds. Thus, there is a tremendous gap between the best known (trivial) constant round lower bound, and the $O(n)$ upper bound.

1.4 Almost Stable Matchings

Given the strong distributed lower bound of Theorem 1.18 and the (centralized) run-time lower bound of, for example, Ng and Hirschberg [33] (Theorem 1.16), it is natural to ask if relaxations of the SMP may admit faster algorithms. Is it possible to find "almost stable" matchings significantly faster than (exactly) stable matchings? This question has recently been addressed in the literature (for example [1, 9, 10, 17, 26]), though there is no consensus on the proper definition of "almost stable." In this section, we survey several notions of almost stability and discuss some results related to almost stable matchings.

1.4.1 Divorce distance to stability

One reasonable way of defining almost stability is that an almost stable matching should have many pairs in common with some (exactly) stable matching. We formalize this notion in terms of divorce distance.

Definition 1.22 (divorce distance)**.** Suppose M and W are disjoint sets with $|M| = |W| = n$, and μ, μ' matchings between M and W. The ***divorce distance*** between μ and μ', denoted $d(\mu, \mu')$, is defined by

$$d(\mu, \mu') = n - |\mu \cap \mu'| .$$

We note that $|\mu \cap \mu'|$ is the number of pairs μ and μ' have in common. The name divorce distance is apropos, as it is the minimum number of divorces needed to transform μ to μ' (or vice versa). We remark that d is a metric on the set S of all perfect matchings between M and W.

Definition 1.23 (divorce distance to stability)**.** Let $(M, W, \mathcal{P})$ be an instance of the SMP, $\mathcal{M}$ the set of stable matchings, and μ' a (not necessarily stable) matching. Then the ***(divorce)***

distance to stability of μ', denoted $d(\mu')$, is defined by

$$d(\mu') = \min_{\mu \in \mathcal{M}} d(\mu', \mu).$$

That is, $d(\mu')$ is the minimum distance of μ' to any stable matching μ.

Definition 1.23 is a generalization of a notion formalized by Ünver [45], who considers only SMP instances which have a unique stable matching. The definition allows us to parametrize a relaxation of the SMP. Namely, we can form a relaxation of the SMP, where we are only required to find a matching which is ε-close to stable.

Definition 1.24. Let $(M, W, \mathcal{P})$ be an instance of the SMP with complete preferences and $|M| = |W| = n$. Let μ be a (not necessarily stable) matching, and $\varepsilon \geq 0$. We say that μ is ε-***close to stable*** if $d(\mu) \leq \varepsilon n$.

In Chapter 2 we analyze this relaxation of the SMP, and show that this variant does not afford any (asymptotic) algorithmic speed-up for any $\varepsilon < 1/2$. While the definition of $d(\mu)$ is a reasonable one, it is not immediately obvious that it is possible to compute it efficiently. While $d(\mu, \mu')$ can easily be computed for any fixed μ and μ', there may be exponentially many (in n) stable matchings $\mu \in \mathcal{M}$. Thus, computing $d(\mu)$ naively may take exponential time. Remarkably, it is possible to compute $d(\mu)$ in polynomial time. We give a proof of this fact in Appendix A. The proof relies heavily on the structure of the set of stable matchings, and reduces computing $d(\mu)$ to a max-flow/min-cut problem.

1.4.2 $(1 - \varepsilon)$-stability

Another reasonable notion of almost stability is that an almost stable matching should have few blocking pairs. We use a definition of almost stability given by Eriksson and Häggström [9], modified to allow for incomplete preference lists.

Definition 1.25. Given an instance $(M, W, \mathcal{P})$ of the SMP (with incomplete preferences) and $\varepsilon \geq 0$, we say that a matching μ is $(1-\varepsilon)$-***stable*** with respect to $\mathcal{P}$ if μ induces at most $\varepsilon |E|$ blocking pairs with respect to $\mathcal{P}$.

Note that a 1-stable matching corresponds precisely to the classical stable matching definition. In Chapter 3, we describe a distributed algorithm which for all fixed $\varepsilon > 0$ finds a $(1-\varepsilon)$-stable matching in poly-logarithmic time. Thus, $(1-\varepsilon)$-stable matchings can be found exponentially faster than their exactly stable counterparts (cf. Theorem 1.18).

For the case of bounded degree graphs (i.e., each vertex has a bounded number of acceptable partners), Floréen, Kaski, Polishchuk, and Suomela [10] showed that terminating the Gale-Shapley algorithm after a constant number of rounds results in an $(1-\varepsilon)$-stable matching. We note that the authors of [10] use a strictly stronger notion of almost stability in their paper, comparing the number of blocking pairs to $|\mu|$, the size of the matching, rather than $|E|$, the number of acceptable pairs. However, in the context of [10] where the communication graph has bounded degree, the two notions of almost-stability agree up to a constant factor. The algorithm we describe in Chapter 3 can be viewed as a generalization of the Gale-Shapley algorithm, and thus, the main result of that chapter generalizes the result of [10].

1.4.3 ε-blocking pairs

Here we give one final definition of almost stability, due to Kipnis and Patt-Shamir [26]. The idea is that we allow blocking pairs (m, w) to be present in a matching μ, so long as m and w do not prefer each other "too much" more than their assigned partners in μ. To formalize this notion, for an instance $(M, W, \mathcal{P})$ and $m \in M$, $w \in W$ acceptable partners, we denote m's rank of w by $\mathcal{P}_m(w)$. That is, $\mathcal{P}_m(w) = 1$ means that w is m's most preferred partner, while

$\mathcal{P}_m(w) = 2$ means w is m's second favorite, and so on. $\mathcal{P}_w(m)$ is defined symmetrically. For each $m \in M$, we define $\deg m$ to be the degree of m in the communication graph—that is, $\deg m$ is the number of m's acceptable partners, $\deg m = |W_m|$.

Definition 1.26 (Kipnis and Patt-Shamir [26])**.** Given $\varepsilon \geq 0$, preferences $\mathcal{P}$, and a matching μ we call an edge $(m, w) \in E$ ***ε-blocking*** if m and w appear an ε-fraction higher on each other's preferences than their assigned partners. Specifically, (m, w) is ε-blocking if

$$\mathcal{P}_m(\mu(m)) - \mathcal{P}_m(w) \geq \varepsilon \deg m$$

and

$$\mathcal{P}_w(\mu(w)) - \mathcal{P}_w(m) \geq \varepsilon \deg w.$$

We say that μ is ***ε-blocking stable*** if it contains no ε-blocking pairs.

We remark that ε-blocking stability is strictly stronger than $(1-\varepsilon)$ stability. If a matching μ is ε-blocking stable, then it is $(1 - 2\varepsilon)$-stable. Conversely, one can construct families of instances of the SMP and matchings μ which are $(1-\varepsilon)$-stable, but which are not ε'-blocking stable for any $\varepsilon' < 1$.

Kipnis and Patt-Shamir [26] proved an $\Omega(\sqrt{n}/\log n)$ round lower bound for finding an ε-blocking stable matching in the distributed setting. In Chapter 3 we describe a poly-logarithmic round algorithm which finds a $(1 - \varepsilon)$-stable matching (Definition 1.25). We feel that this upper bound bolsters the use of Definition 1.25 for almost stability, at least for practical applications. Further, the algorithm we describe in Chapter 3 produces a matching which is nearly ε-blocking stable in the sense that after the removal of an arbitrarily small fraction of "bad" $m \in M$, the resulting matching is ε-blocking stable with respect to the remaining vertices. This dichotomy of Kipnis and Patt-Shamir's polynomial lower bounds for ε-blocking stability [26] versus our poly-logarithmic upper bound for $(1-\varepsilon)$-stability in

Chapter 3 shows that the complexity of finding an almost stable matching in the distributed setting is highly sensitive to the notion of approximation used.

1.5 Three Dimensional Stable Matching Problems

In [29], Knuth asked, *"Can the stable matching problem be generalized to three sets of objects (for example, men, women, and dogs)?"*. A three "gender" variant of the SMP was studied by Alkan in [2]. Specifically, in addition to the disjoint sets M and W, there is a third set D of dogs. The goal is to partition $M \cup W \cup D$ into triples (m, w, d) consisting of exactly one man, one woman, and one dog. We call such a triple a ***family***. For simplicity, we assume that $|M| = |W| = |D| = n$.

In this three gender variant of the SMP, which we refer to as 3GSM, each man ranks pairs $(w, d) \in W \times D$. That is, each $m \in M$ holds a total order $\prec_m$ over $W \times D$. Analogously, each $w \in W$ and $d \in D$ hold rankings $\prec_w$ and $\prec_d$ over $M \times D$ and $M \times W$ respectively. An ***instance*** of 3GSM consists of a quadruple $(M, W, D, \mathcal{P})$ where M, W, and D are disjoint sets, and $\mathcal{P} = \{\prec_v \mid v \in M \cup W \cup D\}$ are preferences as described above. A ***matching*** μ is a partition of $M \cup W \cup D$ into disjoint families. Given a matching μ, we equivalently write

$$(m, w, d) \in \mu \iff \mu(m) = (w, d) \iff \mu(m, w) = d \iff \cdots.$$

Again in a (three gender) matching μ, we assume that $\mu(m), \mu(w), \mu(d) \neq \varnothing$ for all $m \in M$, $w \in W$, and $d \in D$. If μ is a disjoint set of triples which may have $\mu(m) = \varnothing$, we call μ a ***sub-matching***.

Definition 1.27. Let $(M, W, D, \mathcal{P})$ be an instance of 3GSM, and μ a matching. Suppose $(m, w, d) \in M \times W \times D$. We say that (m, w, d) is an ***unstable triple*** if

$$(w, d) \prec_m \mu(m), \quad (m, d) \prec_w \mu(w), \quad \text{and} \quad (m, w) \prec_d \mu(d).$$

A matching μ which induces no unstable triples is a ***stable matching***.

Definition 1.27 is a natural extension of Definition 1.2 to three gender matchings. While two-gender stable matchings always exist, Alkan showed that instances of 3GSM need not admit stable matchings.

Theorem 1.28 (Alkan [2]). There exists an instance $(M, W, D, \mathcal{P})$ of 3GSM which does not admit a stable matching.

Proof. We will construct an instance with $n = 2$. Denote $M = \{m_1, m_2\}$, $W = \{w_1, w_2\}$, and $D = \{d_1, d_2\}$. Consider preference lists $\mathcal{P}$ as described in the following table, where most preferred partners are listed first.

player	preferences		
m_1	(w_1, d_1)	(w_2, d_2)	$\cdots$
m_2	(w_2, d_1)	$\cdots$	
w_1	(m_1, d_1)	$\cdots$	
w_2	(m_1, d_2)	(m_2, d_1)	$\cdots$
d_1	(m_2, w_2)	(m_1, w_1)	$\cdots$
d_2	(m_1, w_2)	$\cdots$	

The ellipses indicate that the remaining preferences are otherwise arbitrary. Suppose μ is a stable matching for $\mathcal{P}$. We must have either $(m_1, w_1, d_1) \in \mu$ or $(m_1, w_2, d_2) \in \mu$, for otherwise the triple (m_1, w_2, d_2) is unstable. However, if $(m_1, w_1, d_1) \in \mu$, then (m_2, w_2, d_1) is unstable. On the other hand, if $(m_1, w_2, d_2) \in \mu$ then (m_1, w_1, d_1) is unstable. Therefore, no such stable μ exists. In particular, every matching μ contains at least one unstable triple. $\square$

Given the result of Theorem 1.28, we would like to understand which instances of 3GSM admit stable matchings, and which do not. Ng and Hirschberg examined the problem of deciding if an instance of 3GSM admits a stable matching in [34]. In particular, they proved the following result.

Theorem 1.29 (Ng & Hirschberg [34])**.** Let $I = (M, W, D, \mathcal{P})$ be an instance of 3GSM. Then the problem of deciding whether or not I admits a stable matching is NP-complete.

By Theorem 1.29, we do not expect there to be an efficient algorithm which determines if an instance of 3GSM admits a stable matching. Chapter 4 analyzes relaxations of 3GSM to finding matchings which are only "almost stable." We consider two variants of this problem, and show that both are still NP-complete to approximate within a fixed constant factor. Nonetheless, we show that it is possible to compute weaker approximations in polynomial time.

Ng and Hirschberg [34] also introduce a gender-less version of 3GSM, the 3-***person stable assignment*** problem or 3***PSA***. In this variant, there is a single set U of agents, and each agent $u \in U$ ranks all (unordered) pairs $\{u_1, u_2\} \subseteq U \setminus \{u\}$. A matching μ is a partition of U into triples. Unstable triple and stable matching are defined analogously to Definition 1.27. The proof of Theorem 1.29 easily extends to 3PSA. In Chapter 4, also consider approximate versions of 3PSA. In our explorations of 3GSM and 3PSA in Chapter 4, we show that we should not expect approximate versions of 3GSM and 3PSA to admit faster algorithms than their exact counterparts for sufficiently good approximation.

Chapter 2

Communication Complexity

In the distributed setting for the SMP (see Section 1.3), each agent is assumed to privately hold that agent's preferences. The agents then compute a stable matching by communicating with one another. A fundamental measure of the complexity of solving such a distributed problem is its ***communication complexity***, which measures how many bits the agents/processors must exchange in order to solve the problem. Communication complexity was first formalized in 1979 by Yao [48]. Since then, communication complexity has become central to the study of general complexity theory, not just for naturally distributed problems. Indeed, communication complexity has deep connections with circuit complexity, data structures, steaming algorithms, space-time tradeoffs, and VLSI circuit design. We refer the interested reader to Kushilevitz and Nisan's excellent introduction [30].

In this chapter, we analyze the communication complexity of the SMP.[1] While an instance of the SMP may have many agents, we consider a simpler two-party model of computation (that originally studied by Yao [48]). Specifically, we assume that one party, which

[1]Much of the material in this chapter originally appeared as joint work with Gonczarowski and Nisan in [13].

will we call Alice, knows all of the women's preferences, while the other party, Bob, knows all of the men's preferences. In this context, we can view the Gale-Shapley algorithm as a ***communication protocol*** between Alice and Bob. When a man m proposes to a woman w, Bob sends Alice the message "m proposes to w," and Alice responds with REJECT or DEFER depending on whether the proposal is rejected by w or deferred. Note that each proposal message and response can be encoded in $O(\log n)$ bits. Since the Gale-Shapley algorithm terminates after $O(n^2)$ proposals, the total ***communication cost*** of the Gale-Shapley algorithm is $O(n^2 \log n)$.

In general, a communication protocol between Alice and Bob could involve arbitrary Boolean queries about the men's and women's preferences—not just queries of the form "does w reject m's proposal?" The result of Ng and Hirschberg [33] (Theorem 1.16) shows that $\Omega(n^2)$ (deterministic) Boolean queries to the men's and women's preferences are necessary, if the questions are of the form "what is w's ranking of m?" and "who does m rank in place k?"

As a consequence of our analysis of the communication complexity of the SMP in this chapter, we generalize Ng and Hirschberg's results in three ways. First, we prove a general $\Omega(n^2)$ query lower bound for *any* Boolean queries to the men and (separately) the women which result in a stable matching. Secondly, our results apply even if the protocol is allowed to be randomized. We remark that in general, there can be an exponential gap between randomized and deterministic communication complexity (see, for example, [30]). Finally, we generalize these lower bounds to several problems related to the original SMP.

Theorem 2.1 (Informal, see Theorem 2.14)**.** Any randomized (or deterministic) algorithm that uses any type of Boolean queries to the women's and to the men's preferences to solve any of the following problems requires $\Omega(n^2)$ queries in the worst case:

(a) finding stable matching, or a matching whose distance to stability is at most εn (see Definition 1.23),

(b) determining whether a given matching is stable or far from stable,

(c) determining whether a given pair is contained in some/every stable matching,

(d) finding any εn pairs that appear in some/every stable matching.

These lower bounds hold even if we allow arbitrary preprocessing of all the men's preferences and of all the women's preferences separately. The lower bound for Part (a) holds regardless of which (stable or approximately stable) matching is produced by the algorithm.

Our proof of Theorem 2.1 comes from a reduction to the well-known lower bounds for the disjointness problem [24, 41] in Yao's [48] model of two-party communication complexity. We consider a scenario in which Alice holds the preferences of the n women and Bob holds the preferences of the n men, and show that each of the problems from Theorem 2.1 requires the exchange of $\Omega(n^2)$ bits of communication between Alice and Bob. Since we only measure the communication between Alice (i.e., the women) and Bob (i.e., the men), we are essentially allowing the women to communicate with each other for free, and similarly for the men. Thus our communication lower bound still applies even if the women and men are allowed to preprocess their inputs arbitrarily.

We note that Segal [44] shows by a general argument that any deterministic or nondeterministic communication protocol among all $2n$ agents for finding a stable matching requires $\Omega(n^2)$ bits of communication. Our argument for Theorem 2.1(a), in addition to being significantly simpler, generalizes Segal's result to account for randomized algorithms,[2] and even when considering only two-party communication between Alice and Bob. Furthermore,

[2] We remark that in general, there may be an exponential gap between deterministic, nondeterministic, and randomized communication complexity.

our lower bound holds even for merely determining whether a given matching is stable or far from stable (Theorem 2.1(b)), as well as for the additional related problems described in Theorem 2.1(c,d). These results immediately imply the same lower bounds for any type of Boolean queries in the original computation model, as Boolean queries can be simulated by a communication protocol.

As indicated above, Theorem 2.1(a), as well as the corresponding lower bound on the two-party communication complexity, holds not only for stable matchings but also for matchings μ with $d(\mu) \leq \varepsilon n$ (see Definition 1.23). In the context of communication complexity, Chou and Lu [6] also study such a relaxation of the stable matching problem in a restricted computational model in which communication is non-interactive (a sketching model). Chou and Lu show that any (deterministic, non-interactive, $2n$-party) protocol that finds a matching where only a constant fraction of agents are involved in blocking pairs requires $\Theta(n^2 \log n)$ bits of communication. We note that Chou and Lu's notion of almost stability is strictly stronger than $(1-\varepsilon)$-stability (Definition 1.25). Our communication complexity lower bounds are not directly comparable to Chou and Lu's results, as the two notions of almost stability are not comparable. We use a significantly more general computation model (randomized, interactive, two-party), but give a slightly weaker lower bound.

Our lower bound for verification complexity (given in Theorem 2.1(b)) is tight. Indeed there exists a simple deterministic algorithm for verifying the stability of a proposed matching, which requires $O(n^2)$ queries even in the weak comparison model that allows only for queries of the form "does woman w prefer man m_1 over man m_2?" and, dually, "does man m prefer woman w_1 over woman w_2?".[3] We do not know whether the lower bound is tight also

[3] By simple batching, this verification algorithm can be converted into one that uses only $O(\frac{n^2}{\log n})$ queries, each of which returns an answer of length $\log n$ bits (with each query still regarding the preferences of only a single agent). This highlights the fact that the lower bounds of [33] crucially depend on the exact type of queries allowed in their model.

for finding a stable matching (Theorem 2.1(a)). Gale and Shapley's algorithm uses $O(n^2)$ queries in the worst case, but $O(n^2)$ of these queries require each an answer of length $\log n$ bits, and thus the algorithm requires a total of $O(n^2 \log n)$ Boolean queries, or bits of communication. We do not know whether $O(n^2)$ Boolean queries suffice for any algorithm. While the gap between Gale and Shapley's algorithm and our lower bound is small, we believe that it is interesting, as the number of queries performed by the algorithm is exactly linear in the input encoding length. Even a slightly sub-linear algorithm would therefore be interesting. We indeed do not know of any $o(n^2 \log n)$ algorithm, even randomized and even in the strong two-party communication model, nor do we have any improved $\omega(n^2)$ lower bound, even for deterministic algorithms and even in the simple comparison model.[4]

Open Question 2.2. Consider the comparison model for the SMP that only allows for queries of the form "does man m prefer woman w_1 over woman w_2?" and, dually, "does woman w prefer man m_1 over man m_2?" How many such queries are required, in the worst case, to find a stable matching?

Open Question 2.3. What is the communication complexity of finding a $(1-\varepsilon)$-stable matching (Definition 1.25)? We note that for randomized protocols, an analogous result to Theorem 2.1 (b) does not hold for $(1-\varepsilon)$-stability. Indeed, if a matching μ induces at least $\varepsilon\,|E|$ blocking pairs, randomly sampling an edge will yield a blocking pair with at least ε probability. Thus by sampling a constant number of edges (independent of n), a randomized protocol can distinguish between stable matchings and matchings with $\varepsilon\,|E|$ blocking pairs with probability, say, $2/3$.

[4]It is interesting to note that Bei *et al.* [5] identify a similar gap for the stable marriage problem in a dramatically different computation model of trial and error.

2.1 Communication Complexity Background

We work in Yao's [48] model of two-party communication complexity (see [30] for a survey). Consider a scenario where two agents, Alice and Bob, hold values $\mathbf{x}$ and $\mathbf{y}$, respectively, and wish to collaborate in performing some computation that depends on both $\mathbf{x}$ and $\mathbf{y}$. Such a computation typically requires the exchange of some data between Alice and Bob. The ***communication cost*** of a given protocol (i.e. distributed algorithm) for such a computation is the number of bits that Alice and Bob exchange under this protocol in the worst case (i.e. for the worst $(\mathbf{x}, \mathbf{y})$); the ***communication complexity*** of the computation that Alice and Bob wish to perform is the lowest communication cost of any protocol for this computation. Generalizing, we also consider ***randomized*** communication complexity, defined analogously using randomized protocols that for every given fixed input, produce a correct output with probability at least $\frac{2}{3}$.[5]

Of particular interest to us is the disjointness function, disj. Let $n \in \mathbf{N}$ and let Alice and Bob hold subsets $A, B \subseteq \{1, 2, \ldots, n\}$, respectively. The value of the disjointness function is 1 if $A \cap B = \varnothing$, and 0 otherwise. We can also consider disj as a Boolean function by identifying A and B with their respective characteristic vectors $\mathbf{x} = (x_i)_{i=1}^n$ and $\mathbf{y} = (y_i)_{i=1}^n$, defined by $x_i = 1 \iff i \in A$ and $y_j = 1 \iff j \in B$. Thus, we can express disj using the Boolean formula $\mathrm{disj}(\mathbf{x}, \mathbf{y}) = \neg \bigvee_{i=1}^n (x_i \wedge y_i)$. By abuse of notation, we will use $\mathbf{x}$ and $\mathbf{y}$ to refer to subsets of $\{1, 2, \ldots, n\}$ and the characteristic vectors of these subsets. All of our results heavily rely on the following result of Kalyanasundaram and Schintger [24] (see also Razborov [41]):

Theorem 2.4 (Communication Complexity of disj [24, 41])**.** Let $n \in \mathbf{N}$. The randomized

[5] The results of this paper hold verbatim even if the constant $\frac{2}{3}$ is replaced with any other fixed probability p with $\frac{1}{2} < p \leq 1$.

(and deterministic) communication complexity of calculating $\operatorname{disj}(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \{0,1\}^n$ is held by Alice and $\mathbf{y} \in \{0,1\}^n$ is held by Bob, is $\Theta(n)$. Further, this lower bound holds even for unique disjointness, i.e. if we require that the inputs $\mathbf{x}$ and $\mathbf{y}$ are either disjoint or ***uniquely intersecting***: $|\mathbf{x} \cap \mathbf{y}| \leq 1$.

Our results regarding lower bounds on communication complexities all follow from defining suitable ***embeddings*** of disj into various problems regarding stable matchings, i.e. mapping $\mathbf{x}$ and $\mathbf{y}$ into suitable instances of the SMP such that finding a stable matching (or solving any of the other problems from Theorem 2.1) reveals the value of disj. Specifically, we will define maps $\mathbf{x} \mapsto \mathcal{P}_W$ and $\mathbf{y} \mapsto \mathcal{P}_M$ where $\mathcal{P}_W$ and $\mathcal{P}_M$ are the women's and men's preferences, respectively. Thus, if Alice is given $\mathbf{x}$ and Bob $\mathbf{y}$, then Alice (and respectively Bob) can construct the women's (respectively men's) preferences independently (without communication) to specify the desired SMP instance. Our proofs presented in Section 2.4 assume that the input to disj satisfies $|\mathbf{x} \cap \mathbf{y}| \leq 1$.

2.2 Summary of Communication Results

All of our results in this chapter provide lower bounds for various computations regarding the stable matching problem. For the duration of this chapter, let $(M, W, \mathcal{P})$ be an instance of with complete preferences, where $\mathcal{P}_W$ is held by Alice and $\mathcal{P}_M$ is held by Bob.

Theorem 2.5 (Communication Complexity of Finding an Almost-Stable Matching)**.** Let $0 \leq \varepsilon < \frac{1}{2}$. The randomized (and deterministic) communication complexity of finding a matching μ whose distance from stability is at most εn with respect to $(M, W, \mathcal{P})$ is $\Omega(n^2)$.

Corollary 2.6 (Communication Complexity of Finding an Exactly Stable Matching)**.** The randomized communication complexity of finding an (exactly) stable matching in $(M, W, \mathcal{P})$

is $\Omega(n^2)$.

Theorem 2.7 (Communication Complexity of Determining the Stability of a Matching)**.** Let $0 \leq \varepsilon < 1$ and let μ be a fixed matching between W and M that is promised to be either stable ($d(\mu) = 0$) or ε-far from stable ($d(\mu) \geq \varepsilon n$) with respect to $\mathcal{P}_W$ and $\mathcal{P}_M$. The randomized communication complexity of determining whether μ is stable or ε-far from stable is $\Omega(n^2)$.

Corollary 2.8 (Communication Complexity of Verifying a Stable Matching)**.** Let μ be a fixed matching between W and M. The randomized communication complexity of determining whether or not μ is stable with respect to $\mathcal{P}_W$ and $\mathcal{P}_M$ is $\Omega(n^2)$.

Remark 2.9. The lower bound given in Corollary 2.8 is tight. Indeed, exhaustively checking all pairs of agents to naïvely check for the existence of a blocking pair requires $\Theta(n^2)$ bits of communication in the worst case.

Remark 2.10. Both Theorem 2.7 and Corollary 2.8 are phrased so that the matching μ is known by both Alice and Bob before the protocol commences. Nonetheless, these results still hold if only one of them knows μ, as the straightforward way of encoding a matching between W and M requires $O(n \log n)$ bits.

Although Corollaries 2.6 and 2.8 are immediate consequences of Theorems 2.5 and 2.7, respectively, we give direct proofs of these important special cases in Section 2.3 as a warm-up to the more general construction given in Section 2.4.

Theorem 2.11 (Communication Complexity of Verifying Marital Status)**.** Let $(w, m) \in W \times M$ be fixed. The randomized communication complexity of determining whether or not (w, m) is contained in some/every stable matching (with respect to $(M, W, \mathcal{P})$) is $\Omega(n^2)$.

Remark 2.12. Gusfield [14] gives a deterministic algorithm for enumerating all pairs that belong to at least one stable matching in $O(n^2 \log n)$ Boolean queries; this yields a $O(n^2 \log n)$ upper bound for the problems described in Theorem 2.11. The question of a tight bound remains open.

Theorem 2.13 (Communication Complexity of Finding Stable Couples)**.** Let $0 < \varepsilon \leq 1$. The randomized communication complexity of finding εn pairs (w, m) that are contained in some/every stable matching (with respect to $(M, W, \mathcal{P})$) is $\Omega(n^2)$.

Theorem 2.14 (Query Complexity)**.** Any randomized (or deterministic) algorithm that uses any type of Boolean queries to the women's and (separately) to the men's preferences to solve any of the following problems requires $\Omega(n^2)$ queries in the worst case:

(a) finding a matching μ with $d(\mu) \leq \varepsilon n$, for fixed ε with $0 \leq \varepsilon < \frac{1}{2}$,

(b) determining whether a given matching μ is stable or if $d(\mu) \geq \varepsilon n$, for fixed ε with $0 \leq \varepsilon < 1$,

(c) determining whether a given pair is contained in some/every stable matching, and

(d) finding any εn pairs that appear in some/every stable matching, for fixed ε with $0 < \varepsilon \leq 1$.

The proofs of Theorems 2.5, 2.7, 2.11 and 2.14 are given in Section 2.4.2. The proofs all follow from the embedding of disjointness into a matching market that is described in Section 2.4.1.

2.3 Lower Bounds for Exactly Stable Matchings

The goal of this section is to prove Corollaries 2.6 and 2.8 directly. We remark that up to a logarithmic factor, the lower bound of Corollary 2.6 is tight, while Corollary 2.8 is tight.

The agents of each gender hold a total of $2n$ sets of preferences, where each preference list is a permutation of n. Thus there are $(n!)^{2n}$ distinct stable marriage instances of size n. Therefore, specifying a single instance requires

$$\log((n!)^{2n}) = \Theta(n^2 \log n)$$

bits in any reasonable encoding. The trivial communication protocol of, say, the men sending all of their preferences then uses $O(n^2 \log n)$, which up the the factor of $\log n$ matches our lower bound in Corollary 2.6. The bound of Corollary 2.8 is tight by the following protocol: for each pair $(m_i, w_j) \in M \times W$, Alice announces if w_j prefers m_i to $\mu(w_j)$, and symmetrically Bob announces if m_i prefers w_j to $\mu(m_i)$. Since $|M \times W| = n^2$, this communication protocol only requires $O(n^2)$ bits of communication. As part of the specification of the protocol, Alice and Bob agree on some ordering of $M \times W$, say lexicographical ordering. Thus the pair (m_i, w_j) is specified by the communication round, and thus Alice and Bob do not need to explicitly communicate i and j.

2.3.1 The special case $n = 2$

As mentioned before, we will prove Corollary 2.6 by embedding any instance of disj into an SMP instance such that the structure of a stable matching reveals the value of disj. Before describing the embedding in full generality, we look at the special case of disj where Alice and Bob each hold a single bit. Thus $\text{disj}(x, y) = 1$ if and only if at least one of the bits is 0. In this case, we will have $n = 2$ men and women. We use Alice's bit to determine the women's preferences, and Bob's bit to determine the men's preferences. Here is our embedding for the women's preferences:

player	preferences if $x = 1$	preferences if $x = 0$
w_1	$1, 2$	$2, 1$
w_2	$1, 2$	$1, 2$

The men's preferences are identical replacing ws with ms and x with y. It is easy to show (by brute force) that for these preferences the unique stable matching is $M = \{(1,1),(2,2)\}$ if $x = y = 1$ and $M = \{(1,2),(2,1)\}$ otherwise. In particular, the (unique) stable matching contains the couple $(1,1)$ if and only if $\mathrm{disj}(x,y) = 0$.

2.3.2 The general case

We are now ready to describe how to construct the women's preferences from a vector $\mathbf{x}$ whose entries are indexed x^i_j with $1 \leq i, j \leq n/2$. The preference structure for w depends on her parity.

even women ($w = w_{2j}$): In this case w's preferences are independent of x. She prefers men m_i with i odd to men m_j with j even. Specifically, her preferences are

$$m_1 \prec_w m_3 \prec_w \cdots \prec_w m_{2i-1} \prec_w \cdots \prec_w m_{n-1} \prec_w m_2 \prec_w m_4 \prec_w \cdots \prec_w m_n.$$

In particular, w prefers all odd men to even men.

odd women ($w = w_{2j-1}$): In this case w's preferences are assigned according to the following groups:

1. Odd men m_{2i-1} such that $x^i_j = 1$
2. Even men m_{2i}
3. Odd men m_{2i-1} such that $x^i_j = 0$

Then w prefers all men in group 1 to group 2, and all men in group 2 to group 3. Within the groups, w's preferences are increasing in i. That is $m_i \prec_w m_{i'}$ if and only if $i < i'$.

The preferences for the men are determined analogously to the women's preferences: just replace ws with ms and $\mathbf{x}$ with $\mathbf{y}$.

2.3.3 Proof of Corollaries 2.6 and 2.8

Corollaries 2.6 and 2.8 will follow immediately from the following claim. We say that a pair $(m, w) \in M \times W$ is ***odd*** if $m = m_{2i+1}$ and $w = w_{2j+1}$ for some $i, j \in \{1, 2, \ldots, n\}$.

Claim 2.15. Assume that $\mathbf{x}, \mathbf{y} \in \{0,1\}^{n^2/4}$ satisfy $|\mathbf{x} \cap \mathbf{y}| \in \{0, 1\}$. Given the preferences described above, a stable matching μ contains an odd pair if and only if $\operatorname{disj}(x, y) = 0$. Further, $\operatorname{disj}(\mathbf{x}, \mathbf{y}) = 1$ if and only if

$$\mu = \{(m_{2i-1}, w_{2i}) \mid 1 \leq i \leq n/2\} \cup \{(m_{2i}, w_{2i-1}) \mid 1 \leq i \leq n/2\} \tag{2.1}$$

is the unique stable matching.

Proof. We first consider the case where $\operatorname{disj}(\mathbf{x}, \mathbf{y}) = 0$. That is, there is a unique index (i, j) such that $x_i^j = y_i^j = 1$. We will show that $(m_{2i-1}, w_{2j-1}) \in \mu$ for any stable matching. To this end, suppose to contrary that $(m_{2i-1}, w_{2j-1}) \notin \mu$ for some stable matching μ. Since μ is stable, (m_{2i-1}, w_{2j-1}) is not a blocking pair. Without loss of generality assume that w_{2j-1} is matched with someone she prefers to m_{2i-1}. Since $x_i^j = 1$, the only men she could prefer to m_{2i-1} are $m_{2i'-1}$ with $x_{i'}^j = 1$ and $i' < i$. Since (i, j) is the unique element in the intersection of $\mathbf{x}$ and $\mathbf{y}$, we must have $y_{i'}^j = 0$, hence $m_{2i'-1}$ prefers all even women to $w_{(2j-1)}$. Since there are $n/2$ even women, $n/2$ odd men, and $m_{2i'-1}$ is matched with an odd woman, some even woman $w_{2j'}$ is matched with an even man. But even women prefer all odd men to even men. Therefore $(m_{2j'-1}, w_{2j'})$ forms a blocking pair, contradicting the assumption that μ is stable. Therefore, (m_{2i-1}, w_{2j-1}) is contained in every stable matching. In particular, every stable matching contains an odd couple.

We now consider the case where $\mathrm{disj}(\mathbf{x}, \mathbf{y}) = 1$. We will show that no stable matching μ contains an odd couple. Suppose to the contrary that $(m_{2i-1}, w_{2j-1}) \in \mu$ for some (i, j). Since x and y are disjoint, $x_j^i = 0$ or $y_j^i = 0$. Without loss of generality, assume that $x_j^i = 0$, so that w_{2j-1} prefers all even men to m_{2i-1}. As before, some even man $m_{2i'}$ is paired with an even woman. But then w_{2j-1} and $m_{2i'}$ mutually prefer each other to their partners in μ, hence μ is not stable. Therefore, no stable matching contains an odd couple.

Finally, we will show that if μ is a stable matching which contains no odd couples then μ must be the matching as in Equation (2.1). Suppose to the contrary that $(m_{2i-1}, w_{2j}) \in \mu$ with, say $i < j$. Then there exists w_{2k} with $k < j$ such that $\mu(w_{2k}) = m_{2\ell-1}$ and $\ell > i$. However, in this case w_{2k} and m_{2i-1} form a blocking pair, a contradiction. Therefore, we must have $\mu(m_{2i-1}) = w_{2i}$ for all i. Similarly, $\mu(w_{2i-1}) = m_{2i}$, which gives the desired result. $\square$

Proof of Corollary 2.6. Suppose a protocol Π finds a stable matching using B bits of communication. Then given an instance of disj of size $n/2$, using the embedding described above, Alice and Bob can compute $\mathrm{disj}(\mathbf{x}, \mathbf{y})$ using B bits of communication. By Claim 2.15, $\mathrm{disj}(\mathbf{x}, \mathbf{y})$ contains an odd pair if and only if $\mathrm{disj}(\mathbf{x}, \mathbf{y}) = 0$. When Π terminates, the men and women can individually detect whether or not there is an odd pair, thereby determining the value of $\mathrm{disj}(\mathbf{x}, \mathbf{y})$. By Theorem 2.4, $B = \Omega(n^2)$, as desired. $\square$

Proof of Corollary 2.8. Suppose a protocol Π determines the stability of any given matching μ using B bits of communication. Then as before, given an instance of disj of size $n/2$, Alice and Bob can use Π to compute $\mathrm{disj}(\mathbf{x}, \mathbf{y})$ using B bits of communication by verifying the stability of μ as described in Equation (2.1). Indeed μ is stable if and only if $\operatorname{disj}(\mathbf{x}, \mathbf{y}) = 1$. □

Remark 2.16. In the proofs above, we only account for communication between the men and women, thus allowing, for example, the men to communicate among each other free of charge. In the distributed setting, the preferences of each player would be known only to that player. Yet our result shows that allowing collaboration among the men and women still yields the (near) optimal $\Omega(n^2)$ lower bound.

2.4 General Proof of Main Results

2.4.1 Embedding disjointness into preferences

Similarly to the proofs given in Section 2.3, the proofs of the remaining results from Section 2.2 follow from embedding suitably large instances of disj into various problems regarding (approximately) stable matchings. In order to prove these remaining results, we reconstruct the embeddings to have the property that small changes in the agents' preferences yield very large changes in the global structure of the stable matchings for these preferences. Informally, we construct the preferences so that resolving blocking pairs resulting from such small changes in agents' preferences creates large rejection chains that ultimately affect most matched pairs.

Preference description

Let $n \in \mathbf{N}$ and let W and M be disjoint sets with $|W| = |M| = n$. We divide the agents into three sets: ***high***, ***mid*** and ***low***, which we denote W_h, W_m and W_l respectively for the women and M_h, M_m and M_l respectively for the men. These sets have sizes

$$|W_h| = |M_h| = \tfrac{1}{2}\delta n$$
$$|W_m| = |M_m| = \tfrac{1}{2}(1-\delta)n$$
$$|W_l| = |M_l| = \tfrac{1}{2}n$$

where δ is a parameter with $0 < \delta \leq 1$, to be chosen later. The low and mid agents preferences will be fixed, while we will use the preferences of the high agents to embed an instance of disjointness of size $(\delta n)^2/4$. We assume that the agents are

$$W = \{w_1, w_2, \ldots, w_n\}, \quad M = \{m_1, m_2, \ldots, m_n\},$$

where in both cases the first $\delta n/2$ agents are high, the next $(1-\delta)n/2$ agents are mid and the remaining $n/2$ agents are low. Since the low and mid agents' preferences are the same for all instances, we describe those first. As before, the agents' preferences are symmetric in the sense that the men's and women's preferences are constructed analogously.

low agents The low women's preferences over men are "in order": $m_1 \prec m_2 \prec \cdots \prec m_n$ (and symmetrically for low men, whose preference over women are "in order"). In particular, each low agent prefers all high agents over all mid agents over all low agents.

mid agents The mid agents prefer low agents over high agents over mid agents. Within each group, the preferences are "in order." Specifically, the mid women have preferences

$$m_{n/2+1} \prec m_{n/2+2} \prec \cdots \prec m_n \prec m_1 \prec m_2 \prec \cdots \prec m_{n/2},$$

and symmetrically for the men.

high agents We use the preferences of each of the high agents to encode a bit vector of length $\delta n/2$. Together, the men and women's preferences thus encode an instance of disj of size $(\delta n)^2/4$. For each $w_i \in W_h$, we denote her bit vector $x^i_1, \ldots, x^i_{\delta n/2}$; the preference list of w_i, from most-preferred to least-preferred, is:

1. men $m_j \in M_h$ such that $x^i_j = 1$;
2. men $m \in M_l$;
3. men $m \in M_m$;
4. men $m_j \in M_h$ such that $x^i_j = 0$.

Within each group, the preferences are once again "in order", i.e. sorted by numeric index. The men's preferences are constructed analogously, with each man m_j encoding the bit vector $y^1_j, \ldots, y^{\delta n/2}_j$ and preferring first and foremost women $w_i \in W_h$ such that $y^i_j = 1$.

Stable Matching Description

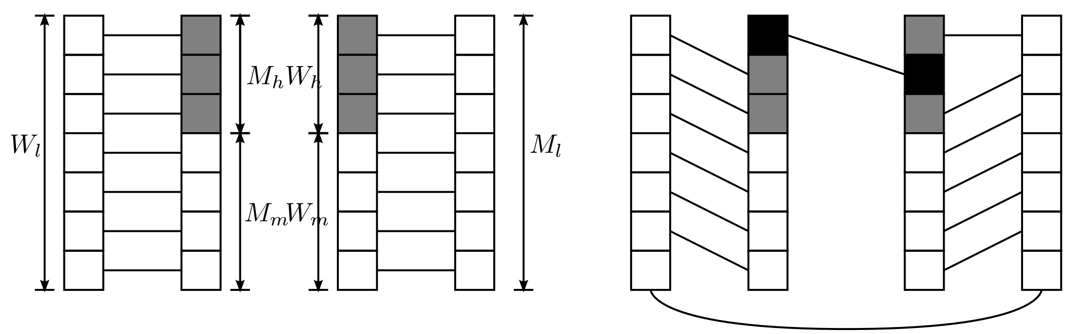

Figure 2.1: The (unique) stable matchings μ_1 for disjoint (left) and μ_0 for uniquely-intersecting (right) instances of the preferences described in Section 2.4.1.

Lemma 2.17. Any instance of the stable marriage problem with preferences described above corresponding to $\text{disj}(\mathbf{x}, \mathbf{y}) = 1$ has a unique stable matching μ_1 given by (see the left side of Figure 2.1)

$$\mu_1 = \left\{(m_i, w_{i+n/2}) \,\middle|\, i = 1, 2, \ldots, n/2\right\}$$
$$\cup \left\{(m_{i+n/2}, w_i) \,\middle|\, i = 1, 2, \ldots, n/2\right\}.$$

Proof. Let μ be a stable matching; we will show that $\mu = \mu_1$. We first argue that every high and mid agent is matched with a low agent in μ. Suppose to the contrary that some $w = w_i$ for $i \leq n/2$ is matched with some $m = m_j$ with $j \leq n/2$ in μ. By the definition of the preferences and the assumption that $\text{disj}(\mathbf{x}, \mathbf{y}) = 1$, at least one of w and m prefers every low agent over their partner. Assume without loss of generality that w prefers all $m' = m_{j'}$ with $j' > n/2$ over m. That is, w prefers all low men over her partner m. Since w is matched with a medium or high man, there must be some low man m' that is matched with a low woman w'. But m' prefers all high and medium women over w'. In particular, he prefers w over w'. Therefore, (w, m') is a blocking pair, so μ is not stable. Thus any stable matching must match low agents to mid or high agents and vice versa.

Now we argue that if $(w_i, m_{j+n/2}) \in \mu$, then we must have $i = j$. The argument for pairs $(w_{i+n/2}, m_j)$ is identical. Suppose that $(w_i, m_{j+n/2}) \in \mu$ with $i < j$. Then there is some $j' < j$ such that $m' = m_{j'+n/2}$ is matched with $w' = w_{i'}$ with $i' > i$. But then (w_i, m') mutually prefer each other, contradicting the stability of μ. We arrive at a similar contradiction if $i > j$, hence we must have $i = j$, as desired. $\square$

Lemma 2.18. Suppose we have a stable matching instance with preferences described above corresponding to $\text{disj}(\mathbf{x}, \mathbf{y}) = 0$, with $\mathbf{x}$ and $\mathbf{y}$ uniquely intersecting. Let $x^\alpha_\beta = y^\alpha_\beta = 1$ be the uniquely-intersecting entry of $\mathbf{x}, \mathbf{y}$. In this case, there exists a unique stable matching

μ_0 given by (see the right side of Figure 2.1)

$$\begin{aligned}\mu_0 = &\{(w_\alpha, m_\beta)\} \cup \left\{(w_i, m_{i+n/2}) \,\middle|\, i < \alpha\right\}\\ &\cup \left\{(w_{i+n/2}, m_i) \,\middle|\, i < \beta\right\}\\ &\cup \left\{(w_i, m_{i+n/2-1}), \alpha < i \leq n/2\right\}\\ &\cup \left\{(w_{i+n/2-1}, m_i), \beta < i \leq n/2\right\} \cup \{(w_n, m_n)\}\,.\end{aligned}$$

Proof. We first argue that $(w_\alpha, m_\beta) \in \mu$ for any stable matching μ for the preferences described above. Since μ is stable, if $(m_\alpha, w_\beta) \notin \mu$, then at least one of w_α and m_β, say w_α, must be matched with someone she prefers over m_β. From w_α's preferences, this implies that $(w_\alpha, m) \in \mu$ for some $m = m_j$ with $j < \beta$ for which $x_j^\alpha = 1$. Since the instance of disj is uniquely intersecting, we must have $y_j^\alpha = 0$. Thus m prefers all low women over w_α. Since at most $n/2 - 1$ medium and high men are matched with low women (indeed m is a high man matched with a high woman) and there are $n/2$ low women, some low woman w is matched with a low man. But then w and m mutually prefer each other, hence forming a blocking pair. Thus, we must have $(w_\alpha, m_\beta) \in \mu$.

The remainder of the proof of the lemma is analogous to the proof of Lemma 2.17 if we remove w_α and m_β from all the agents' preferences. □

Lemma 2.19. The matchings μ_0 and μ_1 from the previous two lemmas satisfy

$$d(\mu_0, \mu_1) \geq (1 - \delta)n.$$

Proof. This follows from the following two observations:

1. All mid women and men $M_m \cup W_m$ have different partners in μ_0 and μ_1.

2. No mid women are matched with mid men in either μ_0 or μ_1.

From these facts, we can conclude that

$$d(\mu_0, \mu_1) = n - |\mu_0 \cap \mu_1| \geq |W_m| + |M_m| = (1 - \delta)n.$$

□

2.4.2 Derivation of main results

In this section we use the construction of Section 2.4.1 to prove all the results formulated in Section 2.2.

Proof of Theorem 2.5. Suppose that Π is a randomized communication protocol (between Alice and Bob) that outputs a $(1 - \varepsilon)$-stable matching μ using B bits of communication. As $\varepsilon < 1/2$, there exists δ sufficiently small such that $\varepsilon < (1 - \delta)/2$. Suppose Π outputs a $(1 - \varepsilon)$-stable matching μ for the preferences described in Section 2.4.1. If $\mathrm{disj}(\mathbf{x}, \mathbf{y}) = 1$, then by Lemma 2.17, μ_1 is the unique stable matching, so $d(\mu, \mu_1) \leq \varepsilon n$.

Suppose $\mathrm{disj}(\mathbf{x}, \mathbf{y}) = 0$. By Lemma 2.18, μ_0 is the unique stable matching, so $d(\mu, \mu_0) \leq \varepsilon n < (1 - \delta)n/2$. Applying Lemma 2.19 and the triangle inequality, we obtain

$$d(\mu_1, \mu) > (1 - \delta)n/2 > \varepsilon n.$$

Thus, if $\mathrm{disj}(\mathbf{x}, \mathbf{y}) = 1$, then $d(\mu, \mu_1) < \varepsilon n$ and if $\mathrm{disj}(\mathbf{x}, \mathbf{y}) = 0$, then $d(\mu, \mu_1) > \varepsilon n$. Given μ, Alice and Bob can compute $d(\mu, \mu_1)$ without communication, so they can use Π to determine the value of $\mathrm{disj}(\mathbf{x}, \mathbf{y})$ using B bits of communication. Thus, $B = \Omega(n^2)$ by Theorem 2.4, as desired. □

Proof of Theorem 2.7. Suppose that Π is a randomized communication protocol that determines whether a given matching μ is stable or ε-unstable with respect to given preferences using B bits of communication. As $\varepsilon < 1$, there exists δ sufficiently small such that $1-\delta > \varepsilon$.

Let μ_1 be the matching defined in Lemma 2.17; by that lemma, if $\text{disj}(\mathbf{x}, \mathbf{y}) = 1$, then μ_1 is stable (with respect to the preferences described in Section 2.4.1). By Lemmas 2.18 and 2.19, if $\text{disj}(\mathbf{x}, \mathbf{y}) = 0$, then μ_1 is ε-unstable. Thus, if Π determines whether μ_1 is stable or ε-unstable, then Π also determines the value of $\text{disj}(\mathbf{x}, \mathbf{y})$, hence $B = \Omega(n^2)$ by Theorem 2.4. □

Proof of Theorem 2.11. Suppose that Π is a randomized communication protocol that for a given pair (w, m) determines whether $(w, m) \in \mu$ for some (every) stable matching μ using B bits of communication. Set $\delta = 1$. By choosing preferences as in Section 2.4.1 and taking $(w, m) = (w_n, m_n)$, by Lemmas 2.17 and 2.18, (w, m) is in some (equivalently every) stable matching for the given preferences if and only if $\text{disj}(\mathbf{x}, \mathbf{y}) = 0$. Thus, once again by Theorem 2.4, $B = \Omega(n^2)$. □

Proof of Theorem 2.13. Suppose that Π is a randomized communication protocol that outputs εn pairs contained in some (every) stable matching using B bits of communication. Choose preferences as described in the Section 2.4.1 with some $0 < \delta < \varepsilon$, say $\delta = \varepsilon/2$. Recall from the proof of Lemma 2.19 that no agents in W_m and M_m are ever matched with one another in a stable matching. Therefore, since $|W_m| + |M_m| = (1 - \delta)n > (1 - \varepsilon)n$ and since Π outputs εn pairs, we have that Π must output some pair (w, m) with $w \in W_m$ or $m \in M_m$. Recall from the proof of Lemma 2.19 that knowing the stable partner of any agent in W_m or M_m reveals the value of $\text{disj}(\mathbf{x}, \mathbf{y})$. Thus, by Theorem 2.4, $B = \Omega(n^2)$. □

Proof of Theorem 2.14. We prove Part (a) of the theorem. Suppose there is a randomized algorithm A that computes a $(1 - \varepsilon)$-stable matching using B Boolean queries to the women and men. We will use A to construct a B-bit communication protocol for the approximate stable marriage problem. The protocol works as follows. Alice and Bob both simulate A. Whenever A queries the women's preferences, Alice sends the result of the query to Bob (since Alice knows the women's preferences). Symmetrically, when A queries the men's preferences, Bob sends Alice the result of the query. This protocol uses B bits of communication. Thus, by Theorem 2.5, we must have $B = \Omega(n^2)$, as desired.

Parts (b)–(d) follow similarly from Theorems 2.7, 2.11 and 2.13, respectively. □

Chapter 3

Approximation Algorithms

In Chapter 2, we analyzed one measure of the distributed complexity of the stable marriage problem, namely its communication complexity. In this chapter, we turn our attention to another measure of complexity: time complexity. We work in the (CONGEST) computational model described in Section 1.3: each agent is represented by a single processor, which initially only knows that agent's preferences. The processors communicate in synchronous rounds, and the time complexity of the (distributed) SMP is the number of rounds necessary to find a stable matching (see Section 1.3 for details). In this computational model, Theorem 1.18 shows that finding an (exactly) stable matching requires time $\Omega(n)$.

Partially due to lower bounds such as Theorems 1.16 and 1.18 for exactly stable matchings, there has recently been interest in approximate versions of the stable marriage problem [1, 9, 10, 17, 26], where the goal is to find a matching which is "almost stable." We reiterate that there is no consensus in the literature on precisely how to measure almost stability, but typically almost stability requires that a matching induces relatively few blocking pairs. Eriksson and Häggström [9] argue that, "the proportion of blocking pairs among all possible pairs is usually the best measure of instability." Using a finer notion of almost stability, Floréen, Kaski, Polishchuk, and Suomela show [10] that for *bounded* preference lists, truncating the Gale-Shapley algorithm after boundedly many communication rounds yields a matching that induces at most $\varepsilon |\mu|$ blocking pairs. Here $|\mu|$ is the size of the matching produced. More recently, Hassidim, Mansour and Vardi [17] show a similar result in a more restrictive "local" computational model, so long as the men's preferences are chosen uniformly at random.

In this chapter, we describe a distributed algorithm which for any fixed $\varepsilon > 0$ finds a $(1 - \varepsilon)$-stable matching (Definition 1.25) μ using a poly-logarithmic number of rounds. Our algorithm can be seen as a generalization of the Gale-Shapley algorithm, where each agent is allowed to make multiple proposals in each round. Thus, our result can be seen as a generalization of the result of Floréen, Kaski, Polishchuk, and Suomela [10].

We note that Kipnis and Patt-Shamir [26] give an algorithm which finds an almost stable matching using $O(n)$ communication rounds in the worst case, using a finer notion of approximate stability than we consider. Specifically, in their notion of almost-stability—which we call ε-blocking stability (Definition 1.26)—a matching is almost stable if no pair of agents can both improve their match by more than an ε-fraction of their preference list by deviating from their assigned partners. They also prove an $\Omega(\sqrt{n}/\log n)$ round lower bound for finding an almost stable matching for ε-blocking stability. Thus, the algorithm described in this chapter is exponentially faster than the best possible algorithm which finds an almost stable matching in the sense of [26].

The material in this chapter originally appeared in [35].

3.1 Chapter Overview

In this chapter, we describe a deterministic distributed algorithm, **ASM**, which produces a $(1-\varepsilon)$-stable matching in $O(\log^5(n))$ rounds. In order to obtain this sub-polynomial run-time, we must use a coarser notion of approximation than ε-blocking stability, as Kipnis and Patt-Shamir [26] prove an $\Omega(\sqrt{n}/\log n)$ round lower bound for their model. We remark that after removing an arbitrarily small fraction of "bad" agents, the output of **ASM** is ε-blocking stable as well. We further describe a faster randomized variant of **ASM** which runs in $O(\log^2(n))$ rounds. For preferences which are "almost regular," (and in particular for complete or bounded preferences) this run-time can be improved to $O(1)$.

Theorem 3.1. There exists a deterministic distributed algorithm **ASM** which produces a $(1-\varepsilon)$-stable matching in $O(\log^5(n))$ communication rounds. A randomized variant of the algorithm, **RandASM** runs in $O(\log^2(n))$ rounds for general preferences, and can be improved to $O(1)$ rounds for almost regular (and in particular complete or bounded) preferences.

ASM can be viewed as a generalization of the classical Gale-Shapley algorithm [11] which allows for multiple simultaneous proposals by the men and acceptances by the women. In **ASM**, the agents quantize their preferences into $O(\varepsilon^{-1})$ quantiles of equal size. In each step of the algorithm, the men propose to all women in their best nonempty quantile. Each woman accepts proposals only from her best quantile receiving proposals. A maximal matching is then found among the accepted proposals, and matched women reject men they do not prefer to their matches. This procedure is iterated until a large fraction of men are either matched, or have been rejected by all women.

The analysis of our algorithm follows in two steps. We first show that by quantizing preferences, the matching found by **ASM** cannot contain a large fraction of blocking pairs among the matched (or rejected) agents. We bound the number of blocking pairs from the remaining "bad" agents by showing there are few such agents, and that only a small fraction can participate in many blocking pairs.

We refer the reader to Sections 1.3 and 1.4 for background on the distributed computational model (CONGEST model), and almost stable matchings. The remainder of the chapter is organized as follows. In Section 3.2 we give an overview of methods for computing maximal matchings which our algorithm will require as subroutines. Section 3.3 describes **ASM** and its subroutines in detail and states basic guarantees for the subroutines. Section 3.4 proves the performance guarantees for **ASM**. Finally, in Section 3.5 we describe the randomized variants of **ASM**, with some details deferred to Section 3.6.

3.2 Preliminaries

3.2.1 Computational model

We describe **ASM** in terms of the CONGEST model formalized by Peleg [39]. In this distributed computational model, each agent $v \in M \cup W$ represents a processor. Given preferences $\mathcal{P}$, the communication links between the agents are given by the set of edges E in the communication graph G. Communication is performed in synchronous rounds. Each communication round occurs in three stages: first, each processor receives messages (if any) sent from its neighbors in G during the previous round. Next, each processor performs local calculations based on its internal state and any received messages. We make no restrictions on the complexity of local computations. Finally, each processor sends short ($O(\log n)$ bit) messages to its neighbors in G—the processor may send distinct messages to distinct neighbors. In the CONGEST model, complexity is measured by the number of communication rounds needed to solve a problem.

Remark 3.2. Although the CONGEST model allows for unbounded local computation during each round, the computations required by **ASM** can be implemented in linear or near-linear time in each processor's input.

3.2.2 Maximal matchings

As a subroutine, **ASM** requires a method for computing maximal matchings in a graph.

Definition 3.3. A matching μ is a ***maximal matching*** if it is not properly contained in any larger matching. Equivalently, μ is maximal if and only if every $v \in V$ satisfies precisely one of the following conditions:

1. there exists a unique $u \in V$ with $(v, u) \in \mu$;
2. for all $u \in N(v)$ there exists $v' \in V$ with $v' \neq v$ such that $(v', u) \in \mu$. Here, we use $N(v)$ to denote the neighborhood of v, i.e., $N(v) = \{u \in V \mid (v, u) \in E\}$.

For the deterministic version of our algorithm, we invoke the work of Hańćkowiak, Karoński, and Panconesi [16] who give a deterministic distributed algorithm which finds a maximal matching in a poly-logarithmic number of rounds.

Theorem 3.4 (Hańćkowiak, et al. [16])**.** There exists a deterministic distributed algorithm, **MaximalMatching**, which finds a maximal matching in a communication graph $G = (V, E)$ in $O(\log^4(n))$ rounds, where $n = |V|$.

We remark that while the authors of [16] do not explicitly use the CONGEST model of computation, their algorithm can easily be implemented in this model.

The randomized variants of **ASM** require faster (randomized) subroutines for computing maximal and "almost maximal" matchings in a communication graph. In Section 3.6, we describe how to modify an algorithm of Israeli and Itai [22] to give the necessary results.

3.3 Deterministic Algorithm Description

In this section, we describe in detail the almost stable matching algorithm, **ASM**. The main algorithm invokes the subroutine **QuantileMatch** which in turn calls **ProposalRound**. In Section 3.3.1 we introduce notation, and describe the internal state of each processor during the execution of **ASM**. Section 3.3.2 contains a description of the **ProposalRound** subroutine, while Section 3.3.3 describes the **QuantileMatch** subroutine. Finally, Section 3.3.4 describes **ASM**.

3.3.1 The state of a processor

In our algorithm, we assume that each agent is represented by an independent processor. Each processor has a unique id and a gender (male or female) both of which are known to that processor. The only global information known to each processor is (an upper bound on) n, the total number of processors. At each step of the algorithm, we specify the state of each processor as well as any messages the processor might send or receive. Throughout, k is a parameter (the number of quantiles) to be chosen later. The state of an agent v consists of:

- Quantized preferences $Q_1, Q_2, \ldots, Q_k$ where we denote $Q = \bigcup Q_i$. For each u in v's preference list, denote $q(u) = \lceil \mathcal{P}_v(u)/k \rceil$. Initially we set $Q_i = \{u \mid q(u) = i\}$, so that Q_1 is the set of v's $\deg(v)/k$ favorite partners, Q_2 is her next favorite $\deg(v)/k$, and so on. We call Q_i v's i***th quantile***. If we wish to make explicit the agent to whom the preferences belong, we may adorn these symbols with a superscript. For example, Q_i^v is v's ith quantile. Throughout the execution of the algorithm, elements may be removed from Q and the Q_i, but elements will never be added to any of these sets.

- A partner p (possibly empty). The partner p is v's current partner in the matching μ our algorithm constructs. To emphasize that p is agent v's partner, we will write $p(v)$. The (partial) matching μ produced by the algorithm at any step is given by $\mu = \{(p(w), w) \mid w \in W, p(w) \neq \varnothing\}$.

Additionally, subroutines of our algorithm will require each processor to store the following variables:

- A set G_0 of "neighbors" of the opposite sex which correspond to accepted proposals.

- A partner p_0 in a matching found in the graph determined by G_0.

Thus each agent knows their preferences, partners (if any) as well as any of their accepted proposals from the current round (stored in G_0). The men $m \in M$ hold the following additional information:

- A set A of "active" potential matches, initially set to Q_1.

3.3.2 The ProposalRound subroutine

At the heart of our algorithm is the **ProposalRound** subroutine (Algorithm 2). **ProposalRound** works in 5 steps which are described in Algorithm 2.

We observe that if each agent v takes $k = \deg v$, then **ProposalRound** mimics the classical (extended) Gale-Shapley algorithm [11, 15]. In this case, each man proposes to his most

Algorithm 2 ProposalRound(Q, k, A)

Step 1: Each man m proposes to all women in A^m by sending each $w \in A$ the message PROPOSE.

Step 2: Each woman w receiving proposals responds with the message ACCEPT to all proposals from her most preferred quantile Q_i^w from which at least one man proposed in Step 1.

Step 3: Let G_0 denote the bipartite graph G_0 of accepted proposals from Step 2. The agents compute a maximal matching μ_0 in G_0, using **MaximalMatching**(G_0), storing their match in G_0 as p_0.

Step 4: Each woman w matched in μ_0 sends REJECT to all men $m \in Q^w$ in a lesser or equal quantile to her partner $p_0(w)$ in μ_0 other than $p_0(w)$. She then removes all of these men from Q^w and the corresponding Q_i^w. The matched women then set $p \leftarrow p_0$, so the partial matching μ now contains the edge $(p_0(w), w)$. Any man m matched in μ_0 sets $p \leftarrow p_0$ and sets $A \leftarrow \varnothing$.

Step 5: The men remove all w from whom they received the message REJECT from their preferences Q, the various Q_i, and A. If a man m receives a rejection from his match $p(m)$ from a previous round, he sets $p \leftarrow \varnothing$.

---

favored woman that has not yet rejected him, and each woman rejects all but her most favored suitor. In this case, computing a maximal matching is trivial, as the accepted proposals already form a matching. The general case has one crucial feature in common with the Gale-Shapley algorithm, which follows immediately from the description of **ProposalRound**.

Lemma 3.5 (Monotonicity)**.** Once a woman w has $p(w) \neq \varnothing$ in some execution of **ProposalRound**, she is guaranteed to always have $p(w) \neq \varnothing$ after each subsequent execution of **ProposalRound**. Further, once matched, she will only accept proposals from men in a strictly better quantile than $p(w)$. That is, if $p(w) \in Q_i^w$, then w will only accept proposals from men $m \in Q_j^w$ with $j < i$.

3.3.3 The QuantileMatch subroutine

Here we describe the **QuantileMatch** subroutine (Algorithm 3), which simply iterates **ProposalRound** until each man m has either been rejected by all women in A^m or is matched with some woman in A^m. In either case, $A^m = \varnothing$ when **QuantileMatch** terminates. We will argue that k (the number of quantiles) iterations suffice.

Algorithm 3 QuantileMatch(Q, k)

```
i ← min {i | Q_i ≠ ∅} ∪ {k} (male only)
if p = ∅ then
    A ← Q_i (male only)
end if
for i ← 1 to k do
    ProposalRound(Q, k, A)
end for
```

Lemma 3.6 (**QuantileMatch** guarantee)**.** At the termination of **QuantileMatch**(Q, k) every man m satisfies $A^m = \varnothing$. In particular, each man who had $A^m \neq \varnothing$ before the first iteration of the loop in **QuantileMatch** has either been rejected by all women in A^m or is matched with some woman in A^m.

Proof. Suppose a woman w receives proposals in the first iteration of the loop in **QuantileMatch**. If she is matched with one of these suitors when **ProposalRound** terminates, she rejects all other men and receives no further proposals during the current **QuantileMatch**. On the other hand, if she is not matched with one of these suitors after the first round, then by the maximality of the matching found in Step 3 of **ProposalRound**, all of the suitors in her best quantile receiving proposals are matched with other women. Thus, in the next iteration, she only receives proposals from men in strictly worse quantiles than she accepted in the first. Similarly, in each iteration of the loop, her best quantile receiving proposals (if any) is strictly worse than the previous iteration. Therefore, after k iterations, no woman will receive proposals, hence each man m must have $A^m = \varnothing$. $\square$

3.3.4 The ASM algorithm

In this section, we describe the main algorithm **ASM** (Algorithm 4). The idea of **ASM** is to iterate **QuantileMatch** until a large fraction men with high degree are either matched or have been rejected by all acceptable partners. We call such men ***good***. By iterating **QuantileMatch** a constant number of times, we can ensure that the fraction of good men is close to 1. In order to bound the number of blocking pairs from men which are ***bad*** (not good), we must ensure that bad men comprise only a small fraction of agents with relatively high degree. To this end, we only allow men who are potentially involved in many blocking pairs (that is, with $|Q|$ relatively large) to participate in later calls to **QuantileMatch**.

Algorithm 4 **ASM**$(\mathcal{P}, \varepsilon, n)$

```
k ← ⌈8ε⁻¹⌉ , δ ← ε/8
for all i ≤ k do
    Q_i ← {v | q(v) = i}
end for
Q ← ⋃_i Q_i, p ← ∅
for i ← 0 to log n do
    if |Q| ≥ 2^i then
        for j ← 1 to 2δ⁻¹k do
            QuantileMatch(Q, k)
        end for
    end if
end for
```

3.4 Performance Guarantees

Here we analyze the performance of **ASM** and its subroutines. The run-time guarantee (Theorem 3.13) is a simple consequence of the description of **ASM** and its subroutines. To prove the approximation guarantee (Theorem 3.7), we consider blocking edges from two sets of men separately. We call a man m ***good*** if when **ASM** terminates, he is either matched or has been rejected by all of his acceptable partners. A man who is not good is ***bad***. We denote the sets of good and bad men by G and B, respectively.

Theorem 3.7 (Approximation guarantee)**.** The matching μ output by **ASM** induces at most $\varepsilon\,|E|$ blocking pairs with respect to $\mathcal{P}$. Thus μ is $(1-\varepsilon)$-stable.

3.4.1 Bounding blocking pairs from good agents

We bound the number of blocking pairs from good men in two steps. First we show that the good men are not involved in any $(2/k)$-blocking pairs (see Definition 1.26). Next, we show that as a result, the good men can only be incident with a small fraction of blocking pairs.

Lemma 3.8 ($(2/k)$-blocking-stability of good men)**.** Let $m \in G$ be good. Then m is not incident with any $(2/k)$-blocking pairs.

Proof. Suppose $m \in G$ and that (m, w) is $(2/k)$-blocking. First consider the case where m is matched, $p(m) \neq \varnothing$. Since m's preferences are divided into k quantiles, w must be in a strictly better quantile than $p(m)$. Thus, m must have proposed to w in a strictly earlier call to **QuantileMatch** than the call in which he was matched with $p(m)$. Thus, by Lemma 3.6, m must have been rejected by w, implying that w was matched with a man m' in the same or better quantile than m in this round. By Lemma 3.5, w's partner when **ASM** terminates is at least as desirable as m'. This contradicts that (m, w) is ε-blocking.

On the other hand, if $p(m) = \varnothing$, then since m is good, he must have been rejected by all of his acceptable partners, and in particular, by w. Thus, as in the previous paragraph, w must be matched with a man in the same or better quantile than m. □

Lemma 3.9 (Few non-$(2/k)$-blocking pairs)**.** There are at most $4\,|E|\,/k$ blocking pairs which are not $(2/k)$-blocking.

Proof. Suppose (m, w) is a blocking pair which is not $(2/k)$-blocking. Thus, at least one of the following holds:

$$\mathcal{P}_m(w) - \mathcal{P}_m(p(m)) \leq 2\deg(m)/k \tag{3.1}$$

$$\mathcal{P}_w(m) - \mathcal{P}_w(p(w)) \leq 2\deg(w)/k, \tag{3.2}$$

where by convention we take $\mathcal{P}_m(\varnothing) = \deg(m) + 1$. Let E_N denote the set of blocking pairs which are not $(2/k)$-blocking. For each m, the number of edges satisfying (3.1) is at most $2\deg(m)/k$, and similarly for the women. Thus

$$|E_N| \leq \sum_{m \in Y} 2\deg(m)/k + \sum_{w \in X} 2\deg(w)/k = 4\,|E|\,/k,$$

as desired. □

Lemma 3.8 shows that no good agent is involved in any $(2/k)$-blocking pairs. Combining Lemmas 3.8 and 3.9, we can bound the number of blocking pairs incident with good men. All that remains is to bound the number of $(2/k)$-blocking pairs incident with bad men. In the next section, we show that the proportion of bad men is small (at most δn), and bound the number of $(2/k)$-blocking pairs they contribute. We remark that by Lemma 3.9 and the lower bound of Kipnis and Patt-Shamir [26], we cannot hope to have all men be good in $o(\sqrt{n}/\log n)$ rounds.

3.4.2 Bounding blocking pairs from bad agents

In this section, we prove the following bound on the number of blocking pairs contributed by the bad men at the termination of **ASM**. Throughout the section, for simplicity of notation, we assume that $\log n$ is an integer.

Lemma 3.10 (Bad men guarantee)**.** At the termination of **ASM**, for any $\delta \leq \frac{1}{2}$ the bad men contribute at most $4\delta\,|E|\,(2/k)$-blocking pairs.

The proof of Lemma 3.10 is in two parts corresponding to guarantees for each of the two nested loops in **ASM**. We refer to men m with $|Q^m| \geq 2^i$ as ***active in the*** i***th iteration*** of the outer loop; the remaining men are ***inactive*** in the ith iteration.

Lemma 3.11 (Few bad men)**.** When the inner loop in **ASM** terminates, at most a δ-fraction of active men are bad.

Proof. Let $\mathcal{A}$ denote the set of active men before executing the inner loop in **ASM**. Suppose that after ℓ iterations of the inner loop, there are b bad men in $\mathcal{A}$. We claim that there must have been at least b bad agents in every iteration of the inner loop. To see this, first note that by Lemma 3.5, the number of matched agents (and hence matched men) can only increase with each call to **ProposalRound**. Second, if a man is rejected by all women on his preference list, he will never become bad. Therefore, the number of good agents can only increase with each iteration of the inner loop. Thus there must have been at least b bad men after each of the ℓ iterations of the inner loop.

Suppose m was bad before some call to **QuantileMatch**, so that $A^m \neq \varnothing$. By Lemma 3.6, after **QuantileMatch** m is either matched, or has been rejected by all women $w \in A^m$. In the former case, $p(m)$ rejected all men in her quantile containing m. In either case, m witnessed the rejection of a quantile of men—either by precipitating the rejection of $p(m)$'s quantile, or by being rejected by all women in A. Notice that the number of women who are matched with new partners during an iteration of the outer loop cannot exceed $|\mathcal{A}|$, as if $|\mathcal{A}|$ women did receive new partners, all active men would be matched. Therefore, the women can send at most $k\,|\mathcal{A}|$ quantile rejections (after which all active men will be rejected by all women). Similarly, the men can receive at most $k\,|\mathcal{A}|$ quantile rejections. Thus, in total, the active men can witness at most $2k\,|\mathcal{A}|$ quantile rejections. Therefore, if there are b bad men after ℓ calls to **QuantileMatch**, we must have $b\ell \leq 2k\,|\mathcal{A}|$. Choosing $\ell = 2\delta^{-1}k$ gives the desired result. □

We say that a man m is ***bad in the ith iteration*** of the outer loop in **ASM** if m became bad during the ith iteration and $|Q^m| < 2^i$. We denote the set bad men in the i iteration by B_i, so that $B = B_1 \cup B_2 \cup \cdots \cup B_{\log n}$. Thus, $m \in B_i$ is bad and will not participate in any further calls to **QuantileMatch**, so he will be bad when **ASM** terminates.

Lemma 3.12 (Few $(2/k)$-blocking pairs)**.** Each $m \in B_i$ participates in fewer than 2^i $(2/k)$-blocking pairs at the termination of **ASM**.

Proof. We will show that each bad $m \in B$ participates in at most $|Q^m|$ $(2/k)$-blocking pairs, whence the lemma follows. To this end, notice that if $w \notin Q^m$, then w must have rejected m in some call to **QuantileMatch**. Therefore, w must have been matched with some m' that is in the same or better quantile as m. By Lemma 3.5, when **ASM** terminates, w is still matched with someone in at least as desirable quantile as m, implying that (m, w) is not $(2/k)$-blocking. Thus, every $(2/k)$-blocking pair (m, w) must have $w \in Q^m$. □

Proof of Lemma 3.10. Let $G_i \subseteq G$ be the set of men which are good at the termination of **ASM** and active after the ith iteration of the outer loop in **ASM**. Then we have $G = G_1 \cup G_2 \cup \cdots \cup G_{\log n}$. By Lemma 3.11, if there were b men which became bad in the i-

th iteration of the outer loop of **ASM**, there were $\frac{1-\delta}{\delta}b$ good men still active during the ith iteration. Since the number of good men can only increase in subsequent iterations, we have

$$|B_i \cup B_{i+1} \cup \cdots \cup B_{\log n}| \leq b \leq \frac{\delta}{1-\delta} |G_i \cup G_{i+1} \cup \cdots \cup G_{\log n}| . \tag{3.3}$$

Applying (3.3), we can greedily form disjoint sets

$$H_{\log n} \subseteq G_{\log n},\; H_{\log n - 1} \subseteq G_{\log_n -1} \cup G_{\log n}, \ldots, H_1 \subseteq G$$

such that for all i, H_i is active (and good) in the ith iteration and $|H_i| = \frac{1-\delta}{\delta} |B_i|$. Then we compute

$$\begin{aligned}
\sum_{m \in B} |Q^m| &= \sum_{i=1}^{\log n} \sum_{m \in B_i} |Q^m| \\
&\leq \sum_{i=1}^{\log n} |B_i| \, 2^i \\
&\leq \sum_{i=1}^{\log n} \frac{2\delta}{1-\delta} |H_i| \, 2^i \\
&\leq \frac{2\delta}{1-\delta} \sum_{m \in G} |Q^m| \\
&\leq \frac{2\delta}{1-\delta} |E| .
\end{aligned}$$

The first inequality holds by Lemma 3.12, while the second holds by the choice of the H_i and the definition of the G_i. □

3.4.3 Approximation guarantee

Proof of Theorem 3.7. By Lemma 3.9, there are at most $4|E|/k$ blocking pairs which are not $(2/k)$-blocking. By Lemma 3.8, all $(2/k)$-blocking pairs are incident with B. Finally, by Lemma 3.10, the bad men contribute at most $4\delta |E|$ blocking pairs for $\delta \leq 1/2$. Therefore, the total number of blocking pairs is at most $4(\delta + 1/k)\,|E|$. Choosing $\delta = \varepsilon/8$ and $k = \lceil 8/\varepsilon \rceil$ gives the desired result. □

3.4.4 Run-time Guarantee

Theorem 3.13. **ASM**(P, ε, n) runs in $O(\varepsilon^{-3} \log^5(n))$ communication rounds.

Proof. Notice that the only communication between processors occurs in **ProposalRound**. **ASM**(P, ε, n) iterates **QuantileMatch**(P, k) a total of $O(\varepsilon^{-2} \log n)$ times, while quantile match invokes **ProposalRound**(Q, k, A) $O(\varepsilon^{-1})$ times. Finally, each step of **ProposalRound** can be performed in $O(1)$ communication rounds, except for Step 3, which calls **MaximalMatching**. By [16], **MaximalMatching** runs in $O(\log^4 n)$ communication rounds. Thus, **ASM** requires $O(\varepsilon^{-3} \log^5(n))$ communication rounds, as claimed. □

Remark 3.14. While the CONGEST model allows for unbounded local computation in each round, the local computations required by **ASM** are quite simple. In fact, each communication round can easily be implemented in nearly-linear time in n. Thus the synchronous run-time of **ASM** is $\tilde{O}(n)$. To our knowledge, this gives the first distributed algorithm whose synchronous run-time is sub-quadratic in n, even for unbounded preferences.

3.5 Randomized Algorithms

The main source of complexity in **ASM** comes from finding a maximal matching. While Hańćkowiak, Karoński, and Panconesi's algorithm [16] is the most efficient known deterministic algorithm, faster randomized algorithms are known. Specifically, we consider the algorithm of Israeli and Itai [22]. They describe a simple randomized distributed algorithm which finds a maximal matching in expected $O(\log n)$ rounds. By simply replacing **MaximalMatching** with a truncated version Israeli and Itai's algorithm, we obtain a faster randomized algorithm for finding almost stable matchings. We refer the reader to Section 3.6 for details on the guarantees for Israeli and Itai's algorithm.

3.5.1 General preferences

Theorem 3.15. There exists a randomized distributed algorithm, **RandASM**$(P, \varepsilon, n, \delta)$, which for any $\delta, \varepsilon > 0$ finds a $(1-\varepsilon)$-stable matching with probability at least $1-\delta$ in $O(\varepsilon^{-3}\log^2(n/\delta\varepsilon^3))$ rounds.

Proof (sketch). We take **RandASM** to be exactly the same as **ASM**, except that we use Israeli and Itai's algorithm [22] for the **MaximalMatching** subroutine. Specifically, for **MaximalMatching**, we iterate **MatchingRound** (see Section 3.6) $O(\log(n/\delta\varepsilon^3))$ times. By Corollary 3.18, each call to **MaximalMatching** will succeed in finding a maximal matching with probability at least $1-O(\delta\varepsilon^3/\log n)$. Since **RandASM** calls **MaximalMatching** $O(\varepsilon^{-3}/\log n)$ times, by the union bound, every call to **MaximalMatching** succeeds with probability at least $1-\delta$. The remaining analysis of **RandASM** is identical to that of **ASM**. □

3.5.2 Almost-regular preferences

For $\alpha \geq 1$, we call preferences $\mathcal{P}$ ***α-almost-regular*** if $\max_{m\in M}\deg m \leq \alpha \min_{m\in M}\deg m$. For example, complete preferences (where all men rank all women) are 1-almost-regular, while uniformly bounded preferences are α-almost-regular for $\alpha = \max_{m\in M}\deg m$. From an algorithmic standpoint, α-almost-regular preferences are advantageous because in order to bound the proportion of blocking edges from bad men, it suffices only to bound the number of bad men. By Lemma 3.11, to obtain such a guarantee, one need only iterate **QuantileMatch** $O(1)$ rounds (instead of $O(\log n)$ times as required by **ASM**).

Further, for α-almost-regular preferences, we can relax our requirement that **MaximalMatching** actually find a maximal matching. We say that a agent v is ***unmatched*** in G_0 if v does not satisfy property 1 or 2 in Definition 3.3. We call a subroutine **AMM**(η, δ) which finds a matching in which only an η-fraction of agents are left unmatched with probability at least $1 - \delta$ (see Section 3.6 for details). These unmatched agents are immediately removed from play. With these simplifications, we obtain the following result.

Theorem 3.16. There exists a randomized distributed algorithm **AlmostRegularASM**$(P, \varepsilon, \delta, \alpha)$ which for α-almost-regular preferences P finds a $(1-\varepsilon)$-stable matching with probability at least $1 - \delta$. The run-time of **AlmostRegularASM**$(P, \varepsilon, \delta, \alpha)$ is $O(\alpha\varepsilon^3 \log(\alpha/\delta\varepsilon))$ rounds.

Proof (sketch). **AlmostRegularASM**$(P, \varepsilon, \delta, \alpha)$ works by iterating **QuantileMatch** $O(\alpha\varepsilon^{-2})$ times, which by Lemma 3.11 implies that only $\varepsilon/4\alpha$ fraction of men are bad.

We modify **ProposalRound** to call **AMM**(η, δ') instead of **MaximalMatching**. **AMM** runs in $O(\log((\eta\delta')^{-1}))$ and finds a $(1-\eta)$-maximal matching with probability $(1-\delta')$. Since **AMM** is called $O(\alpha\varepsilon^{-3})$ times, choosing $\eta = O(\varepsilon^4/\alpha)$ and $\delta' = O(\delta\varepsilon^3/\alpha)$, **AMM** will leave at most an $\varepsilon/4\alpha$ fraction of men unmatched in any call **AMM** with probability at least $1-\delta$, by the union bound. Such unmatched men are immediately removed from play.

By the preceding two paragraphs, **AlmostRegularASM** produces a matching in which at most an $\varepsilon/2\alpha$ fraction of men are either bad or unmatched. By α-almost-regularity, these men can contribute at most $\frac{\varepsilon}{2}|E|$ blocking pairs. The remaining men are good, and therefore by Lemmas 3.8 and 3.9[1] contribute at most $\frac{\varepsilon}{2}|E|$ blocking pairs. □

[1]Although these lemmas were proven assuming that **MaximalMatching** found a maximal matching (not an almost maximal matching) the proofs remain valid as long as the small fraction of unmatched agents immediately remove themselves from play.

3.6 Randomized Maximal and Almost Maximal Matchings

Israeli and Itai's [22] algorithm for finding a maximal matching works by identifying a sparse subgraph of G, then finding a large matching μ_1 in the sparse subgraph. The edges and incident vertices of μ_1, as well as remaining isolated vertices, are removed from G resulting in a subgraph G_1. The process is iterated, giving a sequence of subgraphs $G_1, G_2, \ldots$ and matchings $\mu_1, \mu_2, \ldots$, until $G_k = \varnothing$. At this point, $\mu = \bigcup_{i=1}^{k} \mu_i$ is a maximal matching. We give pseudocode for Israeli and Itai's main subroutine, which we call **MatchingRound**, in Algorithm 5. In [22], Israeli and Itai prove the following performance guarantee for **MatchingRound**.

Algorithm 5 MatchingRound(G): Finds a large matching in a graph

1. Each $v \in V$ picks a neighbor w uniformly at random, forms oriented edge (v, w).
2. Each $v \in V$ with $\deg_{in}(v) > 0$ picks one in-coming edge (w, v) uniformly at random, deletes remaining in-edges. Let G' be the (undirected) graph formed by the chosen edges with orientation ignored.
3. Each $v \in V$ with $\deg_{G'}(v) > 0$ chooses one incident edge (v, w) uniformly at random.
4. The matching μ_1 consists of edges $(v, w) \in G'$ which were chosen by both v and w in the previous round. $G_1 = (V_1, E_1)$ is the induced subgraph of G formed by removing all vertices contained in μ_1 and any remaining isolated vertices from G.
5. Output (G_1, μ_1).

Lemma 3.17. (Israeli and Itai [22]) There exists an absolute constant $c < 1$ such that on input $G = G_0 = (V_0, E_0)$, the resulting graph $G_1 = (V_1, E_1)$ found by **MatchingRound** satisfies $\mathbf{E}(|V_1|) \leq c\,|V_0|$.

As a consequence of Lemma 3.17, we obtain the following useful result.

Corollary 3.18. Let $\eta > 0$ be a parameter. Then $s = O(\log(n/\eta))$ iterations of **MatchingRound** suffice to produce a maximal matching in G with probability at least $1 - \eta$.

Proof. By Lemma 3.17, we have $\mathbf{E}(|V_s|) \leq c^s n$. Therefore, applying Markov's inequality gives

$$\Pr(|V_s| \geq 1) \leq \frac{\mathbf{E}(|V_s|)}{1} \leq c^s n.$$

The result follows by taking $s \geq \log(n/\eta)/\log(c^{-1})$. □

The almost regular variant of **ASM** only requires a subroutine that we finds matchings which are almost maximal.

Definition 3.19. Let $G = (V, E)$ be a communication graph and $\mu \subseteq E$ a matching in G. For $0 < \eta \leq 1$, we say that μ is $(1-\eta)$-***maximal*** if the set V' of vertices not satisfying conditions 1 or 2 in Definition 3.3 satisfies $|V'| \leq \eta\,|V|$.

We can apply Lemma 3.17 to give a constant round algorithm which finds almost maximal matchings.

Corollary 3.20. There exists a randomized distributed algorithm **AMM**(G, η, δ) which finds a $(1-\eta)$-maximal matching with probability at least $(1-\delta)$. **AMM**(G, η, δ) runs in $O(\log(\eta^{-1}\delta^{-1}))$ rounds.

Proof. Consider the algorithm which iterates **MatchingRound** s times. We apply Lemma 3.17 and Markov's inequality to obtain

$$\Pr(|V_s| \geq \eta n) \leq \frac{c^s n}{\eta n} = \eta^{-1} c^s.$$

Choosing $s = O(\log(\delta^{-1}\eta^{-1}))$, we have $\eta^{-1}c^s \leq \delta$, which gives the desired result. □

Chapter 4

Three-gender Stable Matchings

4.1 Chapter Overview

In Chapters 2 and 3, we examined two notions of almost stability and their effects computational complexity. In Chapter 2, we saw that the communication complexity of the stable marriage problem is (asymptotically) unchanged if we consider the relaxation to finding a matching μ which is close to stable in the sense that $d(\mu) \leq \varepsilon n$ for fixed $\varepsilon < 1/2$. In Chapter 3, we saw that in the distributed setting, $(1-\varepsilon)$-stable matchings can be found exponentially faster than exactly stable matchings for any $\varepsilon > 0$. In this chapter, we turn our attention to approximate versions of a three gender variant of the SMP (see Section 1.5).

Since Gale and Shapley first formalized and studied the stable marriage problem (SMP) in 1962 [11], many variants of the SMP have emerged (see, for example, [15, 29, 31, 43]). While many of these variants admit efficient algorithms, two, notably are known to be NP-hard: (1) incomplete preferences with ties [23], and (2) 3 gender stable matchings (3GSM) [34].

In the case of incomplete preferences with ties, it is NP-hard to find a maximum cardinality stable matching [23]. The intractability of exact computation for this problem led to the study of approximate versions of the problem. These investigations have resulted in hardness of approximation results [21, 47] as well as constant factor approximation algorithms [27, 28, 37, 47].

In 3GSM, agents are one of three genders: men, women, and dogs (as suggested by Knuth). Each agent holds preferences over the set of *pairs* of agents of the other two genders. The goal is to partition the agents into families, each consisting of one man, one woman, and one dog, such that no triple mutually prefer one another to their assigned families. In 1988, Alkan showed that for this natural generalization of SMP to three genders, there exist preferences which do not admit a stable matching [2]. In 1991, Ng and Hirschberg showed that, in fact, it is NP-complete to determine if given preferences admit a stable matching [34]. They further generalize this result to the three person stable assignment problem (3PSA). In 3PSA, each agent ranks all pairs of other agents without regard to gender. The goal is to partition agents into disjoint triples where again, no three agents mutually prefer each other to their assigned triples.

Despite the advances for stable matchings with incomplete preferences and ties (see [31] for an overview of relevant work), analogous approximability results have not previously been obtained for 3 gender variants of the stable marriage problem. In this chapter, we achieve the first substantial progress towards understanding the approximability of 3GSM and 3PSA. The material in this chapter original appeared in [36].

4.1.1 Overview of results

3 gender stable matchings (3GSM)

We formalize two optimization variants of 3GSM: maximally stable matching (3G-MSM) and maximum stable sub-matching (3G-MSS). For 3G-MSM, we seek a perfect (3 dimensional) matching which minimizes the number of unstable triples—triples of agents who mutually prefer each other to their assigned families. For 3G-MSS, we seek a largest cardinality *sub*-matching which contains no unstable triples among the matched agents. Exact computation of each of these problems is NP-hard by Ng and Hirschberg's result (Theorem 1.16) [34]. Indeed, exact computation of either allows one to detect the existence of a stable matching.

In this chapter, we obtain the following inapproximability result for 3G-MSM and 3G-MSS.

Theorem 4.1 (Special case of Theorem 4.9)**.** There exists an absolute constant $c < 1$ such that it is NP-hard to approximate 3G-MSM and 3G-MSS to within a factor c.

In fact, we prove a slightly stronger result for 3G-MSM and 3G-MSS. We show that the problem of determining if given preferences admit a stable matching or if all matchings are "far from stable" is NP-hard. See Section 4.2.1 and Theorem 4.9 for the precise statements. In the other direction, we describe a polynomial time constant factor approximation algorithm for 3G-MSM.

Theorem 4.2. There exists a polynomial time algorithm, **AMSM**, which computes a $\frac{4}{9}$-factor approximation to 3G-MSM.

Corollary 4.3. 3G-MSM is APX-complete.

Three person stable assignment (3PSA)

We also consider the three person stable assignment problem (3PSA). In this problem, agents rank all pairs of other agents and seek a (3 dimensional) matching—a partition of agents into disjoint triples. Notions of stability, maximally stable matching, and maximum stable submatching are defined exactly as the analogous notions for 3GSM. We show that Theorems 4.1

and 4.2 have analogues with 3PSA:

Theorem 4.4. There exists a constant $c < 1$ such that it is NP-hard to approximate 3PSA-MSM and 3PSA-MSS to within a factor c.

Theorem 4.5. There exists a polynomial time algorithm, **ASA**, which computes a $\frac{4}{9}$-factor approximation to 3PSA-MSM.

Remark 4.6. The hardness of approximation of 3PSA-MSM bears a strong resemblance to the work of Abraham, Biró, and Manlove [1], who prove a similar hardness of approximation result for the two person stable assignment problem. The authors prove that finding a matching which minimizes the number of "blocking pairs" is NP-hard, as is approximating the minimum number of blocking pairs to within an additive error of $n^{1/2-\varepsilon}$.

Our proofs of the lower bounds in Theorems 4.1 and 4.4 use a reduction from the 3 dimensional matching problem (3DM) to 3G-MSM. Kann [25] showed that Max-3DM is Max-SNP complete. Thus, by the PCP theorem [3, 4] and [7], it is NP-complete to approximate Max-3DM to within some fixed constant factor. Our hardness of approximation results then follow from a reduction from 3DM to 3G-MSM.

Theorems 4.2 and 4.5 follow from a simple greedy algorithm. Our algorithm constructs matchings by greedily finding triples whose members are guaranteed to participate in relatively few unstable triples. Thus, we are able to efficiently construct matchings with a relatively small fraction of blocking triples.

4.2 Background and Definitions

4.2.1 Three gender stable matching (3GSM)

In the 3 gender stable matching problem, there are disjoint sets of ***men***, ***women***, and ***dogs*** denoted by M, W, and D, respectively. We assume $|M| = |W| = |D| = n$, and we denote the collection of ***agents*** by $V = M \cup W \cup D$. A ***family*** is a triple (m, w, d) consisting of one man $m \in M$, one woman $w \in W$, and one dog $d \in D$. To reduce clutter, we will write $mwd = (m, w, d) \in M \times W \times D$. A ***sub-matching*** μ is a set of pairwise disjoint families. A ***matching*** μ is a maximal sub-matching—that is, one in which every agent $v \in V$ is contained in some (unique) family so that $|\mu| = n$. Given a sub-matching μ, associate the function $\mu : V \to V^2 \cup \{\varnothing\}$ which assigns each agent $v \in V$ to their partners in μ, with $\mu(v) = \varnothing$ if v is not contained in any family in μ.

Each agent $v \in V$ has a ***preference***, denoted $\prec_v$ over pairs of members of the other two genders. That is, each woman $w \in W$ holds a total order $\prec_w$ over $(M \times D) \cup \{\varnothing\}$, and similarly for men and dogs. We assume that each agent prefers being in some family to having no family. For example, $md \prec_w \varnothing$ for all $m \in M$, $w \in W$ and $d \in D$. An instance of the ***three gender stable marriage problem*** (3***GSM***) consists of M, W, and D together with preferences $\mathcal{P} = \{\prec_v \mid v \in V\}$ for all of the agents $v \in V$.

Given a sub-matching μ, a triple mwd is an ***unstable triple*** if m, w and d each prefer the triple mwd to their assigned families in μ. That is, mwd is unstable if and only if $md \prec_w \mu(w)$, $wd \prec_m \mu(m)$, and $mw \prec_d \mu(d)$. A triple mwd which is not unstable is ***stable***. In particular, mwd is stable if at least one of m, w and d prefers their family in μ to mwd. Let M_μ, W_μ and D_μ be the sets of women, men and dogs (respectively) which have families in μ. A sub-matching μ is a ***stable sub-matching*** if there are no unstable triples in $M_\mu \times W_\mu \times D_\mu$.

Unlike the two gender stable matching problem, this three gender variant arbitrary preferences need not admit a stable matching. In fact, for some preferences, every matching has many unstable triples (see Section 4.3.1). Thus we consider two optimization variants of the three gender stable matching problem.

Maximally Stable Matching (3G-MSM)

The ***maximally stable matching problem*** (3G-MSM) is to find a matching μ with the maximum number of stable triples with respect to given preferences $\mathcal{P}$. For fixed preferences $\mathcal{P}$ and matching μ, the ***stability*** of μ with respect to $\mathcal{P}$ is the number of stable triples in $M \times W \times D$:

$$\mathrm{stab}(\mu) = |\{mwd \mid mwd \text{ is stable}\}| \, .$$

Thus, μ is stable if and only if $\mathrm{stab}(\mu) = n^3$. Dually, we define the ***instability*** of μ by $\mathrm{ins}(\mu) = n^3 - \mathrm{stab}(\mu)$. For fixed preferences $\mathcal{P}$, we define

$$\mathrm{MSM}(\mathcal{P}) = \max \{\mathrm{stab}(\mu) \mid \mu \text{ is a matching}\} \, .$$

For preferences $\mathcal{P}$ and fixed $c < 1$, we define ***Gap***$_c$-3***G-MSM*** to be the problem of determining if $\mathrm{MSM}(P) = n^3$ or $\mathrm{MSM}(P) \leq cn^3$.

Maximum Stable Sub-Matching (TGMSS)

The ***maximum stable sub-matching problem*** (3G-MSS) is to find a maximum cardinality stable sub-matching μ. We denote

$$\mathrm{MSS}(\mathcal{P}) = \max \{|\mu| \mid \mu \text{ is a stable sub-matching}\}$$

Note that $\mathcal{P}$ admits a stable matching if and only if $\mathrm{MSS}(\mathcal{P}) = n$. For fixed $c < 1$, we define ***Gap***$_c$-3***G-MSS*** to be the problem of determining if $\mathrm{MSS}(\mathcal{P}) = n$ or if $\mathrm{MSS}(\mathcal{P}) \leq cn$.

4.2.2 Three person stable assignment (3PSA)

In the ***three person stable assignment problem*** (3***PSA***), there is a set U of $|U| = 3n$ agents who wish to be partitioned into n disjoint triples. For a set $C \subseteq U$, we denote the set of k-subsets of C by $\binom{C}{k}$. A ***sub-matching*** is a set $S \subseteq \binom{U}{3}$ of disjoint triples in U. A ***matching*** μ is a maximal sub-matching—a sub-matching with $|\mu| = n$. Given a sub-matching μ, U_μ is the set of agents contained in some triple in μ:

$$U_\mu = \{u \in U \mid u \in t \text{ for some } t \in S\}.$$

Each agent $u \in U$ holds preferences among all pairs of potential partners. That is, each $u \in U$ holds a linear order $\prec_u$ on $\binom{U\setminus\{u\}}{2} \cup \{\varnothing\}$. We assume that each agent prefers every pair to an empty assignment. Given a set $\mathcal{P}$ of preferences for all the agents and a sub-matching μ, we call a triple $uvw \in \binom{U_\mu}{3}$ ***unstable*** if each of u, v and w prefer the triple uvw to their assigned triples in μ. Otherwise, we call uvw ***stable***. A sub-matching μ is ***stable*** if it contains no unstable triples in $\binom{U_\mu}{3}$. We define the ***stability*** of μ by

$$\operatorname{stab}(\mu) = \left| \left\{ uvw \in \binom{U_\mu}{3} \;\middle|\; uvw \text{ is stable} \right\} \right|.$$

Dually, the ***instability*** of μ is $\operatorname{ins}(\mu) = \binom{|\mu|}{3} - \operatorname{stab}(\mu)$.

The ***maximally stable matching problem*** (3***PSA-MSM***) is to find a matching μ which maximizes $\operatorname{stab}(\mu)$. The ***maximum stable sub-matching problem*** (3***PSA-MSS***) is to find a stable sub-matching μ of maximum cardinality.

Remark 4.7. We may consider a variant of 3PSA with ***unacceptable partners***. In this variant, each agent $u \in U$ ranks only a subset of $\binom{U\setminus\{u\}}{2}$, and prefers being unmatched to unranked pairs. 3GSM is a special case of this variant where $U = M \cup W \cup D$ and each agent ranks precisely those pairs consisting of one agent of each other gender. This observation will make our hardness results for 3GSM easily generalize to 3PSA.

4.2.3 Hardness of Gap$_c$-3DM-3

Our proofs of Theorems 4.1 and 4.4 use a reduction from the three dimensional matching problem (3DM). In this section, we briefly review 3DM, and state the approximability result we require for our lower bound results.

Let X, Y, Z be finite disjoint sets with $|X| = |Y| = |Z| = m$. Let $E \subseteq X \times Y \times Z$ be a set of edges. A ***matching*** $\mu \subseteq E$ is a set of disjoint edges. The ***maximum*** 3 ***dimensional matching problem*** (***Max-3DM***) is to find (the size of) a matching μ of largest cardinality in E. ***Max-3DM-3*** is the restriction of Max-3DM to instances where each element in $X \cup Y \cup Z$ is contained in at most 3 edges. For a fixed constant $c < 1$, we define ***Gap$_c$-3DM-3*** to be the problem of determining if an instance I of Max-3DM-3 has a perfect matching (a matching μ of size m) or if every matching has size at most cm.

Theorem 4.8. There exists an absolute constant $c < 1$ such that Gap$_c$-3DM-3 is NP-hard.

Kann showed that Max-3DM-3 is Max-SNP complete[1] by giving an L-reduction from Max-3SAT-B to Max-3DM-3 [25]. By the celebrated PCP theorem [3, 4] and [7], Kann's result immediately implies that Max-3DM-3 is NP-hard to approximate to within some fixed constant factor. However, Kann's reduction gives a slightly weaker result than Theorem 4.8. In Kann's reduction, satisfiable instances of 3SAT-B do not necessarily reduce to instances of 3DM-3 which admit perfect matchings. In Section 4.6, we describe a straightforward alteration of Kann's reduction such that satisfiable instances of 3SAT-B admit perfect matchings, while far-from-satisfiable instances are still far from admitting perfect matchings.

[1]The complexity class Max-SNP was introduced by Papadimitriou and Yannakakis in [38], where the authors also show that Max-3SAT-B is Max-SNP complete.

4.3 Hardness of Approximation

In this section, we prove the main hardness of approximation results. Specifically, we will prove the following theorems.

Theorem 4.9. There exists and absolute constant $c < 1$ such that Gap$_c$-3G-MSM and Gap$_c$-3G-MSS are NP-hard.

Theorem 4.10. There exists an absolute constant $c < 1$ such that Gap$_c$-3PSA-MSM and Gap$_c$-3PSA-MSS are NP-hard.

4.3.1 Preferences for 3GSM with many unstable triples

In this section, we construct preferences $\mathcal{P}$ for 3GSM such that any matching μ induce many unstable triples with respect to $\mathcal{P}$.

Theorem 4.11. There exist preferences $\mathcal{P}$ for 3GSM and a constant $c < 1$ for which $\mathrm{MSM}(P) \leq cn^3$.

We first consider the case where $n = 2$. We denote $M = \{m_1, m_2\}$, $W = \{w_1, w_2\}$, and $D = \{d_1, d_2\}$. Consider preference lists $\mathcal{P}$ as described in the following table, where most preferred partners are listed first.

agent	preferences		
m_1	w_1d_1	w_2d_2	$\cdots$
m_2	w_2d_1	$\cdots$	
w_1	m_1d_1	$\cdots$	
w_2	m_1d_2	m_2d_1	$\cdots$
d_1	m_2m_2	m_1w_1	$\cdots$
d_2	m_1m_2	$\cdots$	

The ellipses indicate that the remaining preferences are otherwise arbitrary. Suppose μ is a stable matching for $\mathcal{P}$. We must have either $m_1w_1d_1 \in M$ or $m_1w_2d_2 \in M$, for otherwise the triple $m_1w_2d_2$ is unstable. However, if $m_1w_1d_1 \in M$, then $m_2w_2d_1$ is unstable. On the other hand, if $m_1w_2d_2 \in M$ then $m_1w_1d_1$ is unstable. Therefore, no such stable μ exists. In particular, every matching μ contains at least one unstable triple.

The idea of the proof of Theorem 4.11 is to choose preferences $\mathcal{P}$ such that, when restricted to many sets of two women, two men and two dogs, the relative preferences are as above. Thus any matching containing families consisting of these agents must induce unstable triples.

agent	preferences		
$m_1 \in M_1$	W_1D_1	W_2D_2	$\cdots$
$m_2 \in M_2$	W_2D_1	$\cdots$	
$w_1 \in W_1$	M_1D_1	$\cdots$	
$w_2 \in W_2$	M_1D_2	M_2D_1	$\cdots$
$d_1 \in D_1$	M_2W_2	M_1W_1	$\cdots$
$d_2 \in D_2$	M_1W_2	$\cdots$	

Figure 4.1: Preferences $\mathcal{P}$ inducing many blocking triples. Assuming n is even, we partition each gender into two equal sized sets $M = M_1 \cup M_2$, $W = W_1 \cup W_2$, and $D = D_1 \cup D_2$. The sets appearing in the preferences indicate that the agent prefers all pairs in that set (in any order) followed by the remaining preferences. For example, all $m_1 \in M_1$ prefer all partners $wd \in W_1 \times D_1$, followed by all partners in $W_2 \times D_2$, followed by all other pairs in arbitrary order. Within $W_1 \times D_1$ and $W_2 \times D_2$, m_1's preferences are arbitrary.

Proof of Theorem 4.11. We partition the sets M, W and D each into two sets of equal size: $M = M_1 \cup M_2$, $W = W_1 \cup W_2$, $D = D_1 \cup D_2$. Consider the preferences $\mathcal{P}$ described in Figure 4.1. We will prove that for $\mathcal{P}$, every matching μ contains at least $n^3/128$ unstable triples. Let μ be an arbitrary matching, and suppose $\operatorname{ins}(\mu) < n^3/128$. We consider two cases separately.

Case 1: $|\mu \cap (M_1 \times W_1 \times D_1)| \leq n/4$**.** Let M_1', W_1' and D_1' be the subsets of M_1, W_1 and D_1 respectively of agents not in triples contained in $M_1 \times W_1 \times D_1$. By the hypothesis, $|M_1'|, |W_1'|, |D_1'| \geq n/4$. Let $d_1 \in D_1'$. Notice that if $\mu(d_1) \notin M_2 \times W_2$, then $m_1 m_1 d_1$ is unstable for all $m_1 \in M_1'$, $w_2 \in W_1'$. Since fewer than $n^3/128$ triples in $M_1' \times W_1' \times D_1'$ are unstable, at least $3n/8$ dogs $d_1 \in D_1'$ must have families $m_2 w_2 d_1 \in M_2 \times W_2 \times D_1'$.

Since $|\mu \cap (M_2 \times W_2 \times D_1)| \geq 3n/8$, we must have $|\mu \cap (M_1 \times W_2 \times D_2)| \leq n/8$. Thus, there must be at least $n/8$ men $m_1 \in M_1$ with partners not in $(W_1 \times D_1) \cup (W_2 \times D_2)$. However, every such m_1 forms an unstable triple with every $w_2 \in W_2$ and $d_2 \in D_2$ which are not in families in $M_1 \times W_2 \times D_2$. Since there at least $3n/8$ such m_2 and d_2, there are at least

$$\left(\frac{n}{8}\right)\left(\frac{3n}{8}\right)\left(\frac{3n}{8}\right) > \frac{n}{128}$$

blocking triples, a contradiction.

Case 2: $|\mu \cap (M_1 \times W_1 \times D_1)| > n/4$**.** In this case, we must have $|\mu \cap (M_2 \times W_2 \times D_1)| < n/4$. This implies that

$$|\mu \cap (M_1 \times W_2 \times D_2)| > 3n/8 \tag{4.1}$$

for otherwise triples $m_2 w_2 d_1 \in (M_2 \times W_2 \times D_1)$ with $\mu(m_2) \notin W_1 \times D_1$ form more than $n^3/128$ unstable triples. But (4.1) contradicts the Case 2 hypothesis, as $|M_1| = n/2$.

Since both cases lead to a contradiction, we may conclude that any μ contains at least $n^3/128$ unstable triples, as desired. □

4.3.2 The embedding

We now describe an embedding of 3DM-3 into 3G-MSM. Our embedding is a modification of the embedding described by Ng and Hirschberg [34]. Let I be an instance of 3DM-3 with ground sets X, Y, Z and edge set E. We assume $|X| = |Y| = |Z| = m$. We will construct an instance $f(I)$ of 3G-MSM with sets M, W and D of women, men and dogs of size $n = 6m$ and suitable preferences $\mathcal{P}$. We divide each gender into two sets $M = M^1 \cup M^2$, $W = W^1 \cup W^2$ and $D = D^1 \cup D^2$ where $|M^j| = |W^j| = |D^j| = 3m$ for $j = 1, 2$. Let $X = \{m_1, m_2, \ldots, m_m\}, Y = \{w_1, w_2, \ldots, w_m\}$ and $Z = \{d_1, d_2, \ldots, d_m\}$, and denote

$$E = \bigcup_{i=1}^{n} \{m_i w_{k_1} d_{\ell_1},\ m_i w_{k_2} d_{\ell_2},\ m_i w_{k_3} d_{\ell_3}\}\,.$$

Without loss of generality, we assume each m_i is contained in exactly 3 edges by possibly increasing the multiplicity of edges containing m_i. The idea of the embedding $f(I)$ is that each $m_i \in M$ is mapped to 6 agents $m_i^1[1], m_i^1[2], m_i^1[3] \in M^1$ and $m_i^2[1], m_i^2[2], m_i^2[3] \in M^2$. The three agents in M^1 and M^2 correspond to the three edges in E which contain m_i. We choose preferences in such a way that at most one family from $m_i^1[1]w_k^1 d_\ell^1$, $m_i^1[2]w_k^1 d_\ell^1$, and $m_i^1[3]w_k^1 d_\ell^1$ can be in any (sub)matching, where $m_i w_k d_\ell \in E$. Such a family corresponds to a choice of an edge containing m_i to include in a maximal matching. We then show that if I admits a perfect matching, then $f(I)$ admits a stable matching. On the other hand, if I is far from admitting a perfect matching, then our choice of preferences ensure that any matching induces many unstable triples by appealing to Theorem 4.11.

For $j = 1, 2$, we form the sets

$$M^j = \left\{m_i^j[k] \;\middle|\; i \in [n],\ k \in [3]\right\},$$

$$W^j = \left\{w_i^j,\ x_i^j,\ y_i^j \;\middle|\; i \in [n]\right\}$$

$$D^j = \left\{d_i^j,\ z_i^j,\ t_i^j \;\middle|\; i \in [n]\right\}.$$

We now define preferences for each set of agents, beginning with those in M.

$$\begin{array}{c|llllll} m_i^1[a] & x_i^1 z_i^1 & y_i^1 t_i^1 & w_{k_a}^1 d_{\ell_a}^1 & W^1 D^1 & W^2 D^2 & \cdots \\ m_i^2[a] & x_i^2 z_i^2 & y_i^2 t_i^2 & w_{k_a}^2 d_{\ell_a}^2 & W^2 D^1 & \cdots & \end{array}$$

The agents in W have preferences given by

$$\begin{array}{c|llllll} w_i^1 & m_i^1[1] z_i^1 & m_i^1[2] z_i^1 & m_i^1[3] z_i^1 & M^1 D^1 & \cdots & \\ x_i^1 & m_i^1[1] z_i^1 & m_i^1[2] z_i^1 & m_i^1[3] z_i^1 & M^1 D^1 & \cdots & \\ y_i^1 & M^1 D^1 & \cdots & & & & \\ w_i^2 & m_i^2[1] z_i^2 & m_i^2[2] z_i^2 & m_i^2[3] z_i^2 & M^1 D^2 & M^2 D^1 & \cdots \\ x_i^2 & m_i^2[1] t_i^2 & m_i^2[2] t_i^2 & m_i^2[3] t_i^2 & M^1 D^2 & M^2 D^1 & \cdots \\ y_i^2 & M^1 D^2 & M^2 D^1 & \cdots & & & \end{array}$$

The preferences for D are given by

$$\begin{array}{c|llllll} z_i^1 & m_i^1[3] x_i^1 & m_i^1[2] x_i^1 & m_i^1[1] x_i^1 & M^2 W^2 & M^1 W^1 & \cdots \\ t_i^1 & m_i^1[3] y_i^1 & m_i^1[2] y_i^1 & m_i^1[1] y_i^1 & M^2 W^2 & M^1 W^1 & \cdots \\ d_i^1 & M^2 W^2 & M^1 W^1 & \cdots & & & \\ z_i^2 & m_i^2[3] x_i^2 & m_i^2[2] x_i^2 & m_i^2[1] x_i^2 & M^1 W^2 & \cdots & \\ t_i^2 & m_i^2[3] y_i^2 & m_i^2[2] y_i^2 & m_i^2[1] y_i^2 & M^1 W^2 & \cdots & \\ d_i^2 & M^1 W^2 & \cdots & & & & \end{array}$$

The sets M^j, W^j and D^j in the preferences described above indicate that all agents in these sets appear consecutively in some arbitrary order in the preferences. Ellipses indicate that all remaining preferences may be completed arbitrarily. For example, $m_1^1[1]$ most prefers $x_1^1 z_1^1$, followed by $y_1^1 t_1^1$ and $w_{k_m}^1 d_{\ell_m}^1$. He then prefers all remaining pairs in $W^1 D^1$ in any order, followed by all pairs in $W^2 D^2$, followed by the remaining pairs in any order.

Lemma 4.12. The embedding $f : \text{3DM-3} \longrightarrow \text{3GSM}$ described above satisfies

1. If $\text{opt}(I) = m$—that is, I admits a perfect matching—then $f(I)$ admits a stable matching (i.e. $\text{MSM}(P) = n^3$).

2. If $\text{opt}(I) \le cm$ for some $c < 1$, then there exists a constant $c' < 1$ depending only on c such that $\text{MSM}(P) \le c'n^3$.

Proof. To prove the first claim assume, without loss of generality, that $\mu' = \{m_i w_{k_1} d_{\ell_1} \mid i \in [n]\}$ is a perfect matching in E. It is easy to verify the matching

$$M = \left\{m_i^j[1] w_{k_1}^j d_{\ell_1}^j\right\} \cup \left\{m_i^j[2] x_i^j z_i^j\right\} \cup \left\{m_i^j[3] y_i^j t_i^j\right\}$$

contains no blocking triples, hence is a stable matching.

For the second claim, let μ be an arbitrary matching in $M \times W \times D$. We observe that there are at least $(1-c)m$ agents $m^1 \in M^1$ and $(1-c)m$ agents $m^2 \in M^2$ that are not matched with pairs from their top three choices. Suppose to the contrary that $\alpha > (2+c)m$ agents $m^1 \in M^1$ are matched with their top 3 choices. This implies that more than cm women $m^1 \in M^1$ are matched in triples of the form $m^1 w_k^1 d_\ell^1$ with $m w_k d_\ell \in E$, implying that E contains a matching of size $\alpha - 2m > cm$, a contradiction. Thus at least $2(1-c)m$ men in $M^1 \cup M^2$ are matched below their top three choices.

Let M' denote the set of men matched below their top three choices, and W' and D' the sets of partners of $m \in M'$ in μ. By the previous paragraph, $|M'| \ge 2(1-c)m = (1-c)m/6$. Further, the relative preferences of agents in M', W' and D' are precisely those described in Theorem 4.11. Thus, by Theorem 4.11, any matching μ among these agents induces at least $|M'|^3/128$ blocking triples. Hence μ must contain at least

$$\frac{|M'|}{128} \ge \frac{(1-c)^3}{3456} n^3$$

blocking triples. $\square$

Proof of Theorem 4.9. The reduction $f : \text{3DM-3} \longrightarrow \text{3GSM}$ is easily seen to be polynomial time computable. Thus, by Lemma 4.12, f is a polynomial time reduction from Gap$_c$-3DM-3 to Gap$_{c'}$-3G-MSM where $c' = 1 - (1-c)^3/3456$. The NP hardness of Gap$_c$-3G-MSM is then an immediate consequence of Theorem 4.8.

The hardness of Gap$_c$-3G-MSS is a consequence of the hardness Gap$_c$-3G-MSM. Consider an instance of 3GSM with preferences $\mathcal{P}$. We make the following observations.

1. $\text{MSM}(\mathcal{P}) = n^3$ if and only if $\text{MSS}(\mathcal{P}) = n$.

2. If $\text{MSM}(\mathcal{P}) \leq (1-3\varepsilon)n^3$ for $\varepsilon > 0$, then $\text{MSS}(\mathcal{P}) \leq (1-\varepsilon)n$.

The first observation is clear. To prove the second, suppose that $\text{MSS}(\mathcal{P}) > (1-\varepsilon)n$, and let μ be a maximum stable sub-matching. We can form a matching μ by arbitrarily adding εn disjoint families to μ. Since each new family can induce at most $3n^2$ blocking triples, μ has at most $3\varepsilon n^3$ blocking triples, hence $\text{MSM}(\mathcal{P}) > (1-3\varepsilon)n^3$. The two observations above imply that any decider for Gap$_{(1-\varepsilon)}$-3G-MSS is also a decider for Gap$_{(1-3\varepsilon)}$-3G-MSM. Thus, the NP-hardness of Gap$_c$-3G-MSM immediately implies the analogous result for Gap$_c$-3G-MSS. $\square$

Here we sketch a proof of the analogous lower bounds for 3PSA given in Theorem 4.10.

Proof sketch of Theorem 4.10. As noted in Remark 4.7, we may view 3GSM as a special case of 3PSA with incomplete preferences. The NP-hardness of approximation of 3PSA with incomplete preferences is analogous to the proof of Theorem 4.9. Given an instance I of 3GSM with sets M, W, and D and preferences $\mathcal{P}$, take $U = M \cup W \cup D$ and form 3PSA preferences $\mathcal{P}'$ by appending the remaining pairs to $\mathcal{P}$ arbitrarily. Analogues of Theorem

4.11 and Lemma 4.12 hold for this instance of 3PSA, whence Theorem 4.10 follows. We leave details to the reader. □

4.4 Approximation Algorithms

4.4.1 3GSM approximation

In this section, we describe a polynomial time approximation algorithm for MSM, thereby proving Theorem 4.2. Consider an instance of 3GSM with preferences $\mathcal{P}$, and as before $M = \{m_1, m_2, \ldots, m_n\}$, $W = \{w_1, w_2, \ldots, w_n\}$, and $D = \{d_1, d_2, \ldots, d_n\}$. Given a triple $m_i w_j d_k$, we define its ***stable set*** S_{ijk} to be the set of (indices of) triples which cannot form unstable triples with $m_i w_j d_k$. Specifically, we have

$$S_{ijk} = \left\{\alpha\beta\delta \in [n]^3 \;\middle|\; w_\beta d_\delta \preceq_{m_i} m_j d_k,\ \alpha = i\right\} \cup \left\{\alpha\beta\delta \in [n]^3 \;\middle|\; w_\alpha d_\delta \preceq_{m_j} m_i d_k,\ \beta = j\right\}$$
$$\cup \left\{\alpha\beta\delta \in [n]^3 \;\middle|\; m_\alpha w_\beta \preceq_{d_k} m_i w_j,\ \delta = k\right\}$$

The idea of our algorithm is to greedily form families that maximize $|S_{ijk}|$. Pseudocode is given in Algorithm 6.

Algorithm 6 AMSM$(M, W, D, \mathcal{P})$

```
find ijk ∈ [n]^3 which maximize |S_ijk|
M' ← M \ {m_i}, W' ← W \ {w_j}, D' ← D \ {d_k}
P' ← P restricted to M', W', and D'
return {m_i w_j d_k} ∪ AMSM(M', W', D', P')
```

It is easy to see that **AMSM** can be implemented in polynomial time. The naive algorithm for computing $|S_{ijk}|$ for fixed $ijk \in [n]^3$ by iterating through all triples $\alpha\beta\delta \in [n]^3$ and querying each agent's preferences can be implemented in time $\tilde{O}(n^3)$. The maximal such $|S_{ijk}|$ can then be found by iterating through all $ijk \in [n]^3$. Thus the first step in **AMSM** can be accomplished in time $\tilde{O}(n^6)$. Finally, the recursive step of **AMSM** terminates after n iterations, as each iteration decreases the size of M, W, and D by one.

Lemma 4.13. For any preferences $\mathcal{P}$, and sets M, W and D with $|M| = |W| = |D| = n$, there exists a triple $ijk \in [n]^3$ with

$$|S_{ijk}| \geq \frac{4n^2}{3} - n - 1. \tag{4.2}$$

Proof. We will show that there exists a triple $m_i w_j d_k$ such that at least two of m_i, w_j, and d_k respectively rank $w_j d_k$, $m_i d_k$, and $m_i w_j$ among their top $n^2/3 + 1$ choices. Note that this occurs precisely when at least two of the the following inequalities are satisfied

$$\left|\left\{\beta\delta \in [n]^2 \;\middle|\; w_\beta d_\delta \preceq_{m_i} w_j d_k\right\}\right| \leq \frac{n^2}{3} + 1, \quad \left|\left\{\alpha\delta \in [n]^2 \;\middle|\; m_\alpha d_\delta \preceq_{w_j} m_i d_k\right\}\right| \leq \frac{n^2}{3} + 1,$$

$$\text{and} \quad \left|\left\{\alpha\beta \in [n]^2 \;\middle|\; m_\alpha w_\beta \preceq_{d_k} m_i w_j\right\}\right| \leq \frac{n^2}{3} + 1$$

Mark each triple $m_i w_j d_k$ which satisfies one of the above inequalities. Each m_i induces $\frac{n^2}{3} + 1$ marks, so we get $\frac{n^3}{3} + n$ marks from all $m \in M$. Similarly, we get $\frac{n^3}{3} + n$ marks from W and D. Thus, marks are placed on at least $n^3 + 3n$ triples. By the pigeonhole principle, at least one triple is marked twice.

We claim that the triple $m_i w_j d_k$ satisfying two of the above inequalities satisfies equation (4.2). Without loss of generality, assume that $m_i w_j d_k$ satisfies the first two equations. Thus, m_i and w_j must each contribute at least $\frac{2n^3}{3} - 1$ stable triples with respect to $m_i w_j d_k$. Further, at most $n - 1$ such triples can be contributed by both m_i and w_j, as such triples must be of the form $m_i w_j d_\delta$ for $\delta \neq k$. Thus (4.2) is satisfied, as desired. $\square$

We are now ready to prove that **AMSM** gives a constant factor approximation for the maximally stable matching problem.

Proof of Theorem 4.2. Let μ be the matching found by **AMSM**, and suppose

$$M = \{m_1w_1d_1,\ m_2w_2d_2,\ \ldots,\ m_nw_nd_n\}$$

where $m_1w_1d_1$ is the first triple found by **AMSM**, $m_2w_2d_2$ is the second, et cetera. By Lemma 4.13, $|S_{111}| \geq \frac{4}{3}n^2 - O(n)$. Therefore, the agents m_1, w_1, and d_1 can be contained in at most $\frac{5}{3}n^2 + O(n)$ unstable triples in any matching containing the family $m_1w_1d_1$. Similarly, for $1 \leq i \leq n$ the ith family $m_iw_id_i$ can contribute at most $\frac{5}{3}(n-i+1)^2 + O(n)$ new unstable triples (not containing any m_j, w_j, or d_j for $j < i$). Thus, the total number of unstable triples in μ is at most

$$\sum_{i=1}^{n}\left(\frac{5}{3}(n-i+1)^2 + O(n)\right) = \frac{5}{9}n^3 + O(n^2).$$

Thus, we have $\mathrm{stab}(\mu) \geq 4n^3/9 - O(n^2)$ as desired. $\square$

4.4.2 3PSA approximation

AMSM can easily be adapted for 3PSA. Let U be a set of agents with $|U| = 3n$, and let $\mathcal{P}$ be a set of complete preferences for the agents in U. Given a triple $abc \in \binom{U}{3}$, we form the ***stable set*** S_{abc} consisting of triples that at least one of a, w, c does not prefer to abc. The approximation algorithm **ASA** for 3PSA is analogous to **AMSM**: form a matching μ by finding a triple abc that maximizes $|S_{abc}|$, then recursing. The following lemma and its proof are analogous to Lemma 4.13.

Lemma 4.14. For any set U of agents with $|U| = 3n$ and complete preferences $\mathcal{P}$, there exists a triples $abc \in \binom{U}{3}$ such that

$$|S_{abc}| \geq 6n^2 - O(n).$$

Using Lemma 4.14, we prove Theorem 4.5 analogously to Theorem 4.2.

Proof of Theorem 4.5. Each triple abc can intersect at most $3\binom{3n}{2} \leq \frac{27}{2}n^2$ blocking triples. Thus, by Lemma 4.14, the total number blocking triples in the matching μ found by **ASA** is at most

$$\sum_{i=0}^{n-1} \left(\frac{15}{2}(n-i+1)^2 + O(n) \right) = \frac{5}{2}n^3 + O(n^2).$$

Therefore,

$$\text{stab}(\mu) \geq 2n^3 - O(n^2),$$

as the total number of triples in $\binom{U}{3}$ is $\frac{9}{2}n^3 - O(n^2)$. Hence μ is a $\frac{4}{9}$-approximation to a maximally stable matching, as desired. □

4.5 Concluding Remarks and Open Questions

While **AMSM** gives a simple approximation algorithm for 3G-MSM, we do not generalize this result to 3G-MSS. Indeed, even the first two families output by **AMSM** may include blocking triples. We leave the existence of an efficient approximation for 3G-MSS as a tantalizing open question.

Open Question 4.15. Is it possible to efficiently compute a constant factor approximation to 3G-MSS?

Finding an approximation algorithm for maximally stable matching was made easier by the fact that any preferences admit a matching with $\Omega(n^3)$ stable triples. However, for 3G-MSS, it is not clear whether every preference structure admits stable sub-matchings of size $\Omega(n)$. We feel that understanding the approximability of 3G-MSS is a very intriguing avenue of further exploration.

Open Question 4.16. How small can a maximum stable sub-matching be? What preferences achieve this bound?

In our hardness of approximation results (Theorems 4.9 and 4.10), we do not state explicit values of c for which Gap$_c$-3G-MSM and Gap$_c$-3G-MSS (and the corresponding problems for three person stable assignment) are NP-complete. The value implied by our embedding of 3SAT-B via 3DM-3 is quite close to 1. It would be interesting to find a better (explicit) factor for hardness of approximation. Conversely, is it possible to efficiently achieve a better than $4/9$-factor approximation for maximally stable matching/matching problems?

Open Question 4.17. For the maximally stable matching problems, close the gap between the $4/9$-factor approximation algorithm and the $(1-\varepsilon)$-factor hardness of approximation.

Notice that in the preference structure described in the proof of Theorem 4.11 (upon which our hardness of approximation results rely), for any $i \neq j$, w_i^2 prefers $m_i^2[1]z_i^2$ to $m_i^1[1]z_i^2$, but prefers $m_i^1[1]z_j^2$ to $m_i^2[1]z_j^2$. Thus, depending on the the second agent (z_j^2 or z_i^2), w_i^2 does not consistently prefer pairs involving $m_i^1[1]$ to $m_i^2[1]$ or vice versa. Ng and Hirschberg call such preferences as ***inconsistent***, and asked whether consistent preferences always admit a (3 gender) stable matching. Huang [19] showed that consistent preferences need not admit three gender stable matchings, and indeed it is still NP-complete to determine whether or not given consistent preferences admit a stable matching.

Open Question 4.18. Are MSM and MSS still hard to approximate if preferences are restricted to be consistent?

4.6 Hardness of Gap$_c$-3DM-3

The goal of this section is to prove Theorem 4.8, the NP hardness of Gap$_c$-3DM-3. In the first subsection, we review Kann's reduction from Max-3SAT-B to Max-$3DM$-3. In the following subsection, we describe how to modify Kann's reduction in order to obtain Theorem

4.8.

4.6.1 Kann's reduction

Let I be an instance of 3SAT-B. Specifically, I consists of a set U of n Boolean variables,

$$U = \{u_1, u_2, \ldots, u_n\},$$

and a set C of m clauses

$$C = \{c_1, c_2, \ldots, c_m\}.$$

Each clause is a disjunction of at most 3 literals, and each variable u_i or its negation $\bar{u}_i$ appears in at most B clauses. For each $i \in [n]$, let d_i denote the number of clauses in which u_i or $\bar{u}_i$ appears, where $d_i \leq B$. Kann's construction begins with the classical reduction from 3SAT to 3DM used to show the NP-completeness of 3DM, as described in [12]. Each variable u_i gets mapped to a ***ring*** of $2d_i$ edges. The points of the ring correspond alternatingly to u_i and $\bar{u}_i$. A maximal matching on the ring corresponds to a choice a truth value of u_i: if the edges containing the vertices labeled u_i are in the matching, this corresponds to u_i having the value true; if the edges containing $\bar{u}_i$ are chosen, this corresponds to u_i taking the value false. See Figure 4.2.

In the classical construction, the points of the ring corresponding to u_i are connected to ***clause vertices*** via ***clause edges*** which encode the clauses in C. This is done in such a way that the formula I is satisfiable if and only if the corresponding matching problem admits a perfect matching. The problem with this embedding, however, is that even if a relatively small fraction of clauses in C can be simultaneously satisfied, the corresponding matching problem may still admit a nearly perfect matching.

To remedy this problem, Kann's reduction maps each Boolean variable u_i to many rings. The rings are then connected via a tree structure whose roots correspond to instances of

u_i in the clauses of C. This tree structure imposes a predictable structure on the maximal matchings.

We denote the parameter

$$K = 2^{\lfloor \log(3B/2+1) \rfloor}$$

which is the number of rings to which each variable u_i maps. We denote the "free elements"—the points on the rings—associate to u_i by the variables

$$v_i[\gamma, k] \text{ and } \bar{v}_i[\gamma, k] \quad \text{for} \quad 1 \leq \gamma \leq d_i, \ 1 \leq k \leq K.$$

These vertices are connected to rings as in Figure 4.2. The rings are connected via ***tree edges*** in $2d_i$ binary trees, such that for each fixed γ, $v_i[\gamma, 1], v_i[\gamma, 2], \ldots, v_i[\gamma, K]$ are the leaves of a tree, and similarly for the $\bar{v}_i[\gamma, k]$. We label the root of this tree by $u_i[\gamma]$ or $\bar{u}_i[\gamma]$ depending on the labels of the leaves. See Figure 4.3. We refer to the resulting structure for u_i as the ***ring of trees*** corresponding to u_i.

The root vertices are connected via ***clause edges*** to ***clause vertices***. For each $c_j \in C$, we associate two clause vertices $s_1[j]$ and $s_2[j]$. If c_j is u_i's γ-th clause in C, then we include the edge

$$\{u_i[\gamma], s_1[j], s_2[j]\} \quad \text{or} \quad \{\bar{u}_i[\gamma], s_1[j], s_2[j]\}$$

depending if u_i or its negation appears in c_j and the parity of the of the tree of rings. We denote the resulting instance of 3DM by $f(I)$. It is readily apparent from this construction that $f(I)$ is in fact an instance of 3DM-3: all vertices in the rings of trees are contained in exactly 2 edges, while clause vertices are contained in at most 3 edges. Further, the vertex set V can be partitioned into a disjoint union $W \cup X \cup Y$ such that each edge contains one vertex from each of these sets. Kann makes the following observations about the structure of optimal (maximum) matchings in $f(I)$.

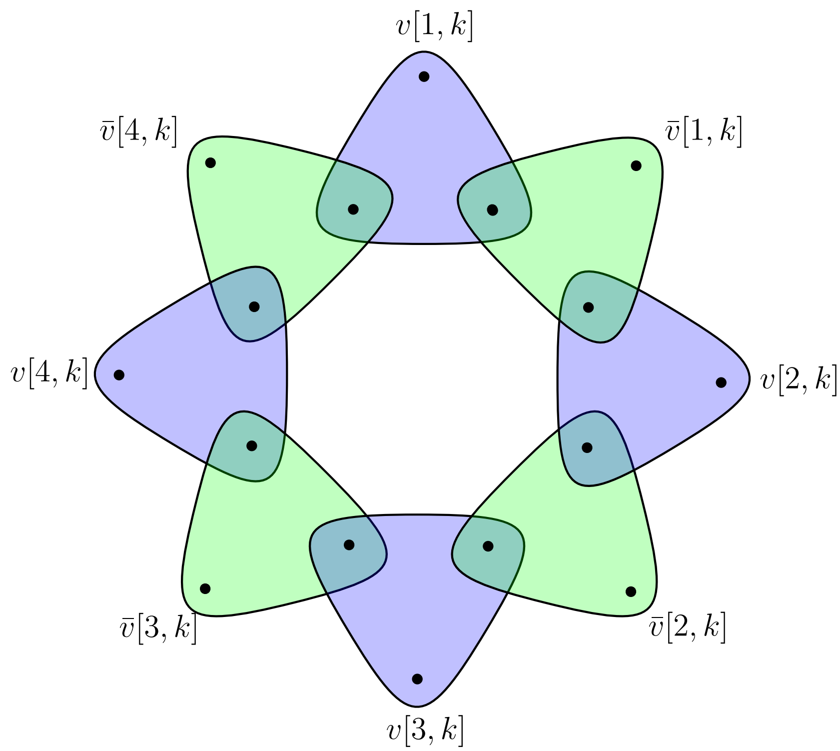

Figure 4.2: The ***ring*** structure for the embedding of 3SAT-B into 3DM-3. The ring shown corresponds to a variable u_i with $d_i = 4$. An optimal matching in the ring corresponds to a truth value of for the variable u_i: the blue edges correspond to the value *true* while the green edges correspond to the value *false*.

Lemma 4.19. Let I be an instance of 3SAT-B. Let $f(I)$ be an instance of 3DM-3 constructed as above. Then each optimal matching μ in $f(I)$ is associated with an optimal assignment in I, and has the following structure.

1. For each variable u_i, μ contains either all edges containing $v_i[\gamma, k]$ or all the edges containing $\bar{v}_i[\gamma, k]$, depending on the value u_i in the optimal assignment for I.

2. From each ring of trees, alternating tree edges are included in μ so as to cover all tree (and ring) vertices, except possibly root vertices.

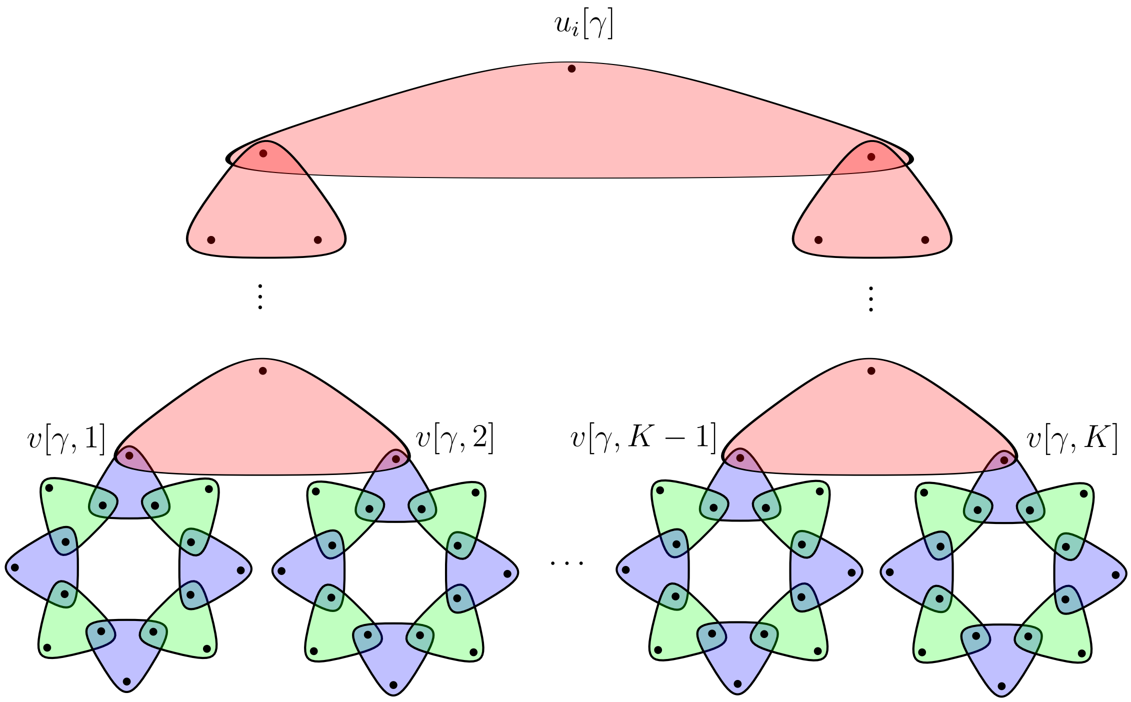

Figure 4.3: The ***ring of trees*** structure for Kann's embedding of 3SAT-B into 3DM-3. The red edges are ***tree edges***. In addition to the tree shown, the ring of trees contains (identical) trees for each $u_i[\gamma]$ and $\bar{u}_i[\gamma]$. In the example pictured, γ ranges from 1 to 4. The ***root vertices*** are labeled $u_i[1], \bar{u}_i[1], \ldots, u_i[d_i], \bar{u}_i[d_i]$ where d_i is the number of occurrences of u_i or its negation in I. As described in Lemma 4.19, an optimal matching in the ring of trees can be obtained by a consistent choice of all blue or green vertices in all the rings associated to u_i, then covering as many remaining vertices as possible with tree edges (a greedy "leaf to root" approach works). It is clear that such a matching will cover all vertices, except for half of the root vertices (those corresponding to either u_i or $\bar{u}_i$).

3. If c_j is satisfied in the optimal assignment in I, then μ contains an edge containing $s_1[j]$ and $s_2[j]$.

4. If c_j is unsatisfied in the optimal assignment in I, then none of the edges containing $s_1[j]$ and $s_2[j]$ are contained in μ.

In particular, the only possible vertices left uncovered in an optimal matching are clause vertices corresponding to unsatisfied clauses and root vertices.

As a consequence of Kann's analysis of the optimal matchings in $f(I)$, he is able to show that f is an L-reduction from 3SAT-B to 3DM-3.

4.6.2 Modification of Kann's reduction

In this section, we describe a reduction $f' : \text{3SAT-}M \longrightarrow \text{3DM-3}$ such that for each satisfiable instance I of 3SAT-B, $f'(I)$ admits a perfect matching. In the reduction f above, even if I is satisfiable, there may be many root vertices that are not in an optimal matching, μ. In particular, if a clause c_j is satisfied by u_i and $u_{i'}$, then at most one of the edges

$$\{u_i[\gamma],\ s_1[j],\ s_2[j]\} \quad \text{and} \quad \{u_{i'}[\gamma'],\ s_1[j],\ s_2[j]\}$$

can appear in μ. Hence, at most one of $u_i[\gamma]$ and $u_{i'}[\gamma']$ can appear in μ. To remedy this problem, we define $f'(I)$ to be the disjoint union of three copies of $f(I)$

$$f'(I) = f(I)_1 \sqcup f(I)_2 \sqcup f(I)_3.$$

We then add an edge for each root vertex in $f(I)$ that contains the corresponding root vertices in each disjoint copy of $f(I)$. Specifically, if $u_i[\gamma]_1$, $u_i[\gamma]_2$, and $u_i[\gamma]_3$ are the three copies in $f'(I)$ of a root vertex $u_i[\gamma]$ in $f(I)$, then we include the edge

$$\{u_i[\gamma]_1,\ u_i[\gamma]_2,\ u_i[\gamma]_3\} \tag{4.3}$$

in $f'(I)$. We now describe the structure of optimal matchings μ' in $f'(I)$. Let μ_1, μ_2, and μ_3 be the restrictions of an optimal matching μ' for $f'(I)$ to $f(I)_1$, $f(I)_2$, and $f(I)_3$ respectively. Thus, we can write

$$M' = M_1 \cup M_2 \cup M_3 \cup R \tag{4.4}$$

where R contains those edges in μ' of the form (4.3).

Lemma 4.20. There exists an optimal matching μ' for $f'(I)$ such that the matchings μ_1, μ_2, and μ_3 contain precisely the same edges as an optimal matching μ for $f(I)$.

Proof. Suppose μ' is an optimal matching for $f'(I)$. We may assume without loss of generality that the matchings μ_1, μ_2 and μ_3 are all identical to some matching μ on $f(I)$. Indeed, if, say μ_1 is the largest of the three matchings, we can increase the size of μ' by replacing μ_2 and μ_3 with identical copies of μ_1. Since the only edges between μ_1, μ_2, and μ_3 are edges of the form (4.3), replacing μ_2 and μ_3 with copies of μ_1 cannot decrease the size of μ'. Thus, we may assume that

$$|\mu'| = 3\,|\mu| + |R|$$

where μ is some matching on $f(I)$, and R consists of edges of the form (4.3).

We will now argue that μ is indeed an optimal matching on $f(I)$, hence has the form described in Lemma 4.19. Notice that if μ is optimal for $f(I)$, then by including all edges in R containing uncovered root vertices, μ' covers every ring, tree, and root vertex. Thus, the only way to obtain a larger matching would be to include more clause edges. However, by Lemma 4.19, including more clause edges cannot increase the size of the matching μ. Thus, we may assume μ is an optimal matching for $f(I)$. □

Corollary 4.21. If I is an instance of 3-SAT-B with m clauses and $\mathrm{opt}(I) = cm$ for some $c \leq 1$, then an optimal matching μ' in $f'(I)$ leaves precisely $6(1-c)m$ vertices uncovered.

Lemma 4.22. There exists a constant $C > 0$ depending only on B such that the number of vertices in $f'(I)$ is at most Cm.

Proof. We bound the number of ring, tree, and clause vertices separately. Since the vertex set of $f'(I)$ consists of three disjoint copies of the vertices in $f(I)$, it suffices to bound the number of vertices in $f(I)$.

Ring vertices For each variable u_i, there are $K = O(B)$ rings, each consisting of $4d_i = O(B)$. Thus there are $O(W^2)$ ring vertices for each variable u_i, hence a total of $O(B^2 n)$ ring vertices in $f(I)$.

Tree vertices Since each ring vertex of the form $v_i[\gamma, k]$ is the leaf of a complete binary tree whose internal nodes and root are tree vertices, there are $O(B^2 n)$ tree vertices in $f(I)$.

Clause vertices There two vertices $s_1[j]$ and $s_2[j]$ associated to each of m clauses, hence there are $O(m)$ clauses in total.

Therefore, the total number of vertices in $f(I)$ and hence $f'(I)$ is $O(m + B^2 n)$. Clearly, we may assume that $n \leq m$, so that there are $O(B^2 m)$ vertices in $f'(I)$. □

Corollary 4.23. Let I be an instance of 3SAT-B, and let $M^* = \frac{1}{3}|f'(I)| = |f(I)|$ be the number of vertices in $f(I)$. Then for any $c < 1$, there exists a constant $c' < 1$ depending only on c and B such that:

- if $\mathrm{opt}(I) = m$ (i.e., I is satisfiable) then $\mathrm{opt}(f'(I)) = M^*$;
- if $\mathrm{opt}(I) \leq cm$ then $\mathrm{opt}(f'(I)) \leq c'M^*$.

Theorem 4.8 follows immediately from Corollary 4.23 and the following incarnation of the PCP theorem.

Theorem 4.24 (PCP Theorem [4, 3, 7])**.** There exist a absolute constants $c < 1$ and B such that it is NP-hard to determine if an instance I of 3SAT-B satisfies $\mathrm{opt}(I) = m$ or $\mathrm{opt}(I) \leq cm$.

Appendix A

Computing Distance to Stability

In this appendix, we describe an efficient method for computing the divorce distance to stability of a given matching, μ. Recall that the divorce distance to stability is given by

$$d(\mu) = \min_{\mu' \in \mathcal{M}} d(\mu, \mu') \quad \text{where} \quad d(\mu, \mu') = n - |\mu \cap \mu'| \,.$$

Since the set $\mathcal{M}$ of all stable matchings can be exponentially large [29, 20, 15], brute-force computation of $d(\mu)$ is infeasible. Fortunately, by exploiting the structure of $\mathcal{M}$, we are able to efficiently reduce the computation of $d(\mu)$ to a max-flow/min-cut problem of size quadratic in n. Thus, any number of efficient algorithms may be applied to compute $d(\mu)$.

A.1 The Rotation Poset and Digraph

Our exposition follows the work of Gusfield [14] and of Irving and Leather [20] (see also [15]). Let μ be a stable matching. A ***rotation*** ρ ***exposed by*** μ is a sequence of pairs

$$(w_0, m_0), (w_1, m_1), \ldots, (w_{r-1}, m_{r-1}) \in \mu$$

such that for each i, w_{i+1} is the first woman on m_i's preference list that prefers m_i to her partner m_{i+1} in μ (where addition is conducted modulo r). Given μ and ρ, we form a matching called the ***elimination of ρ from*** μ, denoted μ/ρ, which contains the pairs

$$(w_1, m_0), (w_2, m_1), \ldots, (w_0, m_{r-1}),$$

in addition to all pairs from μ that are not part of the rotation ρ. It is straightforward to verify that μ/ρ is a stable matching.

Let μ_0 denote the M-optimal stable matching (the matching found by the Gale-Shapley algorithm). Irving and Leather [20] prove that every stable matching can be obtained from μ_0 by successively eliminating a ***unique*** set of rotations that appear in μ_0 and subsequent stable matchings. Given a stable matching μ, let S_μ denote this unique set of rotations, which can be eliminated (starting at μ_0) to obtain μ. (S_μ may contain some rotations that are not exposed in μ_0, but only in a subsequent matching.)

We denote the set of all rotations exposed in one or more stable matchings in $\mathcal{M}$ by $\Pi(\mathcal{M})$. We endow $\Pi(\mathcal{M})$ with a partial order $\prec$ where $\rho \prec \sigma$ if $\rho \in S_\mu$ for every $\mu \in \mathcal{M}$ in which σ is exposed. In other words, $\rho \prec \sigma$ if whenever σ is eliminated during the construction of a stable matching by elimination from μ_0, it is the case that ρ has been eliminated before σ. A subset $S \subseteq \Pi(\mathcal{M})$ is ***(downward) closed*** if for all $\sigma \in S$ and $\rho \prec \sigma$, we have that $\rho \in S$. Irving and Leather prove the following remarkable correspondence between closed subsets of $\Pi(\mathcal{M})$ and stable matchings.

Theorem A.1 (Irving and Leather [20])**.** The map $\mu \mapsto S_\mu$ is a bijection between $\mathcal{M}$ and the set of closed subsets of $\Pi(\mathcal{M})$.

For algorithmic purposes, it is advantageous to have a sparse representation of the partial order $\prec$ on $\Pi(\mathcal{M})$, which preserves its closed subsets. To this end, Gusfield [14] proved the following theorem. For the remainder of this section, we use the standard notation $\tilde{O}$ to suppress $\log n$ factors, which we find less interesting in the context of our discussion of time complexity in this section (in contrast to the discussion of query complexity in the rest of this paper).

Theorem A.2 (Gusfield [14])**.** There exists a directed acyclic graph $G(\mathcal{M})$ with vertices $\Pi(\mathcal{M})$, called the ***rotation digraph***, whose transitive closure is the partial order $\prec$ on $\Pi(\mathcal{M})$. $G(\mathcal{M})$ can be computed from the full preferences of all participants in time $\tilde{O}(n^2)$. In particular, the edge and vertex sets of $G(\mathcal{M})$ both have cardinality $O(n^2)$.

A.2 Rotation Weights

In this section, we show how to assign weights to rotations $\rho \in \Pi(\mathcal{M})$ in such a way that $d(\mu, \mu')$ can be computed directly from $d(\mu, \mu_0)$ and $S_{\mu'}$. Let ρ be any rotation and μ' a stable matching in which ρ is exposed. For any matching μ, we define the ***weight*** γ of ρ relative to μ by

$$\gamma_\mu(\rho) = |\mu \cap (\mu'/\rho)| - |\mu \cap \mu'| \,.$$

(The absence of μ' from the notation $\gamma_\mu(\rho)$ becomes clear in Eq. A.1 below.) That is, $\gamma_\mu(\rho)$ is the net change, following the elimination of ρ from μ', in the number of pairs contained in the intersection of μ and μ'. We note that $\gamma_\mu(\rho)$ can be computed directly from ρ (without being given an explicit stable matching μ' which exposes ρ). Specifically, letting $\rho' = \{(w_1, m_0), (w_2, m_1), \ldots, (w_0, m_{r-1})\}$ be the set of pairs replacing ρ when eliminating ρ from any stable matching, we have

$$\gamma_\mu(\rho) = |\mu \cap \rho'| - |\mu \cap \rho| \,. \tag{A.1}$$

We note that in general, ρ' is not a rotation.

Lemma A.3. For any matching μ and stable matching $\mu' \in \mathcal{M}$,

$$|\mu \cap \mu'| = |\mu \cap \mu_0| + \sum_{\rho \in S_{\mu'}} \gamma_\mu(\rho),$$

where μ_0 is the M-optimal stable matching.

Proof. We argue by induction on $|S_{\mu'}|$. If $S_{\mu'} = \varnothing$, then $\mu' = \mu_0$, so the result is immediate. Suppose the claim is true for μ' and σ is a rotation exposed in μ', then by the induction hypothesis and by definition of $\gamma_\mu(\sigma)$,

$$|\mu \cap (\mu'/\sigma)| = |\mu \cap \mu'| + \big(|\mu \cap (\mu'/\sigma)| - |\mu \cap \mu'|\big) = |\mu \cap \mu_0| + \sum_{\rho \in S_{\mu'}} \gamma_\mu(\rho) + \gamma_\mu(\sigma),$$

which gives the desired result. □

Applying Lemma A.3 and Theorem A.1, we obtain the following result.

Theorem A.4. Let μ be a matching. Then

$$d(\mu) = d(\mu, \mu_0) - \max_S \sum_{\rho \in S} \gamma_\mu(\rho),$$

where the maximum is taken over closed subsets $S \subseteq \Pi(\mathcal{M})$ (and where μ_0 is the M-optimal stable matching).

Proof. By Lemma A.3, for any stable matching μ',

$$d(\mu, \mu') = n - |\mu \cap \mu'| = n - |\mu \cap \mu_0| - \sum_{\rho \in S_{\mu'}} \gamma_\mu(\rho) = d(\mu, \mu_0) - \sum_{\rho \in S_{\mu'}} \gamma_\mu(\rho).$$

By Theorem A.1, $\mu' \mapsto S_{\mu'}$ is a bijection onto the set of closed subsets of $\Pi(\mathcal{M})$. Thus,

$$d(\mu) = \min_{\mu' \in \mathcal{M}} d(\mu, \mu') = d(\mu, \mu_0) - \max_S \sum_{\rho \in S} \gamma_\mu(\rho),$$

as desired. □

A.3 Reduction to Max-Flow/Min-Cut

The divorce distance $d(\mu, \mu_0)$ can easily be computed in $\tilde{O}(n^2)$ time by using the Gale-Shapley algorithm to compute μ_0. Thus, Theorem A.4 reduces the problem of computing $d(\mu)$ to finding the ***maximum closure*** (i.e., maximum-weight closed subset) in a directed acyclic graph. This problem is well studied, in particular for its applications to open-pit mining (see, e.g., [40, 18]). For completeness, we briefly describe Picard's [40] efficient reduction of maximum closure to a max-flow/min-cut problem.

Remark A.5. Picard [40] and Hochbaum [18] use the term "closure" to refer to a subset of a poset which is *closed under successors* or *upward closed*. That is, S is a closure if whenever $\rho \in S$ and $\rho \prec \sigma$, then $\sigma \in S$. This disagrees with Irving and Leather's [20] (and our) usage of closed to mean downward closed, which appears to be more prevalent in the stable matching literature. In order to directly apply Picard's reduction, we will reverse the direction of edges in the rotation digraph $G(\mathcal{M})$ so that our (downward) closed subsets become upward closed with respect to the modified graph.

Let $G(\mathcal{M})$ be the rotation digraph described in Theorem A.2. Since the partial order $\prec$ on $\Pi(\mathcal{M})$ is the transitive closure of $G(\mathcal{M})$, they share the same closed subsets, and so to compute $d(\mu)$, it suffices to maximize $\sum_{\rho \in S} \gamma_\mu(\rho)$ over closed subsets of $G(\mathcal{M})$. Denote the vertex and edge sets of $G(\mathcal{M})$ by $V = \Pi(\mathcal{M})$ and E respectively. Let

$$V^+ = \{\rho \in V \mid \gamma_\mu(\rho) \geq 0\} \quad \text{and} \quad V^- = \{\rho \in V \mid \gamma_\mu(\rho) < 0\}.$$

We form a new st-graph $\tilde{G} = (\tilde{V}, \tilde{E})$ where $\tilde{V} = V \cup \{s, t\}$ and

$$\tilde{E} = E' \cup \left\{(s, \rho) \,\middle|\, \rho \in V^+\right\} \cup \left\{(\rho, t) \,\middle|\, \rho \in V^-\right\}.$$

Here, $E' = \{(\sigma, \rho) \mid (\rho, \sigma) \in E\}$ (see Remark A.5). We assign capacity $c(u, v)$ to each edge

$(u, v) \in \tilde{E}$ by

$$c(u,v) = \begin{cases} \infty & (u,v) \in E' \\ \gamma_\mu(\rho) & (u,v) = (t,\rho) \\ -\gamma_\mu(\rho) & (u,v) = (\rho,s). \end{cases}$$

Theorem A.6 (Picard [40])**.** The source set of a minimum st-cut in $\tilde{G}$ is a maximum closure in $G(\mathcal{M})$.

In light of Theorem A.6, we can reduce the computation of $d(\mu)$ to known efficient algorithms for max-flow/min-cut. We summarize the procedure as follows.

Algorithm 7 $\mathrm{DivorceDistance}(\mu, P_W, P_M)$ — compute $d(\mu)$ with respect to preferences P_W, P_M.

1. Use the Gale-Shapley algorithm to compute μ_0 and compute $d(\mu, \mu_0) = n - |\mu \cap \mu_0|$.
2. Construct the rotation digraph $G(\mathcal{M})$ and the related graph $\tilde{G}$.
3. Compute the weights in $\tilde{G}$ by computing $\gamma_\mu(\rho)$ for each rotation $\rho \in G(\mathcal{M})$, using Eq. (A.1).
4. Find a minimum st-cut in $\tilde{G}$, and let S be the source set in the cut.
5. Return $d(\mu, \mu_0) - \sum_{\rho \in S} \gamma_\mu(\rho)$.

Theorem A.7. Given an SMP instance $(M, W, \mathcal{P})$ and a matching μ, running the procedure $\mathrm{DivorceDistance}(\mu, P_W, P_M)$ computes the divorce distance to stability $d(\mu)$ in time $\tilde{O}(n^4)$.

We remark that since (P_W, P_M) has size $\tilde{\Theta}(n^2)$, the runtime of $\mathrm{DivorceDistance}$ is nearly quadratic in the input size.

Proof. The correctness of $\mathrm{DivorceDistance}(\mu, P_W, P_M)$ follows immediately from Theorems A.4 and A.6. We analyze the runtime as follows. Step 1 can be computed in time

$\tilde{O}(n^2)$ using the Gale-Shapley algorithm and brute force computation of $d(\mu, \mu_0)$. For Step 2, by Theorem A.2, $G(\mathcal{M})$ (and hence $\tilde{G}$) can also be computed in time $\tilde{O}(n^2)$. The weights in Step 3 can be computed in linear time for each rotation, so computing all the weights can be accomplished in time $\tilde{O}(n^3)$. For Step 4, the min-cut can be computed in time $\tilde{O}(|\tilde{E}||\tilde{V}|) = \tilde{O}(n^4)$ using, for example, Hochbaum's algorithm [18]. $\square$

We remark that since Algorithm 7 finds both μ_0 and $S = S_{\mu'}$ for a stable matching μ' closest to μ, it is a trivial task, which does not increase the asymptotic runtime complexity of Algorithm 7, to also compute μ' in addition to computing $d(\mu) = d(\mu, \mu')$.

Bibliography

[1] David J Abraham, Péter Biró, and David F Manlove. Almost stable matchings in the roommates problem. In *Approximation and online algorithms*, pages 1–14. Springer, 2006.

[2] Ahmet Alkan. Nonexistence of stable threesome matchings. *Mathematical Social Sciences*, 16(2):207–209, 1988.

[3] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *J. ACM*, 45(3):501–555, 1998.

[4] Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[5] Xiaohui Bei, Ning Chen, and Shengyu Zhang. On the complexity of trial and error. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pages 31–40, 2013.

[6] Jen-Hou Chou and Chi-Jen Lu. Communication requirements for stable marriages. In *Proceedings of the 7th International Conference on Algorithms and Complexity (CIAC)*, pages 371–382, 2010.

[7] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12, 2007.

[8] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, PODC '14, pages 367–376, New York, NY, USA, 2014. ACM.

[9] Kimmo Eriksson and Olle Häggström. Instability of matchings in decentralized markets with various preference structures. *International Journal of Game Theory*, 36(3):409–420, March 2008.

[10] Patrik Floréen, Petteri Kaski, Valentin Polishchuk, and Jukka Suomela. Almost Stable Matchings by Truncating the Gale–Shapley Algorithm. *Algorithmica*, 58(1):102–118, 2010.

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

[12] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[13] Yannai A. Gonczarowski, Noam Nisan, Rafail Ostrovsky, and Will Rosenbaum. A stable marriage requires communication. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1003–1017. SIAM, 2015.

[14] Dan Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987.

[15] Dan Gusfield and Robert W Irving. *The Stable Marriage Problem: Structure and Algorithms*, volume 54. MIT press Cambridge, 1989.

[16] Michal Hańćkowiak, Michal Karoński, and Alessandro Panconesi. On the distributed complexity of computing maximal matchings. *SIAM Journal on Discrete Mathematics*, 15(1):41–57, 2001.

[17] Avinatan Hassidim, Yishay Mansour, and Shai Vardi. Local computation mechanism design. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 601–616. ACM, 2014.

[18] Dorit S Hochbaum. A new—old algorithm for minimum-cut and maximum-flow in closure graphs. *Networks*, 37(4):171–193, 2001.

[19] Chien-Chung Huang. Two's Company, Three's a Crowd: Stable Family and Threesome Roommates Problems. In Lars Arge, Michael Hoffmann, and Emo Welzl, editors, *Algorithms – ESA 2007*, pages 558–569. Springer Berlin Heidelberg, 2007.

[20] Robert W. Irving and Paul Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667, 1986.

[21] Robert W Irving, David F Manlove, and Gregg O'Malley. Stable marriage with ties and bounded length preference lists. *Journal of Discrete Algorithms*, 7(2):213–219, 2009.

[22] Amos Israeli and Alon Itai. A fast and simple randomized parallel algorithm for maximal matching. *Information Processing Letters*, 22(2):77–80, 1986.

[23] Kazuo Iwama, Shuichi Miyazaki, Yasufumi Morita, and David Manlove. Stable marriage with incomplete lists and ties. In *Automata, Languages and Programming*, pages 443–452. Springer, 1999.

[24] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

[25] Viggo Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37(1):27–35, 1991.

[26] Alex Kipnis and Boaz Patt-Shamir. A note on distributed stable matching. In *Proceedings of the 28th ACM symposium on Principles of distributed computing*, pages 282–283, New York, NY, USA, 2009. ACM.

[27] Zoltán Király. Better and simpler approximation algorithms for the stable marriage problem. *Algorithmica*, 60(1):3–20, 2011.

[28] Zoltán Király. Linear time local approximation algorithm for maximum stable marriage. *Algorithms*, 6(3):471–484, 2013.

[29] Donald E. Knuth. *Marriage stables et leurs relations avec d'autres problèmes combinatoires*. Les Presses de l'Université de Montréal, Quebec, Canada, 1976.

[30] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, UK, 1997.

[31] David Manlove. *Algorithmics of matching under preferences*. World Scientific Publishing, 2013.

[32] D. G. McVitie and L. B. Wilson. The stable marriage problem. *Communications of the ACM*, 14(7):486–490, 1971.

[33] Cheng Ng and Daniel Hirschberg. Lower bounds for the stable marriage problem and its variants. *SIAM Journal on Computing*, 19(1):71–77, 1990.

[34] Cheng Ng and Daniel S Hirschberg. Three-Dimensional Stable Matching Problems. *SIAM Journal on Discrete Mathematics*, 4(2):245–252, 1991.

[35] Rafail Ostrovsky and Will Rosenbaum. Fast distributed almost stable matchings. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, PODC '15, pages 101–108, New York, NY, USA, 2015. ACM.

[36] Rafail Ostrovsky and Will Rosenbaum. It's not easy being three: The approximability of three dimensional stable matching problems. In *Proceedings of the 3rd International Workshop on Matching Under Preferences*, MATCH-UP '15, pages 90–101, 2015.

[37] Katarzyna Paluch. Faster and simpler approximation of stable matchings. *Algorithms*, 7(2):189–202, 2014.

[38] Christos H Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 425–440, New York, NY, USA, 1991. ACM.

[39] David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.

[40] Jean-Claude Picard. Maximal closure of a graph and applications to combinatorial problems. *Management Science*, 22(11):1268–1272, 1976.

[41] Alexander A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106, 1992.

[42] Alvin E. Roth. On the allocation of residents to rural hospitals: A general property of two-sided matching markets. *Econometrica*, 54(4):425–427, 1986.

[43] Alvin E Roth and Marilda A Oliveira Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press, 1992.

[44] Ilya Segal. The communication requirements of social choice rules and supporting budget sets. *Journal of Economic Theory*, 136:341–378, 2007.

[45] M Utku Ünver. On the survival of some unstable two-sided matching mechanisms. *International Journal of Game Theory*, 33(2):239–254, 2005.

[46] L. B. Wilson. An analysis of the stable marriage assignment algorithm. *BIT Numerical Mathematics*, 12(4):569–575, 1972.

[47] Hiroki Yanagisawa. *Approximation algorithms for stable marriage problems*. PhD thesis, Kyoto University, Graduate School of Informatics, 2007.

[48] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.