Deep Learning for Two-Sided Matching Markets

Shira Li

submitted in partial fulfillment of the requirements
for the degree of Bachelor of Arts with Honors in
Mathematics and Computer Science

Advisors: David Parkes and Scott Duke Kominers

Math Advisor: Clifford Taubes

Harvard University

March 25, 2019

Abstract

In the economic theory problem of two-sided matching, we seek to create matchings between agents on two sides of a market, each of whom has ranked, ordinal preferences over agents on the other side. Of particular interest are stable matchings, where no pair of agents would mutually prefer to be matched to each other than to their assigned partners. Not only theoretically relevant, matching algorithms can successfully solve real-world conundrums, such as how to optimally match medical students to hospital residency programs.

The first part of this thesis gives an introduction to the theory of two-sided matching, focusing on methods of finding stable matchings, the structure of the set of stable matchings, and questions of strategic behavior under stable matching mechanisms. The second part introduces a novel deep learning framework that models one-to-one matching markets and demonstrates its ability to learn approximately stable matchings, successfully replicating Gale-Shapleys original stability results. Furthermore, this framework finds matchings which are both complete and achieve high aggregate welfare. Given this, our framework holds promise for examining the until now, poorly understood trade-offs between dominant-strategy incentive compatibility, a strategic question for individual agents, and the aggregate stability for all agents of a matching.

Acknowledgments

First, I would like to thank my advisors, Professor David Parkes and Professor Scott Duke Kominers for introducing me to this wonderful field of research in their classes, CS136 and ECON2099 as well as supporting and advising me in my research every step of the way over the past two semesters.

I would also like to thank my math advisor Professor Clifford Taubes, whose feedback throughout the process has greatly enhanced the readability and quality of this thesis.

I also thank Zhe Feng, who has spent countless hours discussing ideas, teaching me about the intricacies of PyTorch and the Odyssey GPU, and helping me debug the seemingly endless code errors. This thesis could not have existed without his extensive guidance.

In addition, I would like to thank Professor Shengwu Li and Professor Martin Nowak for reading and providing feedback on this thesis as my external Computer Science and Mathematics thesis readers, respectively.

More generally, I would also like to thank the Mathematics, Computer Science, and Economics departments for always challenging me intellectually and allowing me to discover exciting fields of research at their intersection.

I would also like to thank my friends for all the advice and encouragement they have provided throughout this process - all the members of Scott's Market Design group for research pointers, Kevin Zhang with matplotlib, Susan Xu for the hours spent coding in MD, Andy Wang and Molly Nolan for their amazing emotional support in the form of food and hugs, and all my friends on Israel Trek who supported me over spring break as I wrote this thesis.

Lastly, I would like to thank my parents for everything they have sacrificed for me. From an early age, they instilled in me a deep love for learning, and to them, I am forever grateful.

Thesis Structure

This paper is divided into two parts. Part I (Chapters 1-4) serves as a primer on one-to-one and many-to-one matching markets. This section is expository and heavily draws upon the work of Roth and Sotomayor in their book, Two-Sided Matching, A Study in Game-Theoretic Modeling and Analysis [28], covering the history, mathematical structures, and strategic questions that arise from the stable matching problem. Part II (Chapters 5-7) focuses on applications of deep learning to the stable matching problem. This section contains original research demonstrating the successful replication of theoretical stability results using deep learning methods and lays the groundwork for future work using this deep learning framework to understand the trade-offs between dominant strategy incentive compatibility and stability.

NB: As Part I is more expository in nature, it is intended to satisfy the Mathematics thesis requirements. Part II, focused more on original research using deep learning methods, is intended to satisfy the Computer Science thesis requirements.

Contents

Part I

Foundations of Matching Markets

Chapter 1

A Brief History

In this chapter, we provide a brief history of the field of matching theory. We begin with a motivating example and move on to an overview of the theoretical questions related to matching that concern mathematicians, computer scientists, and economists alike. Assuming limited knowledge of the field, we will then move from theory to practice and discuss concrete examples of both historical and present-day problems where matching theory has been successfully applied, to illustrate its real-world importance.

1.1 Mathematics of Matching

In this section, we will first present a motivating example, and then provide an overview of the large, theoretical questions in the field of matching that will be covered more in depth in the following chapters of Part I.

1.1.1 The Marriage Problem

Consider a small, isolated village in which we'd like to arrange marriages between pairs of men and women (assuming only heterosexual preferences), and we want to pair them such that there are no divorces [14]. In particular, this means that if a man and woman who are not paired mutually prefer to be married to each other over their current spouses, then two divorces would occur, and we want to avoid this scenario. In this example, we also assume that people would prefer to be married than single. We see that this is an example of one-to-one matching, because there is no polygamy, and therefore, each man is married to at most one woman, and vice versa.

First, consider a village with only 1 man, and 0 women. How can we arrange marriages so that there are no divorces? There are no possible marriages since there are no women, and trivially, we will have no divorces either.

Now, consider a village with exactly 1 man and 1 woman. How can we arrange marriages so that there are no divorces? Given that each individual prefers to be married rather than being single, we can pair the two together, and there will be no divorces either, because there are no alternative options.

However, consider if the woman finds the man unacceptable to marry, and would prefer to be single than marry him. In this case, even though there are two people, no marriages can occur, because the woman does not find the man acceptable. This leads to a concept of individual rationality, which we will formally introduce in Chapter 2, which asserts that marriages can only occur between mutually acceptable partners.

Now, consider a village with 2 men and 1 woman. How can we arrange marriages so that there are no divorces? Since the number of marriages is limited by the number of women in this scenario, should she marry man A or man B? Say that the woman prefers intelligence (man A) to muscular physique (man B). Then, given that there are fewer women, intuition tells us that pairing the woman with man A will be the only possible marriage that does not lead to divorce, as if she is married to man B, she will divorce him in favor of marrying man A.

Now, consider a village with n total men and 1 woman. By similar intuition to the previous example, we can only have one marriage, since the number of marriages is limited by the number of women. Thus, out of the n total men, the only marriage that will not lead to divorce is when the singular woman marries her most preferred man out of the group.

Simple enough, you might say. However, what happens if there are n men and v women in this small, isolated village. Now, the preferences of each of the men and women might conflict with each other, and the answer is not as simple as it was above. How do we arrange marriages so that there are no divorces then?

1.1.2 Theoretical Questions

In the fields of mathematics, computer science, and economics, the above problem is known as the *stable marriage problem* or *stable matching problem.* Simply put, to solve this problem is to find a stable matching between two equally sized sets of elements given an ordering of preferences for each element, where a matching is a mapping from elements of one set to elements of the other set, and a stable matching is such that no pair (A, B) would *both* be better off matched with each other than their current matches. This question was solved in 1962, with the publication of the now-famous paper, College Admissions and the Stability of Marriage, by David Gale and Lloyd Shapley. Most importantly, their paper demonstrated that in any setting, not only does a stable matching always exist, but it can be found in quadratic time using the deferred acceptance algorithm. In the following decades, even greater progress was made in the field of matching markets, with researchers understanding not only the existence of a stable matching in all settings, but the types and structure of stable matchings as well as the game-theoretic properties of matching on a market level and the strategic incentives of individual agents. Furthermore, the results of stable matching derived by theoretical economists are beautiful examples of lattice structures in mathematics, and many older results have been re-derived using the mathematics of lattices, such as Tarski's Fixed Point Theorem.

1.2 From Theory to Practice

While the stable marriage problem and matching markets at large are interesting theoretical questions to study in it of itself, what makes the field most palpably interesting for me is the extraordinary extent to which the results have impact beyond a research paper or classroom chalkboard. Below, we shall provide a few examples, both historical and modern, that demonstrate the critical importance of continuing to push boundaries in the field.

1.2.1 Medical Matching

In the 1920s, hospitals introduced the first medical residency programs, offering concentrated exposure to clinical medicine for the students and a supply of relatively cheap labor for the hospitals. The competition for medical interns was great, given that a far higher number of new residency positions were available than the number of graduating medical students applying for such positions [28]. As a result, two types of market failures began to occur.

Firstly, each hospital, in its desire to finalize binding agreements with their interns earlier than their competitor hospitals, would move their acceptance date ever so slightly earlier. Whereas previously, interns had to accept their offers by the end of their senior year of medical school, by 1944, the acceptance date had been moved *two full years* earlier. Hospitals were accepting interns before they had ever seen a student's final grades, and students were accepting offers before they had even realized they might be afraid of surgery or blood! To rectify this, the Association of American Medical Colleges (AAMC) prevented the release of transcripts or recommendation letters until the end of junior year, effectively delaying residency selection until the senior year of medical school.

However, hospitals still wanted to finalize their interns faster than their competitors, so the time limit for responding to offers rapidly shortened. The AAMC mandated a ten-day consideration period in 1945, and by 1949, it was shortened to just twelve hours. Unsurprisingly, the compressing of this pivotal decision-making period led to a slew of additional problems. Hospitals whose first-choice interns eventually rejected them would see their preferred alternate candidates accept other positions as well, leaving them with no one. Students who were on the wait list at their first-choice residency program would be forced to accept a less preferable choice, even if seats later opened up at their first-choice. Much to the displeasure of hospitals, some students even reneged on their original residency commitments and chose to accept a later offer from a more preferable hospital.

Given these market failures, where individually rational decisions made by individual hospitals and students led to both costly and sub-optimal larger market outcomes, a centralized matching system, called the National Resident Matching Program (NRMP) was proposed by the AAMC in 1952. Students would rank in order of preference hospital programs, and similarly, hospitals would rank in order of preference the students who had applied to their programs. These rankings would be submitted to a central bureau, who would input this information into an algorithm to produce a matching for hospitals and interns. While this new system was completely voluntary and hospitals and students remained completely free to make separate arrangements with hospitals, over 95% of eligible students and hospitals participated in the centralized system through the 1970s. This leads us to two questions:

1. What explains the ridiculous actions of hospitals and interns in their residency labor market prior to 1952?
2. Why was the centralized matching system such a successful alternative, able to attain such high rates of *voluntary* participation?

For answers, we turn to the field of market design – more specifically, matching theory.

1.3 Market Design and Matching Theory

Market design is a field at the intersection of economics and game theory that takes an engineering approach to designing economic incentives in strategic settings, such that players who act rationally converge upon a desired objective. For matching theory in particular, we want a mathematical relationship that describes the formation of new relationships (matches), from unmatched agents of two types. As shown in the above example, matching theory has often been used to study labor markets, as employment can be characterized as a match between a firm and employee. Moreover, we may also consider heterosexual marriages, which require the pairing of men and women, or urban school districts like Boston, New York City, or Chicago, which require the pairing of students and high schools.

The types of problems studied in matching theory have a few characteristics. Firstly, they are *two-sided*, meaning that agents in these markets belong to one of two disjoint sets. It is not possible for an intern to become a hospital or a man to become a woman (in this setting). Secondly, the nature of matching markets is that they are *bilateral*. In a marriage, if a woman is married to a man, the man is also married to the woman and if I am an intern at a hospital, that hospital also employs me as an intern. We also distinguish between matching markets that are one-to-one, many-to-one, or many-to-many. *One-to-one matchings*, such as heterosexual marriages, require each agent on one side of the market to match to exactly one agent on the other side of the market. *Many-to-one* matchings can have many agents on one side of the market match to one agent on the other side. For example, many residents can intern at one hospital. Lastly, *many-to-many* matchings allow multiple agents on one side of the market to match multiple agents on the other side. For example, we can consider firms and consultants, where firms can hire more than one consultant, and consultants can work for more than one firm. For the purpose of this thesis, we will be focusing specifically on two-sided, bilateral, one-to-one matching markets. Lastly, and most importantly, each agent in the market has heterogeneous and idiosyncratic preferences with individual-specific utilities. Each side of the market is unlikely to have uniform agreement over who ought to be ranked first, second, and so on, and therefore, the utility functions of each agent can vary greatly.

1.4 Field Development and Further Applications

Over the past sixty years, the theoretical and practical applications of matching theory and market design have developed in parallel, each informing the other. In 1984, Alvin

Roth would use the theoretical foundations provided by Gale and Shapley in 1962 to prove the effectiveness of the NMRP centralized matching system. Roth would go on to help redesign both the New York City [1] and Boston [2] public school matching systems, creating a centralized system for students and schools to express their preferences over each other while also incorporating necessary logistical constraints, from distance of bus transportation to keeping siblings in the same schools. Furthermore, Roth created the New England Program for Kidney Exchange [20], facilitating a registry and matching program that pairs compatible kidney donors and recipients.

The use of matching theory can even be seen on Harvard's campus. The freshman seminar program uses one version of the deferred acceptance algorithm to be explained in Chapter 2 to place freshmen into seminars according to their ranked preferences as well as the professor's preferences over students given their application essays. In addition, the finance recruitment process for roles in investment banking and sales and trading on Harvard's campus has unraveled in a similar fashion to the hospital-intern example. Just five years ago, students applied for junior summer internships in their junior spring, receiving offers in March or April. Three years ago, the recruiting timeline began inching earlier, and offers for junior summer internships were given out by the end of junior fall. Two years ago, internship offers were given out sophomore summer, and nowadays, students are asked to respond to internship offers for their junior summer before sophomore spring has even ended. Even worse, this has led to 're-negging', where students who have confirmed their internship offers will later reject it for another position. While financial recruiting may be a harder market to tackle than the hospital-intern market, this type of unraveling is astonishingly similar to the behavior discussed in Section 1.2.1.

Chapter 2

Introduction to Stable Matching

In this chapter, we will formally define the stable matching problem for the one-to-one and many-to-one settings and also prove the existence of stable matchings in both cases.

2.1 One-to-One Matching

2.1.1 Formal One-to-One Model

To formalize the marriage problem, we define the following notation for a one-to-one matching market. The *stable marriage* problem is defined such that:

- M is a finite set of n men, $\{m_1, \ldots, m_n\}$,
- W is a finite set of v women, $\{w_1, \ldots, w_v\}$,
- $M \cap W = \emptyset$,
- $P(i)$ with $i \in M \cup W$ is the strict preference relation of individual i over every member of the opposite gender.

To be clear, say that man m_1 prefers most to be married to woman w_1, second to w_2, and so on. Then, we use $\succ$ to denote the strict preference relation, and say that the preferences of man m_1 can be represented as $P(m_1) = w_1 \succ w_2 \succ \cdots \succ w_v$. However, it is possible that there are some women that man m_1 finds unacceptable, and he would prefer to be single rather than married to them. In this case, we include himself, m_1 in the relation. If his preferences are of the form $P(m_1) = w_1 \succ m_1 \succ w_2 \succ \cdots \succ w_v$, then his first choice is to be married to woman w_1, and otherwise, he would prefer to be single (matched to himself) over being married to any of the other women in the market. When considering individuals, we can also write that $w_1 \succ_{m_1} w_2$ if we want to specifically indicate that man m_1 prefers w_1 over w_2.

Definition 2.1 (One-to-One Matching)**.** *A matching μ is defined as a one-to-one correspondence from the set $M \cup W$ onto itself such that if $\mu(m) \neq m$ then $\mu(m) \in W$ and if $\mu(w) \neq w$ then $\mu(w) \in M$ for $m \in M$ and $w \in W$.*

This means that if an individual is not single, then they must be matched to someone of the opposite gender in a marriage. We refer to $\mu(x)$ as the match of x, and we require $\mu^2(x) = x$, meaning that if man m is matched to woman w, then woman w is also matched to man m (bilateral condition).

Definition 2.2 (Individual Rationality)**.** *A matching μ is individually rational if each agent is acceptable to their respective spouse.*

If a matching μ contains a pair (m, w) where at least one of the individuals would prefer to be single than be matched to each other, the matching is not individually rational. An individually rational matching must always exist because the matching such that every individual is single is one such example.

Definition 2.3 (Stability)**.** *A matching μ is stable if there are no blocking pairs (m, w) with $m \in M, w \in W$ such that $m \succ_w \mu(w)$ and $w \succ_m \mu(m)$ and μ is individually rational.*

In the context of the stable marriage problem, this means that there are no divorces, because if a blocking pair (m, w) existed, m, w both prefer each other to their current partner, and thus, would divorce their spouses to marry each other.

2.1.2 Existence of Stable Matchings

Given that matching pairs of men and women together such that no divorces occur seems to be a difficult problem, it is surprising that Gale-Shapley [8] were able to constructively prove that a stable matching always exists using the deferred acceptance algorithm.

Algorithm 2.4 (Deferred Acceptance)**.** *The procedure of the deferred acceptance algorithm is as follows.*

1. *To start, each man proposes to his most preferred woman. Each woman rejects the proposal of any man who is unacceptable to her, and each woman who receives more than one proposal rejects all but her most preferred man of these. Any man whose proposal is not rejected is engaged.*

2. *At every following step, any man who was rejected at the previous step proposes to his next highest choice, so long as there remains an acceptable woman to whom he has not yet proposed. If at any step a man has proposed and been rejected by all acceptable woman on his list, he issues no further proposals.*

3. *Each woman receiving new proposals rejects any from unacceptable men, and also rejects all but her most preferred among the group consisting of new proposers together with any man she may have kept engaged from the previous step.*

This algorithm stops after any step in which no man is rejected, because at this point, every man is either engaged to some woman or has been rejected by every acceptable woman on his list. If there are n men and n women, there are a maximum of n^2 proposals to go through, and this algorithm will run in $O(n^2)$ time. Moreover, the algorithm must always terminate because there are only a finite number of men and women, with no man proposing more than once to any woman. We call this algorithm deferred acceptance, because women are able to keep the best available man at any step engaged, without immediately accepting him.

To better understand the deferred acceptance algorithm, consider the following example below, with three men and three women.

$$\begin{aligned} P(m_1) &= w_1 \succ w_2 \succ w_3 \qquad & P(w_1) &= m_3 \succ m_1 \succ m_2 \\ P(m_2) &= w_1 \succ w_3 \succ w_2 \qquad & P(w_2) &= m_2 \succ m_1 \succ m_3 \\ P(m_3) &= w_3 \succ w_2 \succ w_1 \qquad & P(w_3) &= m_1 \succ m_2 \succ m_3 \end{aligned}$$

Step 1: Each man proposes to his most preferred woman. m_1, m_2 both propose to w_1 and m_3 proposes to w_3.

Step 2: Since w_1 has two proposals and prefers m_1 to m_2, w_1 rejects m_2. The current matches are (m_1, w_1) and (m_3, w_3).

Step 3: Since both m_1, m_3 have engagements, only m_2 proposes to his second choice, w_3.

Step 4: w_3 prefers m_2 to m_3 so w_3 rejects m_3. The current matches are (m_1, w_1) and (m_2, w_3).

Step 5: Since both m_1, m_2 have engagements, only m_3 proposes to his second choice, w_2.

Step 6: All men and women now have matches, so the algorithm ends. Our final marriages are (m_1, w_1), (m_2, w_3), and (m_3, w_2).

This leads us to the proof that shows the existence of a stable matching.

Theorem 2.5 (Stable Matching)**.** *Given any set of strict preferences for men and women, a stable matching always exists between the two sides of the market.*

Proof. To prove that the matching μ derived from the deferred acceptance algorithm is stable, consider for contradiction that some man m and some woman w are not married to each other in μ, but m prefers w to his current match $\mu(m)$. Then, woman w must have been acceptable to man m and thus, he must have proposed to her before proposing to his current match $\mu(m)$. However, since he was not engaged to w when the deferred acceptance algorithm stopped, this means that woman w must have at some point rejected man m for someone she liked at least as well as him. Thus, while m may prefer woman w to $\mu(m)$, we see that w must prefer her current matching $\mu(w)$ over man m. Thus, for any choice of man m and woman w, (m, w) do not constitute a blocking pair. Thus, there are no blocking pairs and μ is stable. □

2.2 Many-to-One Matching

However, as discussed in the introduction, not all markets are one-to-one. For example, in the hospital-intern market, many interns can be matched to one hospital. We will now define the analogous mathematical language to discuss the many-to-one hospital-intern problem and again, show the existence of stable matchings in this scenario as well.

2.2.1 Formal Many-to-One Model

The *hospital-intern problem* is defined such that:

- H is a finite set of hospitals $\{h_1, \ldots, h_n\}$,
- S is a finite set of students $\{s_1, \ldots, s_m\}$,
- $H \cap S = \emptyset$,
- q_{h_i} is the quota for hospital h_i, a positive integer that indicates the maximum number of seats hospital h_i has to fill,
- $P(i)$ for $i \in H \cup S$ is the strict preference relation for each hospital or student in the market over every member of the opposite set.

Definition 2.6 (Many-to-One Matching)**.** *A matching μ is defined as a function from the set $H \cup S$ to a set of unordered elements of $H \cup S$ such that*

1. *$|\mu(s)| = 1$ for every student s and $\mu(s) = s$ if $\mu(s) \notin H$*

2. *$|\mu(h)| = q_h$ for every hospital $h \in H$ and if the number of students in $\mu(h) = r < q_h$, then $\mu(h)$ also contains $q_h - r$ copies of hospital h*

3. *$\mu(s) = h \iff s \in \mu(h)$*

The conditions tell us a few things. First, every student is matched to either a hospital or his/herself. Second, every hospital is matched to q_h students and if the number of students currently accepted does not hit the quota, the hospital will also be matched to itself. Lastly, any student-hospital pair is bilateral.

Definition 2.7 (Group Preferences)**.** *The preference relation $P_h^{\#}$ denotes preferences over groups of students for each hospital $h \in H$. $P_h^{\#}$ over sets of students is responsive to preferences $P(h)$ over individual students if for $\mu'(h) = \mu(h) \cup \{\sigma_1\} \setminus \{\sigma_2\}$ with $\sigma_1 \notin \mu(h)$ and $\sigma_2 \in \mu(h)$, then h prefers $\mu'(h)$ to $\mu(h)$ under $P_h^{\#}$ if and only if h prefers σ_1 to σ_2 under $P(h)$.*

In the marriage model, each individual's preferences over different matchings only depended on the spouse to who they were matched. While this is true for students in the hospital admissions models, this is not true of hospitals, which may have preferences over groups of students. For example, if a hospital is assigned their first and second choice in μ_1 and assigned their first and third choice in μ_2, then the hospital would prefer μ_1 to μ_2. This intuition is formalized in the definition of group preferences for hospitals $P_h^{\#}$.

Moreover, we require group preferences to be responsive, as defined below.

Definition 2.8 (Responsive Preferences)**.** *The preferences $P_h^{\#}$ of hospital h over sets of students is responsive if they are consistent with a complete, transitive preference relation $\succ_h$ over students and a quota q_c. Mathematically, for all $S' \subseteq S$ with $|S'| < q_h$ and any students $i, j \in S \setminus S'$, we have*

1. *$(S' \cup \{i\}) \succ_h^{\#} (S' \cup \{j\}) \Leftrightarrow i \succ_h j$*

2. *$(S' \cup \{i\}) \succ_h^{\#} S' \Leftrightarrow i \succ_h \emptyset$*

Furthermore, analogous definitions regarding stability follow in the many-to-one setting.

Definition 2.9 (Individual Rationality)**.** *A matching μ is individually rational if each student s is acceptable to their matched hospital h and vice versa.*

Definition 2.10 (Pairwise Stability)**.** *A matching μ is pairwise stable if there is no hospital-student blocking pair (h, s) with $h \in H, s \in S$ such that for some $\sigma \in \mu(h)$, $\mu(s) \neq h$, $h \succ_s \mu(s)$, and $s \succ_h \sigma$.*

However, since we are in the many-to-one setting, we can also consider the case where a matching is blocked by a coalition of multiple students and hospitals as opposed to a single hospital-student pair. Intuitively, this would mean that a set of multiple students and hospitals in A could all get an assignment preferable to their assignment in μ by matching among themselves rather than taking their assignment in μ.

Definition 2.11 (Group Stability)**.** *A matching μ is group stable if there is **no** coalition A and alternative matching μ' such that for all students s and all hospitals h in A:*

1. *$\mu'(s) \in A$ (every student in A who is matched by μ' is matched to a hospital $c \in A$)*
2. *$\mu'(s) \succ_s \mu(s)$ (every student in A prefers their new match to their old one)*
3. *$\sigma \in \mu'(h) \implies \sigma \in A \cup \mu(h)$ (every hospital in A is matched at μ' to a new student only from A, but may continue to be matched with some of its old students from $\mu(h)$)*
4. *$\mu'(h) \succ_h \mu(h)$ (every hospital in A prefers its new set of students to its old one)*

At first, the definitions of pairwise and group stability appear fundamentally different and would seem to make the computation of stable matchings in the many-to-one setting much more difficult. Whereas in the one-to-one setting, we were able to check all n^2 blocking pairs, the number of possible coalitions in the many-to-one setting is exponential. However, this seemingly intractable problem is resolved by the following theorem, which shows that the two definition of stability are in fact, equivalent. Hence, we only need to look at blocking pairs rather than blocking coalitions in our many-to-one settings, significantly simplifying the problem.

Theorem 2.12. *A matching μ is group stable if and only if it is pairwise stable.*

Proof. First, we will show if a matching μ is pairwise unstable, then it is also group unstable. If a matching μ is pairwise unstable, then there is a hospital-student blocking pair (h, s). However, this pair can also be considered a coalition A consisting of one hospital and one student, who would be better off matching with each other rather than participating in the matching μ.

Second, we will show if a matching is group unstable, then it is also pairwise unstable. Consider the coalition A and alternative matching μ' which cause matching μ to be group unstable. Let hospital h be in the coalition A. Then, the conditions of group instability tell us that $\mu'(h) \succ_h \mu(h)$. This means that there is a student $s_1 \in \mu'(h) \setminus \mu(h)$ and student $s_2 \in \mu(h) \setminus \mu'(h)$ such that $s_1 \succ_h s_2$. In words, this means that in the alternative matching μ', hospital h is matched to a more preferred student s_1, whereas in matching μ, hospital h is matched to a less preferred student s_2. By the third condition from Definition 2.11, since

s_1 was not matched to hospital h in μ originally, s_1 must be in the coalition A as well. Thus, since $s_1 \in A$ and $h \succ_{s_1} \mu(s_1)$, we know that μ is pariwise unstable by the hospital-student pair (h, s_1). □

2.2.2 Connecting Marriage Markets with Hospital Admissions

Consider the hospital admissions problem and the related marriage market where each hospital h is broken into q_h individual pieces, denoted by $h_1, \ldots, h_{q_h}$. Each of these pieces has a quota of 1 with preferences over students identical to those of the original hospital h. Given that a student's preference list once consisted of hospitals, it now is modified by replacing a hospital h with the list $h_1, \ldots, h_{q_h}$. If the preferences over individuals are strict, then there is a one-to-one correspondence between matchings in the hospital admissions problem and matchings in the newly constructed marriage market.

Theorem 2.13. *A matching of the hospital admissions problem is pairwise stable if and only if the corresponding matchings of the related marriage market are stable.*

Proof. We will first show that if a hospital-student blocking pair exists in the hospital admissions problem, then it will also cause a blocking pair to exist in the related marriage market. Let the hospital-student blocking pair be (h, s) for $h \in H, s \in S$. Then, (h_1, s) form a blocking pair as well in the related marriage market. Hence, a pairwise unstable hospital admissions problem implies an unstable marriage market.

Second, we will show that if a blocking pair exists in the related marriage market, then a hospital-student blocking pair exists in the hospital admissions problem as well. Let the blocking pair in the marriage market be (h_i, s) for some $i \in [1, q_h]$ since the student is indifferent over the hospitals $h_1, \ldots, h_{q_h}$. Then, the corresponding hospital-student pair (h, s) also blocks the matching in the hospital admissions problem. Thus, an unstable marriage market implies that the hospital admissions problem is also pairwise unstable. □

Given the above theorem, we know that the set of stable matchings for the hospital admissions problem cannot be empty. Using the description of the one-to-one deferred acceptance algorithm above, we can easily modify it to solve the hospital admissions problem as well. If the students are proposing, then the hospitals will start to reject acceptable students only when its quota q_h is full, in the same way that women would reject all but their most acceptable man in each turn of the man-proposing one-to-one algorithm. If the hospitals are proposing, then each hospital makes as many proposals at each step as they need to fulfill its quota of engagements, meaning that if a hospital currently has n engagements with $n < q_h$, then it should make $q_h - n$ proposals in the next step to have a total of q_h engagements. Thus, not only do we know that a stable matching exists in the many-to-one setting, but we can also explicitly find one such matching.

2.2.3 The NMRP Medical Matching Algorithm

Although the previous section extends the one-to-one algorithm of deferred acceptance to compute stable matchings in the many-to-one setting, the actual algorithm implemented in the medical match is slightly different. We will now describe it in detail according to the description given in [22].

Algorithm 2.14 (NMRP)**.** *Each hospital program rank orders the students who have applied to it and each student orders the hospital programs to which they have applied to, on both sides, indicating any which are unacceptable. At the central clearinghouse, any student or hospital that has been marked unacceptable will be removed off the corresponding hospital or student's list of preferences, so that the list only contains students or hospital programs which are acceptable to the other. There are two phases, a matching phase and a tentative assignment and update phase.*

In the matching phase, we first check if there are any students and hospital programs which are mutually top-ranked ($1:1$ *step). If a hospital has a quota of* q_{h_i}*, then the* q_{h_i} *highest students in its ranking are considered top-ranked. If no such matches are found, we move on to the* $2:1$ *step, where the second ranked hospital program on each student's ranking is compared with the top-ranked students on that hospital's rankings. At any point when no matches are found, the algorithm moves on to the next step such that the* $k:1$ *step of the matching phase tries to find hospital-student pairs where the student is top-ranked on the hospital ranking list and the hospital is* k*-ranked on the student ranking list. At any step where matches are found, we move on to the tentative assignment and update phase.*

In the tentative assignment and update phase after the $k:1$ *step of the matching phase, each student who is a top-ranked choice of their* k*-ranked hospital is tentatively assigned to that hospital. Then, the rankings of each student and hospital are updated as follows. Because student* s_i *is tentatively assigned to its* k*-ranked hospital, any hospital on its list below their* k*-th choice is removed. Similarly, the student* s_i *is deleted from the ranking list of any hospital that was deleted from the ranking list of* s_i*. If a hospital's top-ranked candidate is deleted, a lower-ranked student will move up to take their place since the updated ranking list has fewer students but the hospital has the same quota. Therefore, the updated rankings of each hospital now include only the applicants who haven't yet been tentatively assigned to* *a hospital they prefer. After the rankings have been updated, the algorithm goes back to the matching phase starting with the* $1:1$ *step again.*

As the algorithm moves back and forth between the two phases, any new tentative matches found for a student replace any prior tentative matches involving the same student, since a new match can only improve the student's tentative assignment. The algorithm terminates when no new tentative matches are found. At that point, all tentative matches become final and any student or hospital position which was not tentatively matched is left unassigned.

We will now prove that this algorithm also leads to stable matchings, which explains why the centralized medical matching system had such high voluntary participation rates and was such a popular alternative to individually arranging residency programs for oneself. Amazingly, this algorithm that preserves stability was developed independently of the theoretical literature of matching markets, and it was not until 1984 that the medical matching algorithm was shown to have the same theoretical properties as the deferred acceptance algorithm developed by Gale-Shapley in 1962.

Theorem 2.15. *The NMRP algorithm is a stable matching mechanism, meaning that it produces a stable matching with respect to any stated preferences.*

Proof. When the algorithm terminates, each hospital h_i is matched with the top q_i choices on its final updated rank-order list because the algorithm cannot terminate while tentative $k:1$ matches can still be found. This assignment is also stable since any student s_j who some hospital h_i originally ranked higher than one of its final assignees was deleted from h_i's ranking when s_j was given a tentative assignment higher in his or her ranking than hospital h_i. Thus, the final assignment gives student s_j a position he or she ranked higher than hospital h_i, and the final matching cannot be unstable with respect to any possible blocking pair (h_i, s_j). □

2.3 Chapter Summary

To conclude this chapter, we see that stable matchings exist for both the one-to-one and many-to-one matching settings, and similar ideas of tentative engagement or assignment of women to men or students to hospitals can be applied to construct this stable matching. This chapter provides the mathematical formalisms that allow us to precisely speak about these concepts prove the existence of and construct explicit stable matchings for these types of matching problems. As we can see from the NMRP algorithm for the medical residency match, the proof of the existence of stable matchings in every scenario is critically important to being able to use this stable matching algorithm year after year. Similarly, for other applications, such as urban high school admissions, the theoretical guarantee of the existence of at least one stable matching is crucial for school districts to be able to use this method consistently and continuously.

Chapter 3

Structures of Stable Matchings

Now that we have laid the groundwork to formalize and talk about matching problems, let us look at the structure of stable matchings in both the one-to-one and many-to-one settings. In particular, although we have shown that at least one stable matching exists, we would like to understand whether there are multiple stable matchings possible and how to compare these matchings. In addition, we hope to understand how the set of stable matchings compares on metrics like Pareto optimality to the general set of possible matchings.

3.1 Structures of One-to-One Matchings

In an above subsection, we demonstrated one example of deferred acceptance. Now, let us provide a second example in order to understand the types of stable matchings that exist.

3.1.1 Deferred Acceptance Example, Revisited

In the description of the deferred acceptance algorithm, we have men propose, but men and women play symmetrical roles in the marriage market, so we can consider the matching that occurs when women propose. Let us call the matching from Section 2.1.2 μ_M to denote the fact that men propose.

Now, let us now consider the matching μ_W, which results if the women propose instead of the men, while keeping the same preferences as the example in Section 2.1.2. Given that the first-choice preferences of each woman is different, we only need to complete one step of the deferred acceptance algorithm to reach a stable matching, namely that μ_W consists of the marriages (w_1, m_3), (w_2, m_2), and (w_3, m_1).

Note that all women clearly prefer μ_W to μ_M, as in μ_W, each woman marries their first choice man, whereas in μ_M, each woman marries either their second or last choice man. Similarly, note that all men prefer μ_M to μ_W. In μ_M, each man marries either their first or second choice woman, whereas in μ_W, each man marries their last choice woman.

This example elucidates three points. Firstly, there may be more than one stable matching, and in this section, we find two distinct stable matchings by switching which side of the market gets to propose in the deferred acceptance algorithm. Secondly, one stable matching may be unilaterally preferred by one side of the market compared to another stable matching. Lastly, we see that the side of the market that proposes gets more favorable outcomes in their matches.

3.1.2 Comparing Stable Matchings

To formalize our comparison of different stable matchings, let $\mu \succ_M \mu'$ denote that all men like matching μ at least as much as μ', with at least one man preferring μ to μ' outright. Mathematically, we would say that $\mu(m) \succeq_m \mu'(m)$ for all $m \in M$, and $\mu(m) \succ_m \mu'(m)$ for at least one man $m \in M$. Similarly, $\mu \succeq_M \mu'$ implies that either $\mu \succ_M \mu'$ or all men are indifferent between the two matchings. Unlike an individual's preference relation $\succ_i$ for $i \in M \cup W$, the preference relation $\succ_M$, representing the common preferences of all men, is a partial ordering. This is because it is possible for some men to prefer μ and others to prefer μ', so there is no common agreement about which matching is preferred by all. We can similarly define $\succ_W$ and $\succeq_W$ for women, and like individual preferences, these common preferences are also transitive.

Definition 3.1 (M-Optimal). *A stable matching μ is M-optimal if every man likes it at least as well as any other stable matching, that is, if for every other stable matching μ', $\mu \succ_M \mu'$.*

Definition 3.2 (W-Optimal). *A stable matching ν is W-optimal if every woman likes it at least as well as any other stable matching, that is, if for every other stable matching ν', $\nu \succ_W \nu'$.*

3.1.3 Optimal Stable Matchings

Definition 3.3 (Achievable)**.** *A woman w and a man m are achievable for each other in a marrige market if m and w are paired in some stable matching.*

Based on this definition, if men and women all have strict preferences in our marriage market, each individual has only one favorite match among their set of achievable spouses. Thus, a M-optimal stable matching would naturally match each man to his most preferred achievable woman and similarly, a W-optimal stable matching must match each woman to her most preferred achievable man. This means there is at most one M-optimal stable matching and one W-optimal stable matching, and we will prove that these matchings do, in fact, exist.

Theorem 3.4. *When all men and women have strict preferences, there always exists an M-optimal stable matching and a W-optimal stable matching. The M-optimal stable matching μ_M is produced by deferred acceptance with men proposing and the W-optimal stable matching μ_W is produced by deferred acceptance with women proposing.*

Proof. We seek to show using induction that when men propose using deferred acceptance, no man is ever rejected by an achievable woman. Thus, the stable matching μ_M that is produced must match each man to his most preferred achievable woman and is the unique M-optimal stable matching.

First, assume that up to step $n-1$ in the deferred acceptance procedure, no man has been rejected yet by any woman who is achievable for him. In step n, suppose that woman w rejects man m. We must show that w is not achievable for m here as well by considering two cases.

Case 1: If woman w rejects man m as unacceptable because she would prefer to be single rather than marry man m, then woman w is unachievable for man m and we are done.

Case 2: If woman w rejects man m in favor of man m', to whom she stays engaged to, then she prefers m' to m. The deferred acceptance procedure tells us that m' must prefer w to any woman except for those who have already rejected him in a previous step, who by the inductive assumption, are unachievable for him. Consider for contradiction a matching μ that matches m to w and everyone else to an achievable mate. Then, m' prefers woman w to his current match $\mu(m')$. Thus, (m', w) form a blocking pair because w prefers m' to m and m' prefers w to $\mu(m')$, implying the hypothetical matching μ is unstable. Thus, there are no stable matchings which pair m and w, so they are unachievable for each other and prove the inductive step. □

Given this theorem, agents on one side of the market are in agreement on the best stable matching when preferences are strict, and thus, have a common interest regarding the set of stable matchings. In fact, it turns out that agents on opposite sides of the markets have opposite interests in this regard, and the optimal stable matching for one side of the market is the worst stable matching for agents on the other side of the market. Moreover, any stable matching that is better for all men is worse for all women, and vice versa.

Lemma 3.5. *When all agents have strict preferences, if μ and μ' are stable matchings, then all men like μ at least as well as μ' if and only if all women like μ' at least as well as μ. More concisely, $\mu \succ_M \mu' \iff \mu' \succ_W \mu$.*

Theorem 3.6. *When all agents have strict preferences, the M-optimal stable matching is the worst stable matching for the women and it matches each woman with her least preferred achievable mate. Similarly, the W-optimal stable matching matches each man with his least preferred achievable mate.*

Proof. Let μ, μ' be stable matchings and assume that $\mu \succ_M \mu'$. Assume for contradiction that we do not have $\mu' \succ_W \mu$. Then there must exist some woman w who prefers μ to μ', meaning $\mu(w) \succ_w \mu'(w)$. Because individuals have strict preferences and at least one woman has a different mate in μ than in μ' because at least one man does and the marriage are bilateral, we see that woman w must have a different spouse in μ than in μ'. Thus, man $m = \mu(w)$ must also have a different spouse in μ than in μ'. Since all stable matchings are individually rational, the fact that w prefers $\mu(w)$ to $\mu'(w)$ implies w is not single at μ. Since man m also has strict preferences, then (m, w) forms a blocking pair for the matching μ', which contradicts the original assumption that mu' is stable. Thus, $\mu' \succ_W \mu$ as desired. □

While we have discussed the M and W-optimal stable matchings at length, the set of stable matchings is not limited to just these. In fact, there can exist a much larger set of stable matchings which vary in optimality for the two sides of the market.

3.1.4 Lattice Structure of Stable Matchings

Definition 3.7 (Lattice). *A lattice is a partially ordered set L for which any two of its elements a, b have a supremum, denoted $a \vee b$, and an infimum, denoted by $a \wedge b$.*

Definition 3.8 (Distributive Lattice). *A lattice is distributive if and only if for all $a, b, c \in L$*

the two following conditions hold

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$
$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Using these definitions of algebraic lattices, we provide three theorems linking the structure of stable matchings to lattice theory.

Theorem 3.9. *When all preferences are strict, if μ, μ' are stable matchings, then the functions $\lambda = \mu \vee_M \mu'$ and $\nu = \mu \wedge_M \mu'$ are both stable matchings [13].*

Proof. We need to show that both λ and ν are stable matchings. First, we show λ is a matching by showing that $\lambda(m) = w$ if and only if $\lambda(w) = m$. Because both μ, μ' are stable, we know that $\lambda(m) = w$ implies $\lambda(w) = m$.

In the other direction, let us define the set

$$M' = \{m \text{ such that } \lambda(m) \text{ is in } W\} = \{m \text{ such that } \mu(m) \text{ or } \mu'(m) \text{ is in } W\}$$

Similarly, define the set

$$W' = \{w \text{ such that } \lambda(w) \text{ is in } M\} = \{w \text{ such that } \mu(w) \text{ or } \mu'(w) \text{ is in } M\}$$

Using the direction proved above, we know that $\lambda(M')$ is contained in W'. However, we also know that the size of $\lambda(M')$ is the same as M' because $\lambda(m) = \lambda(m') = w$ only if $m = m' = \lambda(w)$. Furthermore, $\lambda(M')$ is at least as large as $\mu(W')$, so $\lambda(M')$ and W' are the same size and $\lambda(M') = W'$. Thus, for $w \in W'$, $\lambda(w) = m$ for some $m \in M'$ so $\lambda(w) = m$. If $w \notin W'$ then $\lambda(w) = w$. This means $\lambda(w) = m$ implies $\lambda(m) = w$. This shows that λ is a matching.

Now, we will prove that λ is stable. Suppose for contradiction that (m, w) is blocking λ. Then

$$w \succ_m \lambda(m) \implies w \succ_m \mu(m) \implies w \succ_m \mu'(m)$$

On the other hand, we also have $m \succ_w \lambda(w)$. Thus, (m, w) blocks the matching μ if $\lambda(w) = \mu(w)$ and blocks the matching μ' if $\lambda(w) = \mu'(w)$. In both cases, we get a contradiction, since both μ, μ' are stable and cannot have any blocking pairs. Thus, λ is a stable matching.

By a symmetric argument, we can show that ν is also a stable matching. □

Corollary 3.10. *When preferences are strict, the set of stable matchings is a distributive lattice under the common order of men dual to the common order of women.*

Proof. This corollary follows directly from Theorem 3.9. □

Mathematically, we can also see that both Theorem 3.9 and Corollary 3.10 are direct results of Tarski's fixed point theorem [7, 11].

Not only do we see structure at the macro level, using the lattice theory language of meets and joins to understand the set of stable matchings, but there is also structure at the micro level in the set of individual agents involved in every stable matching.

Theorem 3.11 (Lone Wolf Theorem)**.** *In a marriage market with strict preferences, the set of people who are single is the same for all stable matchings [18].*

3.1.5 Pareto Optimality

While we have seen that the M-optimal stable matching is the best stable match men can achieve, a stronger characteristic of matchings that we might care about is Pareto optimality, which measures efficiency.

Definition 3.12 (Weak Pareto Optimality)**.** *A matching μ is weakly Pareto optimal for men if there are no alternative matchings that would would be* strictly preferred *by* all *men [21].*

Definition 3.13 (Strong Pareto Optimality)**.** *A matching μ is strongly Pareto optimal for men if there are no alternative matchings that would would be strictly preferred by some men and weakly preferred by all men. This means that the matching is liked at least as well by all men and preferred by some men.*

In particular, we are interested to see if there is an unstable matching that all men would prefer, indicating that the men are paying a price for stability. However, we see that this is not true.

Theorem 3.14. *There is no individually rational matching μ (stable or not) such that $\mu \succ_m \mu_M$for all $m \in M$. This means μ_M is weakly Pareto optimal for all men M.*

Proof. Let us consider the man-proposing deferred acceptance algorithm. Assume for contradiction that a better matching μ existed. Then, it would match every man m to some woman w who had rejected him in the algorithm in favor of some other man m'. This means that that set of women, $\mu(M)$, would have also been matched under μ_M, more formally written as $\mu_M(\mu(M)) = M$. However, since all men in M are matched under the M-optimal matching μ_M, then any woman who got a proposal in the last step of the algorithm has not rejected any acceptable man, since the algorithm stops as soon as every woman in $\mu_M(M)$ holds an acceptable proposal. Since every man prefers μ to μ_M, this would mean that such a woman would have had to be single at μ, which contradicts the fact that $\mu_M(M) = \mu(M)$. □

Through the below example, we see that the M-optimal matching, while weakly Pareto optimal, is not strongly Pareto optimal.

Consider the following preferences for a set of three men and three women.

$$\begin{aligned} P(m_1) &= w_2 \succ w_1 \succ w_3 & P(w_1) &= m_1 \succ m_2 \succ m_3 \\ P(m_2) &= w_1 \succ w_2 \succ w_3 & P(w_2) &= m_3 \succ m_1 \succ m_2 \\ P(m_3) &= w_1 \succ w_2 \succ w_3 & P(w_3) &= m_1 \succ m_2 \succ m_3 \end{aligned}$$

Then using the man-proposing deferred acceptance algorithm, we get than the M-optimal matching is

$$\mu_M(m_1) = w_1 \quad \mu_M(m_2) = w_3 \quad \mu_M(m_3) = w_2$$

However, the below matching

$$\mu(m_1) = w_2 \quad \mu(m_2) = w_3 \quad \mu(m_3) = w_1$$

leaves m_2 with the same match as he had under μ_M, but improves the matches of m_1 and m_3. Thus, the M-optimal matching is not strongly Pareto optimal, and there may be some non-stable matchings which are liked at least well as μ_M by all men and preferred by some men.

3.2 Structures of Many-to-One Matchings

In this section, we will see that the results of optimal stable matchings extend from the one-to-one to the many-to-one setting, wherease the Pareto-optimality conditions no longer hold true when looking at the many-to-one market.

3.2.1 Optimal Stable Matchings

In Theorem 2.15, we proved that the NMRP algorithm gives a stable matching. Extending the one-to-one definition, a hospital H and student s are achievable if there is some stable matching in which they are matched. For each hospital H_i with quota q_i, we can define a_i to be the number of achievable students, and let $k_i = \min(a_i, q_i)$. Thus, for any set of preferences submitted by hospitals over students, the NMRP algorithm produces a matching that is hospital-optimal, which we will now prove.

Theorem 3.15. *For any submitted list of strict preferences over individuals, the NMRP algorithm produces a matching that gives each hospital H_i its k_i highest ranked achievable students.*

Proof. In order to show that the final assignment gives each hospital its top q_i achievable students, whether it has many achievable students when $a_i > q_i$ or all its achievable students if $a_i < q_i$, it is equivalent to show that no achievable student is ever deleted from a hospital's rank-order list. We will proceed by induction.

Suppose that up to the r-th iteration of the algorithm, no student has been deleted from the ranking of a hospital for whom they are achievable and that on the $(r+1)$-th iteration, student s_j is tentatively matched with hospital H_i and deleted from the ranking of hospital H_k. Then, any assignment that matches s_j with H_k and assigns achievable matches to H_i must be unstable for two reasons. First, $H_i \succ_{s_j} H_k$. Second, because s_j was top-ranked by H_i at the end of the r-th iteration when no achievable students had yet been deleted from H_i's rank-order list, s_j must be ranked higher by H_i than one of its assignees. Therefore, s_j is not achievable for H_k. Therefore, no achievable student will ever deleted from a hospital's rank order list and the final assignment must give each hospital its top q_i achievable students. □

Corollary 3.16. *When preferences are strict, there exists a hospital-optimal stable matching that every hospital likes as well as any other matching, and a student-optimal stable matching that every student likes as well as any other stable matching.*

Proof. The existence of the hospital-optimal stable matching follows from the preceding theorem. The existence of the the student-optimal stable matching when preferences are strict follows directly from the one-to-one setting. □

Since the matching found is the unique hospital-optimal stable matching, hospitals ought to prefer the matching from the NMRP algorithm to any other stable matching possible given the preference lists. In fact, this result is stronger than the one-to-one setting. Whereas in the M-optimal matching, the men receive their most preferred achievable mate, in the many-to-one setting, each hospital is able to receive its k_i most preferred achievable students.

3.2.2 Geographic Constraints

At the end of Section 3.1.4, we see that the set of single individuals is the same in every stable matching. This result extends to the many-to-one setting as follows.

Theorem 3.17. *When all preferences over individuals are strict, the set of students admitted and seats filled in the hospital-intern problem is the same at every stable matching [25].*

Proof. This proof follows directly from the one-to-one result (Theorem 3.11) and the equivalence of the hospital-intern problem with the related marriage markets problem (Theorem 2.13). □

Unlike the one-to-one setting, this has direct implications in the many-to-one setting of the hospital-intern market. Using the NMRP algorithm, it was noticed that many rural hospitals have difficulty filling positions and are largely filled with foreign students. In particular, this has led to concerns about the quality of medical care in rural communities. Given this problem, it has been proposed to alter the NMRP algorithm so as to equalize the distribution of interns across urban and rural hospitals. However, it is highly difficult to modify the algorithm to more fairly re-distribute interns across US hospitals while still maintaining the stability constraint. In fact, Theorem 3.17 demonstrates that this goal is impossible, due to the structural limitations of who gets assigned in every stable matching.

3.2.3 Pareto Optimality

In contrast with the Pareto optimality results from the one-to-one matching setting, similar results do not hold for the hospital-intern model.

Theorem 3.18. *When preferences over individuals are strict, the student-optimal stable matching is weakly Pareto optimal for students, but the hospital-optimal stable matching may not be weakly Pareto optimal for hospitals.*

Proof. The first part can be proved by looking at the weak Pareto optimality result from one-to-one matching in Theorem 3.14 and noting that the set of stable matchings from the hospital-intern model and the related marriage market problem are equivalent.

The second part can be shown via counterexample. Consider the set of hospitals $\{h_1, h_2, h_3\}$ and set of students $\{s_1, s_2, s_3, s_4\}$, with $q_{h_1} = 2$ and $q_{h_2} = q_{h_3} = 1$. The preferences are given below.

$$\begin{aligned} P(s_1) &= h_3 \succ h_1 \succ h_2 & P(h_1) &= s_1 \succ s_2 \succ s_3 \succ s_4 \\ P(s_2) &= h_2 \succ h_1 \succ h_3 & P(h_2) &= s_1 \succ s_2 \succ s_3 \succ s_4 \\ P(s_3) &= h_1 \succ h_3 \succ h_2 & P(h_3) &= s_3 \succ s_1 \succ s_2 \succ s_4 \\ P(s_4) &= h_1 \succ h_2 \succ h_3 \end{aligned}$$

The only stable matching that is hospital-optimal is μ_H given by

$$\mu_H(h_1) = s_3, s_4 \quad \mu_H(h_2) = s_2 \quad \mu_H(h_3) = s_1$$

However, the matching μ given by

$$\mu(h_1) = s_2, s_4 \quad \mu(h_2) = s_1 \quad \mu(h_3) = s_3$$

gives both hospitals h_2, h_3 their first choice student over their second choice student, so they both prefer μ to μ_H. Since h_1 gets their second and fourth choice students over their third and fourth choice students, h_1 also prefers μ to μ_H, meaning every hospital prefers μ to μ_H and the hospital-optimal stable matching may not be even weakly Pareto optimal. Note that μ is not a stable matching because (h_1, s_1) are a hospital-student blocking pair, as h_1 would prefer to be matched to s_1, s_4 over s_2, s_4, and s_1 prefers h_1 to h_2. □

3.3 Chapter Summary

In Chapter 2, we saw direct analogies between the one-to-one and many-to-one settings in the existence of stable matchings through constructive algorithms. However, in this chapter, we see that the results from one-to-one matching do not always transfer to the many-to-one matching setting. While there are direct connections between the set of optimal stable matchings as well as the set of individuals matched in each stable matching, characteristics of efficiency such as weak Pareto optimality do not hold in the many-to-one setting as they do in the one-to-one setting. Thus, when implementing matching algorithms in practice, especially in many-to-one settings, the efficiency-stability tradeoff must be considered. Additionally, we see that the lattice structure of the set of stable matchings implies that any implementation of two-sided matching algorithms in the real-world will favor one side of the market. Hence, the choice of who gets to propose in matching mechanisms such as deferred acceptance remains in question for any policymaker seeking to implement them.

Chapter 4

Strategic Incentives for Individual Agents

Above, we looked at the structure of the matchings we might observe, and crucially, assumed that each individual submitted truthful preferences to the centralized algorithm. However, we can also ask how we might expect individual agents in these markets to behave and whether they ought to be truthful in their stated preferences as previously assumed. Given that preferences are actually private information, each individual is capable of submitting non-truthful preferences to the clearinghouse in what shall henceforth be known as *strategic behavior*.

4.1 One-to-One Questions

4.1.1 Example of Strategic Behavior

Consider the following set of preferences for a market with two men and two women.

$$\begin{aligned} P(m_1) &= w_1 \succ w_2 \succ m_1 \qquad & P(w_1) &= m_2 \succ m_1 \succ w_1 \\ P(m_2) &= w_2 \succ w_1 \succ m_2 \qquad & P(w_2) &= m_1 \succ m_2 \succ w_2 \end{aligned}$$

Using the deferred acceptance algorithm when men propose, the M-optimal stable matching would be

$$(m_1, w_1), (m_2, w_2)$$

However, consider the following misreport of w_2's preferences, where $P(w_2) = m_1 \succ w_2 \succ m_2$, meaning w_2 finds m_2 unacceptable. In this case, the new stable matching produced would be

$$(m_1, w_2), (m_2, w_1)$$

By truncating and misreporting her preferences, w_2 is able to force a better stable matching for herself, since she prefers being matched to m_1 than m_2.

To formalize this notion of truthful reporting, we define a concept known as dominant strategy incentive compatibility.

Definition 4.1 (Dominant Strategy Incentive Compatibility). *A mechanism is dominant strategy incentive compatible (DSIC) if truth-telling is a weakly dominant strategy, meaning that every individual fares best, or at least not worse, regardless of what other agents do.*

The notion of DSIC is the strongest notion of incentive compatibility, the idea that every participant can achieve their best outcome by acting truthfully. DSIC mechanisms are also called *truthful* or *strategyproof*, because strategic considerations, such as preference misrepresentation, would not help any agent achieve a better outcome. Thus, we argue that the stable matching mechanism is *not* strategyproof. By definition, this means that there is some incentive for some individual in the market to misreport their preferences to the clearinghouse, because they will receive a better matching than had they submitted their preferences truthfully.

This example naturally leads to the following theorem.

Theorem 4.2 (Impossibility Theorem). *No stable matching mechanism exists for which stating the true preferences is a dominant strategy for all agents. Alternatively, no stable matching mechanism is also DSIC.*

Proof. In order to prove this, we need to demonstrate a particular marriage market such that for any stable matching mechanism, truth telling is not a dominant strategy for all agents. We are able to demonstrate this for a marriage market with two men and two women and since this smaller market can be embedded in any large marriage market without changing their preferences, the result follows for all marriage markets.

Consider the same set of preferences as before, with

$$\begin{aligned} P(m_1) &= w_1 \succ w_2 \succ m_1 \qquad & P(w_1) &= m_2 \succ m_1 \succ w_1 \\ P(m_2) &= w_2 \succ w_1 \succ m_2 \qquad & P(w_2) &= m_1 \succ m_2 \succ w_2 \end{aligned}$$

There are two stable matchings, μ, ν with $\mu(m_i) = w_i$ for $i \in \{1, 2\}$ and and $\nu(m_i) = w_j$ for $i, j \in \{1, 2\}, j \neq i$. Therefore, any stable matching mechanism would result in either

μ, ν given the individuals' stated preferences. In the above example, we show a profitable manipulation for w_2. In the symmetric case for ν, there is a profitable manipulation for m_2 with the misreported preferences $P(m_2) = w_2 \succ m_2 \succ w_1$ which leads to a more profitable stable matching for m_2, where he is matched to w_2. □

Furthermore, we see that this tension between incentive compatibility and stability is pervasive. We might hope that certain stable matching mechanisms are better than others, rarely giving individuals such strategic incentives to misreport preferences. However, the following theorem shows that one agent usually has an incentive to strategically deviate.

Theorem 4.3. *When any stable mechanism is applied to a marriage market in which preferences are strict and there is more than one stable matching, then at least one agent can profitably misreport their preferences, assuming that the others tell the truth. Moreover, this agent can misreport their preferences so that they are matched to their most preferred achievable mate under the true preferences at every stable matching.*

Proof. From our proof statement, we know $\mu_M \neq \mu_W$. Suppose that when all agents state their true preferences, the mechanism results in a stable matching $\mu \neq \mu_W$. Let w be any woman such that $\mu_W(w) \succ_w \mu(w)$, and now, let w misreport her preferences by stating that she would prefer to be single rather than marry any man who ranks below $\mu_W(w)$ on her true preference list.

The matching μ_W will still be stable under these altered P' since there are now fewer possible blocking pairs. Given that the set of single individuals is the same for all stable matchings, we see that w is not single in μ' either and hence is matched with someone she likes at least as much as $\mu_W(w)$, since all other less acceptable men have been removed from her list. Since our new matching is also stable under the original preferences, we actually have that $\mu'(w) = \mu_W(w)$, which is preferred by w to her original matching $\mu(w)$. Thus, $\mu' \succ_w \mu$, and a symmetric argument can be made for any man m who strictly prefers μ_M. □

While the above theorem might seem concerning, we also see that there are limits to how far an arbitrary stable matching mechanism can be manipulated.

4.1.2 Limits on Strategic Manipulation

Lemma 4.4 (Blocking Lemma)**.** *Let μ be any individually rational matching with respect to strict preferences P, and let M' be all men who prefer μ to μ_M. If M' is nonempty, there is a pair (m, w) that blocks μ such that $m \in M \setminus M'$ and $w \in \mu(M')$ [9].*

Proof. We can consider this in two cases.

Case 1: $\mu(M') \neq \mu_M(M')$. Choose $w \in \mu(M') \setminus \mu_M(M')$, for example, $w = \mu(m')$. Then, m' prefers w to $\mu_M(m')$ so w prefers $\mu_M(w) = m$ to m'. But m is not in M' since w is not in $\mu_M(M')$, so m prefers w to $\mu(m)$. Thus, (m, w) blocks μ.

Case 2: $\mu_M(M') = \mu(M') = W'$. Let w be the last woman in W' to receive a proposal from an acceptable member of M' in the deferred acceptance algorithm. Since all $w \in W'$ have rejected acceptable men from M', w had some man m engaged when she received this last proposal. Then, (m, w) is the desired blocking pair for the following reason. First, m is not in M' because if so, after having been rejected by w, he would have proposed again to another member of W', contradicting the fact that w received the last proposal. But m prefers w to his mate under μ_M and since he is no better off under μ, he prefers w to $\mu(m)$. On the other hand, m was the last man to be rejected by w so she must have rejected her mate under μ before she rejected m and hence she prefers m to $\mu(w)$, so (m, w) blocks μ as claimed. □

Theorem 4.5 (Limits on Successful Manipulation)**.** *Let P be the true preferences (not necessarily strict) of the agents, and let P' differ from P in that some coalition C of at least two agents have misreported their preferences. Then, there is no matching μ, stable for P', which is preferred to* every *stable matching under the true preferences P by all members of the coalition C.*

Proof. Consider for contradiction that some nonempty subset $M' \cup W'$ of men and women misreport their preferences and are strictly better off under some matching μ, stable under P', than under any stable matching under P. This would imply that

$$\mu(m) \succ_m \mu_M(m) \quad \text{for every } m \in M'$$
$$\mu(w) \succ_w \mu_W(w) \quad \text{for every } w \in W'$$

The above relations imply that μ must be individually rational, and by applying the blocking lemma, we are done. □

Thus, while truth-telling may not be a dominant strategy for every agent in the market, only individuals, not coalitions, can successfuly manipulate the market by mis-reporting their preferences to improve their match.

4.1.3 Strategic Incentives in Optimal Stable Matchings

Furthermore, the strategic incentives of individual agents is also linked to the structure of the set of stable matchings. More specifically, truth-telling is a dominant strategy for the agents on the proposing side of the market, whereas there are strategic incentives for agents on the other side of the market to misrepresent their preferences.

Theorem 4.6. *The mechanism that yields the M-optimal stable matching (in terms of stated preferences) makes it a dominant strategy for each man to state his true preferences.*

Proof. This follows directly from the preceding theorem if we take the coalition C to be the set of all men. □

Corollary 4.7. *When preferences are strict and the M-optimal stable mechanism is employed, there will be an incentive for some woman to misrepresent her preferences whenever more than one stable matching exists.*

Proof. This corollary follows directly from Theorem 4.5, Theorem 4.4 and Theorem 4.2. □

4.2 Many-to-One Questions

In the one-to-one setting, we saw that truth-telling was a dominant strategy for the proposing side of the market. However, we see that this is not necessarily true in the many-to-one setting.

Theorem 4.8. *In the hospital-intern problem, no stable matching mechanism exists that makes it a dominant strategy for all hospitals to state their true preferences.*

Proof. Since a stable matching mechanism that gives all hospitals a dominant strategy would do so for all examples, it suffices to show for one example that no such stable matching mechanism exists.

Consider the example used to prove Pareto optimality from before, with the set of hospitals $\{h_1, h_2, h_3\}$ and set of students $\{s_1, s_2, s_3, s_4\}$, with $q_{h_1} = 2$ and $q_{h_2} = q_{h_3} = 1$. The preferences are given below.

$$\begin{aligned}
&P(s_1) = h_3 \succ h_1 \succ h_2 \quad && P(h_1) = s_1 \succ s_2 \succ s_3 \succ s_4\\
&P(s_2) = h_2 \succ h_1 \succ h_3 \quad && P(h_2) = s_1 \succ s_2 \succ s_3 \succ s_4\\
&P(s_3) = h_1 \succ h_3 \succ h_2 \quad && P(h_3) = s_3 \succ s_1 \succ s_2 \succ s_4\\
&P(s_4) = h_1 \succ h_2 \succ h_3 &&
\end{aligned}$$

The hospital-optimal outcome gives us

$$\mu(h_1) = \{s_3, s_4\},\ \mu(h_2) = \{s_2\},\ \mu(h_3) = \{s_1\}$$

Moreover, this matching μ is the unique stable matching, so that $\mu = \mu_H = \mu_S$, and any stable matching procedure must select this outcome.

Now, consider if hospital h_1 were to mis-state its preferences and give

$$P'(h_1) = s_1 \succ s_4 \succ h_1,$$

meaning that hospital h_1 now finds students s_2, s_3 unacceptable. This alternative set of preferences also produces an unique stable matching, given by

$$\mu'(h_1) = \{s_1, s_4\}\ \mu'(h_2) = \{s_2\}\ \mu'(h_3) = \{s_3\}$$

Since $\mu'(h_1) \succ_{h_1} \mu(h_1)$ because it would prefer its first and fourth choice as opposed to its third and fourth choice, hospital h_1 does better by mis-stating its preferences rather than telling the truth. This completes the proof, because in this case, no stable matching mechanism can make it a dominant strategy for h_1 to state their true preferences. □

This theorem leads to further corollaries which show that the one-to-one matching results of Lemma 4.4 and Theorem 4.5 do not extend to the many-to-one setting.

Corollary 4.9. *In the hospital-intern problem, a coalition of agents, or even a single agent, may be able to misrepresent their preferences so that they do better than at any stable matching under true preferences.*

Despite the fact that Theorem 4.8 tells us that no stable matching mechanism gives hospitals a dominant strategy, the student situation mirrors that of the one-to-one marriage market.

Theorem 4.10. *A stable matching procedure that yields the student-optimal stable matching makes it a dominant strategy for all students to state their true preferences.*

Proof. If the theorem were false, then there would be a hospital-student matching market in which a student could profitably misreport their preferences. Given that the hospital-intern problem can be converted to a marriage market problem, this would also mean that they can profitably misreport their preferences in the corresponding marriage problem. However, we know this is not possible in the one-to-one scenario from Theorem 4.6. □

4.3 Chapter Summary

In Chapter 4, we show that individual agents in the one-to-one setting and coalitions of agents in the many-to-one setting have strategic incentives to mis-report their preferences. Extending our conclusions from the last chapter, *both* the strategic results of this chapter and the structure of the set of stable matchings inherently imply that one side of the market is favored. Given that only one side of the market has incentive to truthfully report their preferences, to decide which side of the market gets to propose in the real-world is a policy decision.

For example, in Boston public schools, it is well documented that wealthier parents better understand how to strategically mis-report their children's school preferences in comparison to poorer parents, and as a result, had a higher probability of getting their children into better schools [19]. To ameliorate this educational inequity, when deferred acceptance was implemented in Boston public schools, it was decided that students should propose so that the incentive to misreport preferences was removed for all parents. In addition, it was assumed that there would be little incentive for schools to mis-report their preferences, as the majority of schools desire similar sets of characteristics, from good test scores to a diverse student body.

Another example of this is the switch in the medical match from the hospital-proposing to the doctor-proposing stable matching mechanism [27]. Given that only one side of the market has incentive to truthfully report their preferences, the switch to the doctor-proposing mechanism was driven in large part because of a desire to make the problem strategically simple for doctors, as well as a recognition that there was no way to make the mechanism strategyproof for hospitals.

Thus, when considering the implementation of matching algorithms in real-world settings, it is absolutely necessary to consider these theoretical results demonstrating an incentive to strategic manipulate stable matching mechanisms in order to understand how real-world agents might act in such a market.

Part II

Deep Learning for Matching Problems

Chapter 5

Deep Learning for Economic Theory

In this chapter, we will first motivate the use of deep learning methods to approach economic theory problems such as stable matching. Next, we will provide a background in the machine learning methods that are used for the experimental results. Lastly, we will introduce the notion of randomized matching as a more general form of one-to-one matching and discuss the necessity of re-framing the stable matching problem as a randomized matching problem when applying deep learning methods.

5.1 Why Deep Learning

Despite significant progress in economic theory over the past few decades, there are still many fundamental theoretical questions which remain unanswered. A long-standing one is the problem of designing an auction that maximizes expected revenue, yet only the case where a single-item is up for auction is fully understood. Motivated by the ability of neural networks to automatically pick up the relevant features in the underlying data and recent theoretical results on the ability of neural networks to find globally optimal solutions for non-concave/non-convex objective functions, Dütting et al. [6] demonstrate how to cast the problem of designing an (essentially) revenue-optimal, DSIC mechanism as a supervised machine learning problem. By doing so, they are able to not only replicate prior theoretical and analytical results in the field of auction theory, but are also able to design essentially optimal auctions for more poorly understand auction design problems as well as for combinatorial settings.

Whereas in auctions, bidders have valuations over items but not vice versa, in matching markets, we have two-sided preferences. However, the setup of these two types of economic problems are still remarkably similar. Given the success of this deep learning framework in aiding the understanding of problems within auction theory, we hoped similar deep learning methods could also be applied successfully to problems in the study of matching markets. In particular, we hope to address two questions with this methodology.

1. Can deep learning methods replicate theoretical results like Gale-Shapley [8] and given a preference list for every agent in the market, find a stable matching?
2. Theorem 4.2 tells us that no stable matching mechanism is also DSIC. Can deep learning methods be used to create an approximately stable, DSIC mechanism that help us better understand and quantify the tradeoffs between these two properties?

5.2 Fundamentals of Deep Learning

Before beginning to discuss deep learning methods and neural networks in more detail, I would like to first take a step back and note that deep learning methods, while used extensively throughout many fields of research and in practice, are not yet well understood in a rigorous, mathematical way. Originally, neural networks were designed to work like biological neurons in our brain. If we see a green traffic light when driving, that input would travel through our retina to the brain, where various neurons would fire in succession, processing this data and then translating that into an action, like stepping on the gas pedal. The field of computational learning theory continues to work on discovering the theoretical underpinnings that make machine learning methods like deep learning work, but for the purpose of this thesis, we will only describe at a high level what steps our deep learning algorithm is taking, without necessarily providing mathematical justifications.

5.2.1 A Deep Learning Framework

At the most general level, deep learning is one method used in the broader field of machine learning that takes in a set of inputs X and uses those inputs to predict an output Y [17]. Given pairs of inputs and outputs (the training set), a deep learning algorithm will try to learn associations and patterns that allow it accurately map inputs to outputs in the training set. Ideally, this mapping would also allow us to more accurately predict outputs on future, previously unseen inputs (test set). Consider the classical example of image classification. Inputs to this algorithm would be images of either cats or dogs, and the corresponding outputs would be labels for what animal is depicted in each image. Then, the deep learning algorithm will learn features that can be used to distinguish these two types of images, such as sharp teeth, fur patterns, etc.

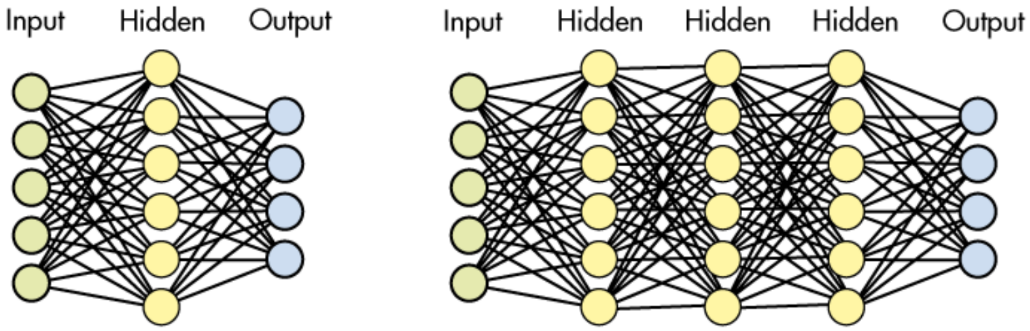


Figure 5.1: Neural Network vs. Deep Neural Network

In order to learn these features, deep learning algorithms use a data structure known as a neural network (Figure 5.1), which is composed of an input layer, an output layer, and hidden layers in between. Input layers take in a numerical representation of the data, output layers output the prediction, and hidden layers perform the computations in between. As seen in Figure 5.2, each node receives input from either other nodes or an external source, and computes the sum of a linear combination of these inputs through the weights **w** and bias b, denoting the relative importance of each input. After computing this sum, a non-linear activation function such as tanh, sigmoid, etc. is applied, which acts as a *threshold* determining whether a node fires and triggers the following connected nodes.

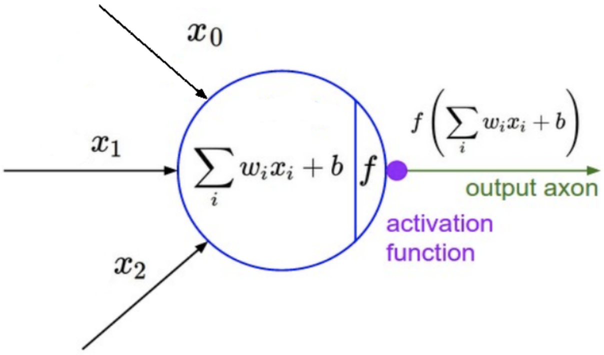


Figure 5.2: Neural Net Node

We call the neural network *fully connected* if every node in each layer is connected to every other node in the following layer, as depicted in Figure 5.1. We also call this process 'deep' learning because as we increase the ‘depth’ of the neural network with more hidden layers, the neural network improves its ability to approximate more complex functions and learn more subtle features.

After the neural network converts every input to an output through the *feed forward* process, we evaluate how good the predicted outputs are relative to the expected outputs through the *loss function* $\mathcal{L}$. The ultimate goal of our deep learning algorithm is to minimize this loss by adjusting the weights and biases at each node of the neural network through *back propagation* via *gradient descent*. This process backtracks through every layer of the neural network and updates the weights and biases of each node by iteratively moving in the direction of steepest descent (the negative gradient) in order to find the minimum of this loss function at the ‘bottom of the valley’. A *learning rate* parameter, which we can choose, determines how far we travel in the direction of steepest descent, which acts as a multiplier for the gradient vector. Thus, through the deep learning algorithm, we are able to iteratively update our neural network such that it becomes an increasingly accurate predictor of correct outputs given new inputs.

5.2.2 Augmented Lagrangian Method

Whereas in the general description of deep learning algorithms, we seek to solve an unconstrained optimization problem, in our experimental setup, we need to solve a constrained optimization problem. To do so, we use the augmented Lagrangian method, which solves problems of the form

$$\min f(x) \text{ s.t. } c_i(x) = 0 \ \forall i \in I$$

Specifically, we use the quadratic penalty method, which minimizes the unconstrained function

$$\Phi(x) = f(x) + \sum_i \lambda_i c_i(x) + \frac{\rho}{2} \sum_i c_i(x)^2$$

Starting with initial weights for λ_i, ρ, we iteratively update the λ_i by the following rule:

$$\lambda_i \leftarrow \lambda_i + \rho c_i(x_k) \text{ where } x_k \text{ is the solution at the } k\text{-th step}$$

The hyperparameter ρ is set at the beginning of the training process and remains fixed throughout.

5.2.3 Gradient Descent and the ADAM Optimizer

To perform the iterative process of gradient descent, we first split the training data into multiple sections called mini-batches. Starting with the first mini-batch, we evaluate the error for each data point in the mini-batch and then update the model accordingly. By splitting the training data into mini-batches, we are able to evaluate the gradient faster than if we were to load in the whole training data set, but we are also able to prevent 'noisy' gradients, which would occur if we updated the model after evaluating on a single data point each time. Once we have iterated through every mini-batch in the training data, we have completed one model training cycle, called an *epoch*.

To begin, we need to initialize our weights and biases correctly. If they start too small, the signal shrinks as it passes through each layer until its too tiny to be useful and if they start too large, the signal grows as it passes through each layer until its too massive to be useful. To initialize them just right, Xavier initialization [10] is used where each weight and bias is drawn from a uniform distribution with mean zero and variance $\frac{1}{n_{\text{in}}}$, where n_{in} is the number of neurons feeding into it. To compute the gradients at every step, automatic differentiation, also known as autograd, maintains a computational graph of all mathematical operations performed on a node and analytically differentiates the function by iteratively applying the chain rule.

In our case, we use an extension of the standard gradient descent methods, an optimizer called ADAM. Whereas gradient descent maintains one learning rate for all the model parameters, ADAM keeps track of a per-parameter learning rate *and* updates them using an exponential moving average of the first and second moments of the gradient. This optimizer works well for models with many parameters that are training on a large set of data.

5.3 The Randomized Matching Model

Given that we need to take gradients at every step of our deep learning algorithm, it is important that the functions we are differentiating are relatively smooth. However, one-to-one matching, as discussed in Part I, is a binary, where (m, w) is either matched or not. This leads to a non-smooth function for us to differentiate and therefore, requires us to introduce the concept of randomized matching, which will allow us to differentiate our loss more easily.

In the first part of this thesis, we discussed the formal model for one-to-one matching and more specifically, the deterministic model of one-to-one matching. Given a set of n men and

v women, we might represent the resulting stable matching μ as a $n \times v$ matrix d, where $d_{mw} = 1$ if man m and woman w are matched and otherwise, $d_{mw} = 0$. However, we might want to expand to consider probabilistic distributions over possible matchings rather than deterministic ones. To do so, we can introduce what are known as random matchings.

Definition 5.1 (Random Matching). *A random matching $r = [r_{mw}]_{m \in M, w \in W}$ is a real, doubly stochastic matrix, i.e., it satisfies*

(i) $0 \leq r_{mw} \leq 1$ *for all* $m \in M, w \in W$,

(ii) $\sum_{m \in M} r_{mw} = 1$ *for all* $w \in W$,

(iii) $\sum_{w \in W} r_{mw} = 1$ *for all* $m \in M$.

Here, r_{mw} represents the probability that man m is matched with woman w and moreover, the stochastic row vector $r_m = (r_{mw})_{w \in W}$ represents the probability distribution of man m matching with any woman, and the stochastic column vector $r_w = (r_{mw})_{m \in M}$ represents the probability distribution of woman w matching with any man.

Definition 5.2 (Permutation Matrix). *A permutation matrix is a square binary matrix that has exactly one entry of 1 in each row and column and 0s elsewhere.*

Definition 5.3 (Deterministic Matching). *A random matching r that is also a permutation matrix is called a deterministic matching.*

In Part I, we defined an important notion of stability for deterministic matchings, namely that a matching is stable if there are no blocking pairs (m, w) such that $m \succ_w \mu(w)$ and $w \succ_m \mu(m)$. For randomized matching, we define *ex ante justified envy* below for men, with the definition for women symmetrical.

Definition 5.4 (Ex Ante Justified Envy). *A random matching r causes ex ante justified envy of agent $i \in M$ towards a lower-ranked agent $j \in M \setminus \{i\}$ with $i \succ_c j$ for some woman $c \in W$ if $r_{ia} > 0$ for some $c \succ_i a$ with $a \in W \setminus \{a\}$ and $r_{jc} > 0$.*

Intuitively, this definition of ex ante justified envy tells us that if woman c prefers man i to man j and man i also prefers woman c to woman a, then (i, c) should be matched, and the probabilities that (i, a) and (j, c) are matched should converge to zero.

Definition 5.5 (Ex Ante Stability). *A random matching r is ex ante stable if no man or woman experiences ex ante justified envy as a result of r.*

As shown in Kesten and Ünver [12], for a deterministic matching, ex ante stability and the traditional definition of stability from Chapter 2 are equivalent. From here on out, we will always be referring to the one-to-one randomized matching when speaking about matching, as opposed to deterministic matchings in Part I.

Chapter 6

Finding Approximately Stable Matchings

As in Dütting et al.'s paper on deep learning for optimal auction design, we begin by demonstrating that our deep learning methods can replicate the theoretical results already know. In particular, we know that for any set of preferences, a stable matching must exist. We hope to answer the following question.

> Using a deep learning framework with randomized matchings, can we encode the notion of stability into a loss function to learn approximately stable matchings?

6.1 Experimental Setup

6.1.1 Neural Net Architecture

Using PyTorch, we implement a feed-forward neural network with a total of three linear, fully connected layers and a variable number of hidden nodes (default 100). Starting with a small case, as inputs, we take in randomly generated preference data for sets of five men and five women. At each node in the neural network, we use the activation function $\tanh x$. Given that our desired output is a random matching, we require that our resulting matrix output be of dimension 5×5 and be doubly stochastic.

As we train our model, the neural network repeatedly updates the probabilities in the random assignment matrix r. To compute r, we first have the neural network compute two random matchings $r_{1_{ij}}$ and $r_{2_{ij}}$, with the first matching normalized along the rows and the second matching normalized along the columns. Both normalizations are performed using the softmax function and the probability of matching man i with woman j can be computed as the minimum of the corresponding normalized scores, i.e.

$$r_{ij} = \min\left\{\frac{e^{r1_{ij}}}{\sum_{k=1}^{n} e^{r1_{kj}}}, \frac{e^{r2_{ij}}}{\sum_{k=1}^{v} e^{r2_{ik}}}\right\}$$

Note that by setting $r = \min(r_1, r_2)$, the resulting matrix may have row and column sums less than 1; we will address this issue below.

6.1.2 Preference Inputs

Given that we assume full preferences for each individual, meaning that everyone would prefer to be married to any member of the opposite sex rather than not being married at all, if we have m men and w women, our input is two $m \times w$ preference matrices P and Q, with entries limited to be within $(0, 1]$. A row in the first matrix P_m represents the preference profile of man m over all possible choices of women $w \in W$. In particular, the row P_m represents the discrete linear ordering in evenly spaced intervals from $\frac{1}{|W|}$ to 1. In contrast, a row in the second matrix Q_m is a vector where each Q_{mw} represents the the ranking of man m in the preference profile of woman w. Alternatively, whereas each row is a preference profile for each man in the first matrix P, each column is a preference profile for each woman in the second matrix Q. Combined, these provide a full set of preferences of every man over every woman, and vice versa. Since for all experiments, we look at the small market of five men and five women, each individuals' preference list can be represented as some permutation of the list $[1, 0.8, 0.6, 0.4, 0.2]$.

6.1.3 Optimization Problem

Using deep learning methods, our goal is to minimize the ex ante stability violation of the resulting random matching r given a set of preference inputs. First, we must encode the definition of ex ante stability into a continuous and differentiable loss function. To do so, we compute the expected ex post stability violation for a random matching r as a proxy for the ex ante stability violation by considering every possible blocking pair (i, c) with $i \in M, c \in W$, and looking at whether each man i or woman c experiences ex ante justified envy. We compute this as follows with P, Q defined as the input preference matrices of men's ranked lists over women and women's ranked lists over men, respectively.

Definition 6.1 (Expected Ex Post Stability Violation). *The expected ex post stability violation for a random matching r is given by*

$$stv(r) = \sum_{(i,c)\in M\times W} \left(\sum_{j\in M\setminus\{i\}} r_{jc} \cdot \max\{Q_{ic} - Q_{jc}, 0\} \right) \left(\sum_{a\in W\setminus\{c\}} r_{ia} \cdot \max\{P_{ic} - P_{ia}, 0\} \right)$$

Parsing this, if (i, c) is a blocking pair where $i \succ_c j$ and $c \succ_i a$ (encoded by the two max functions over P, Q), then we want to minimize the assignment probabilities of r_{jc} and r_{ia}, which represent sub-optimal pairings (encoded by minimizing the product r_{jc} and r_{ia}).

Additionally, notice in Definition 5.1 that the doubly stochastic matrix is specified as having the sum of entries in each row and each column be equal to 1. However, since we compute r_1 and r_2 by taking the softmax of rows and columns, respectively, and then taking the minimum of the two matrices, we can only guarantee that rows and columns of r sum to at most 1. To see how this might be problematic, consider for example

$$r_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; r_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; r = \min(r_1, r_2) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

In a deterministic matching setting, the all-zeros assignment matrix would be interpreted as keeping every agent single, but the matching would not be stable because any pair of agents (m, w) would prefer to be matched than to remain single. In fact, the ideal solution would be for every agent to have a spouse, so that the sum of entries in the random assignment matrix r is maximized. However, our inability to output a doubly stochastic matrix with entries that sum in each row and column to *exactly* 1 causes this computational problem with our theoretical definition of ex ante stability. Thus, in the 5×5 case with random matching r, our constrained optimization problem to solve would be

$$\min stv(r) \text{ s.t. } \sum_{i=1}^{|M|} \sum_{j=1}^{|W|} r_{ij} = 5$$

Experimentally, we saw that imposing a penalty when $\sum r_{mw} < 5$ was unsuccessful at preventing non-viable solutions like the non-zeros assignment matrix. Upon further examination, we realized that the divergence of r_1, r_2 is a major factor in causing the all-zeros matrix to be a solution. Thus, we argue that minimizing the divergence of r_1, r_2 is equivalent to maximizing the sum of entries in the random matching r, and attacks the problem more closely to its source, rather than after the minimum of r_1, r_2 has already been computed.

Therefore, the two constrained optimization problems

$$\min stv(r) \text{ s.t. } \sum_{i=1}^{|M|}\sum_{j=1}^{|W|} r_{ij} = 5 \text{ and } \min stv(r) \text{ s.t. } d(r_1, r_2) = 0$$

are equivalent.

Definition 6.2 (Divergence)**.** *The divergence of two matrices a, b is defined as follows.*

$$d(a,b) = \sum_{i=1}^{|M|}\sum_{j=1}^{|W|} |a_{ij} - b_{ij}|.$$

Furthermore, we also compute our expected ex post stability violation by decomposing it into two parts, the ex post stability violation for the column-stochastic matrix r_1 and the column-stochastic matrix r_2. Thus, our final constrained optimization problem is as follows

$$\min stv(r_1) + stv(r_2) \text{ s.t. } d(r_1, r_2) = 0.$$

Then, using the augmented Lagrangian method as discussed in Section 5.2.2, our loss function is given by

$$\mathcal{L} = stv(r_1) + stv(r_2) + \lambda d(r_1, r_2) + \frac{\rho}{2}(d(r_1, r_2))^2$$

If we have n men and n women, then we expect the loss computation to require $O(n^3)$ time, because there are n^2 possible blocking pairs to check, and $2n - 2$ possible independent deviations to check for each blocking pair, with $n - 1$ possibilities for $j \neq i$ and $n - 1$ possibilities for $a \neq c$.

6.2 Model Implementation

For each of our experiments, we include the following parameters. To set up the neural network, we have the number of men/women on one side of the market, the number of nodes in each hidden layer of the neural network, the learning rate for the ADAM optimizer, and the number of total epochs to train the model for. For building and evaluating the model, we have parameters that control the number of data points in the training and test set as well as the batch size in each set. We also have a parameter that controls how frequently the model is evaluated on the test set to measure improved performance over time. Lastly, for the augmented Lagrangian method, we have the hyper-parameters λ_{pen} and ρ_{pen} that control the size of the linear and Lagrangian penalty term, as well as a parameter for the update frequency of λ_{pen}. We use the PyTorch deep learning libraries to implement our model and run the experiments on the Harvard Odyssey computing cluster with NVIDIA GPU cores.

To assess our deep learning algorithm, we looked at four main metrics. First and foremost, we looked at the expected ex post stability violation $stv(r_1) + stv(r_2)$. Second, we looked at the overall loss $\mathcal{L}$, which was the summation of our ex post stability violation and the Lagrangian penalty term. We expect these to converge towards zero as we train our model.

Furthermore, we look at the completeness and overall social welfare of agents in this market, defined as follows.

Definition 6.3 (Completeness). *The completeness of a random matching* r*, which must be in the range* $[0, |M|]$*, is defined as*

$$\sum_{i=1}^{|M|} \sum_{j=1}^{|W|} |r_{mw}|$$

Completeness is the L1 norm of the random matching and we seek to maximize this by equivalently minimizing $d(r_1, r_2)$ in our loss function's Lagrangian penalty term.

Theorem 6.4 (Birkhoff-von Neumann). *If* A *is a doubly stochastic matrix, then there exist* $\theta_1, \dots, \theta_k \geq 0$ *with* $\sum_{i=1}^{k} \theta_i = 1$ *and permutation matrices* $P_1, \dots, P_k$ *such that*

$$A = \theta_1 P_1 + \dots + \theta_k P_k$$

For a random matching r with entries in each row and column summing to 1, we can decompose it using the Birkhoff-von Neumann Theorem into a probability distribution $\vec{\theta}$ over a set of deterministic matchings $P_1, \dots, P_k$.

Definition 6.5 (Welfare). *The overall social welfare created by a random matching* r*, with inputted preferences* P, Q *of men and women, is*

$$\sum_{i=1}^{|M|} P(m_i) \cdot r_i + \sum_{j=1}^{|W|} P(w_i) \cdot r_i^\top$$

In words, the overall social welfare of a random matching is measured by summing over the dot product of the preference list of each individual agent with their probability vector containing their randomized assignments. Note that we do not seek to maximize welfare in our optimization problem, but rather, look at it in comparison to the social welfare maximizing matching which is not necessarily stable.

Parameter	**Optimal Value**
Agents	5
Hidden Nodes	100
Epochs	300
Training Data Size	6400
Training Batch Size	128
Test Data Size	6000
Test Batch Size	100
Test Log Interval	10
λ_{pen}	0
ρ_{pen}	0.001
λ_{pen} Update Frequency	3

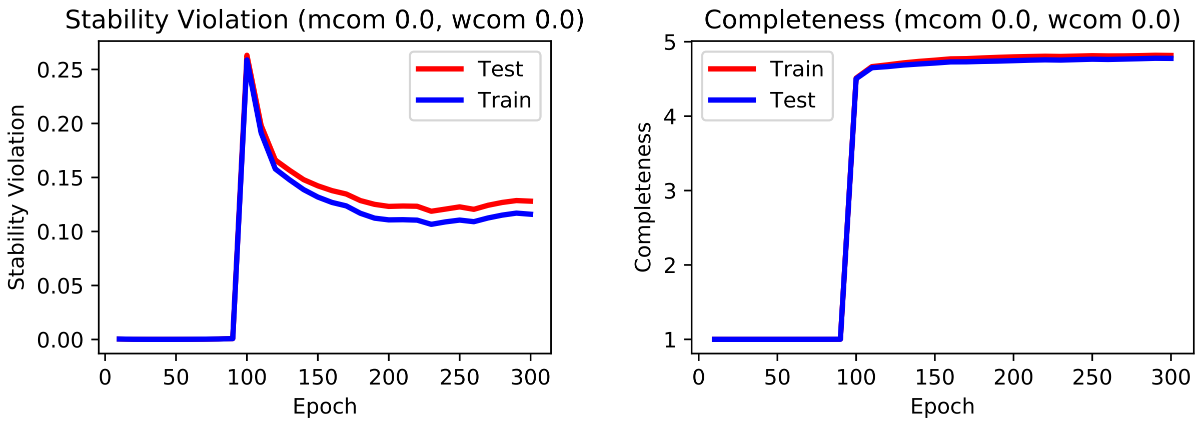
Table 6.1: Final Model Configuration

To find the optimal parameters, we iterated through many combinations of $\lambda_{\text{pen}}, \rho_{\text{pen}}$, and the update frequency that would minimize our stability violation below 0.1 and maximize the completeness to be above 4.5. The optimal parameters for our final set of experiments presented here are listed in Table 6.1.

6.3 Uncorrelated Preferences

After appropriately tuning our initial parameters, we look at the simplest case of matching for preferences drawn independently from the uniform distribution over all possible discrete preference lists (a total of 5! of them for any given individual) given the final model configuration in Table 6.1. For completely uncorrelated preferences on both sides, we see in Figure 6.1 plots showing the evolution of the stability violation, completeness, loss, and welfare over 300 epochs of training.

Looking first at our stability violation graph in Figure 6.1a, we see that it sharply increases for about the first 100 epochs before quickly decreasing and plateauing. This is a result of the interplay between the direct objective we are trying to minimize, the stability violation, and the Lagrangian penalty term which intends to minimize $d(r_1, r_2)$. Since we initialize very small $\lambda_{\text{pen}}, \rho_{\text{pen}}$, our Lagrangian penalty term starts out very small relative to the stability violation. In an effort to minimize the larger stability violation term, the match matrix quickly converges to be all-zero, which trivially minimizes the stability violation, while the Lagrangian penalty term grows quickly, given that $d(r_1, r_2)$ is extremely large. Near the 100th epoch, the Lagrangian penalty term has grown sufficiently large to balance



(a) Stability Violation

(b) Completeness

Figure 6.1: Plots of stability violation and completeness when both men and women's preferences are drawn uniformly at random.

the stability violation term, and for a short while, we have a sharp increase in the stability violation. However, over the remaining 200 epochs, the two terms balance each other out, achieving the goals of minimizing stability violation and and $d(r_1, r_2)$ simultaneously.

Experimentally, the stability violation achieves a minimum of 0.1281 at the end of 300 epochs. Given that we compute the stability violation over all ten agents (five men, five women), we can see that the stability violation is in fact, only 0.01281 for each individual agent, which is incredibly small compared to the fact that each individual's preferences is represented by five discrete, linearly spaced intervals ([0.2, 0.4, 0.6, 0.8, 1]). Alternatively, since 0.1281 is approximately halfway between 0 and 0.2, we can interpret the stability violation as follows. If we use this deep learning method to compute matchings for a set of five men and five women on a large set of preferences, for approximately half of them, only one agent receive a matching that violates their preferences and leads to a blocking pair.

In Figure 6.1b, we can see the opposite effect, where completeness remains near zero for the first 100 epochs, and then climbs quickly to reach its maximal level soon afterwards. Experimentally, the final matching achieves a maximum completeness of 4.7761. This is a positive result given that our completeness is extremely close to its maximum of 5. To interpret our result, if we use this deep learning method to compute matchings for a set of five men and five women on a large set of preferences, we will be able to decompose our final matching into a probability distribution over deterministic, approximately stable matchings more than 95% of the time.

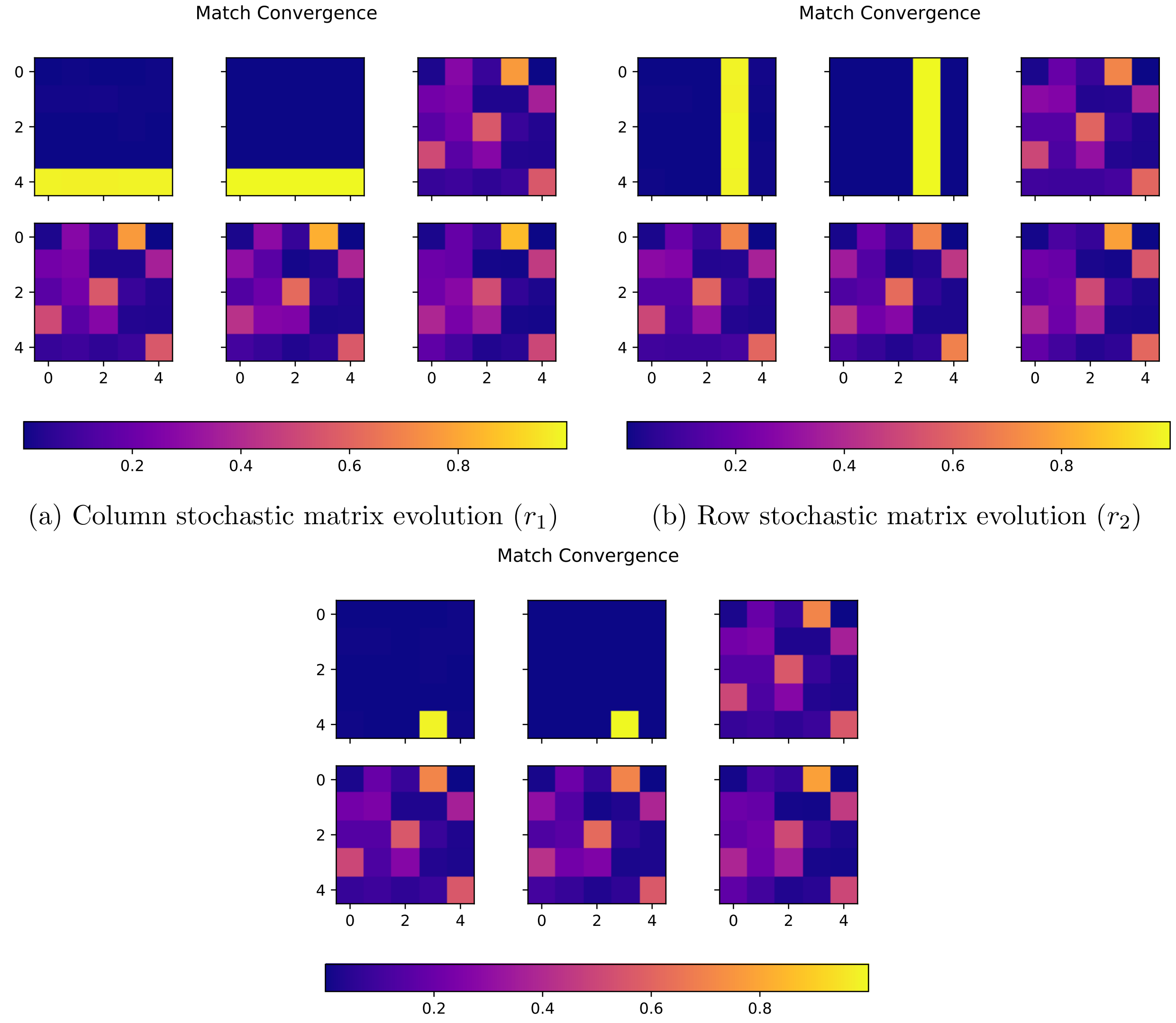
Overall, we are also able to see that in the case of uncorrelated preferences for all agents, the results of the training and test sets are extraordinarily similar. Given how closely they track each other despite the small size of our randomly drawn training and test sets (6400 independent preference sets in the training data and 6000 independent preference sets in the test data), it is likely that this method could be extended for larger data sets as well.

In addition to looking at the evolution of our stability violation and completeness over the 300 epochs, we also want to evaluate the efficacy of our Lagrangian penalty term, which measures the divergence of the row-stochastic and column-stochastic matrices used to derive the final matching. As noted prior, penalizing the divergence of these two matrices is used as a proxy to maximize the L1 norm of the matrix, and in Figure 6.2, we can see the evolution of $r_1, r_2, \min(r_1, r_2)$ over the 300 training epochs, with each matrix heat map updating every 60 epochs moving horizontally and then down within each subfigure. In the first 100 epochs, mirroring the graphs above, we see that a trivially stable matching is learned, with a singular matching between the last man and the second-to-last woman. However, this comes at the cost of the Lagrangian penalty term, meaning the row-stochastic and column-stochastic matrices diverge greatly. As we see in the first two plots of Figure 6.2a, every woman is matched with probability 1 to the last man, and in Figure 6.2b, every man is matched with probability 1 to the second-to-last woman. Clearly, these two matchings differ greatly, and when the minimum of the two matrices is taken, we get the first two plots in Figure 6.2c, which demonstrate the singular pairing that arises and reflect both the low stability violation and completeness prior to the spikes in Figures 6.1a and 6.1b. In the bottom right heat map of each subfigure, visualizing the final match, we see that there are some matches that occur with close to probability 1, such as the first man with the second-to-last woman, but we see that the matchings are less clear for the first and second women, for example.

6.4 Correlated Preferences

In the prior section, we assumed uncorrelated preferences for each man and woman. However, we would like to better understand how correlated preferences on one side or both sides of the market affect aggregate welfare. Using the additional parameters $\text{mcom\_val}, \text{wcom\_val}$, which vary between $[0, 1]$, we are able to denote the probability of an individual agent having a common value preference list as opposed to a private preference list [16, 3]. Mathematically, with n men and n women, we model this as so.

First, we draw a common preference list uniformly at random from the finite set of possible



(a) Column stochastic matrix evolution (r_1)

(b) Row stochastic matrix evolution (r_2)

(c) Final allocation matrix evolution ($\min(r_1, r_2)$)

Figure 6.2: Evolution of matching when both men and women's preferences are uncorrelated. In each subfigure moving horizontally and then down, each 5×5 matrix heat map (generated every 60 epochs) shows the current matching at that epoch. Brighter colors indicate higher probability matchings and darker colors indicate lower probability matchings.

discrete preferences for both men and women, notated as vectors of length n

$$C_m = \{c_m\}_{m \in M} \quad C_w = \{c_w\}_{w \in W}$$

Secondly, we draw private preferences for each individual agent, which gives a $n \times n$ matrix denoting each man's discrete preferences over every woman, and an $n \times n$ matrix denoting each woman's discrete preferences over every man. To generate the final preference matrix, we first draw n values from a Bernoulli distribution with probability mcom_val. For each successful draw, we replace the n-th man's private preference list with the common preference list. Similarly, we draw another n values from a Bernoulli distribution with probability wcom_val and for successful draws, replace the private preferences of the n-th woman with the common preference list. Thus, when mcom_val = wcom_val = 0, we have independent, uniformly random preferences being drawn for every agent, whereas when mcom_val = wcom_val = 1, every man has the same set of preferences over all women, and similarly for women.

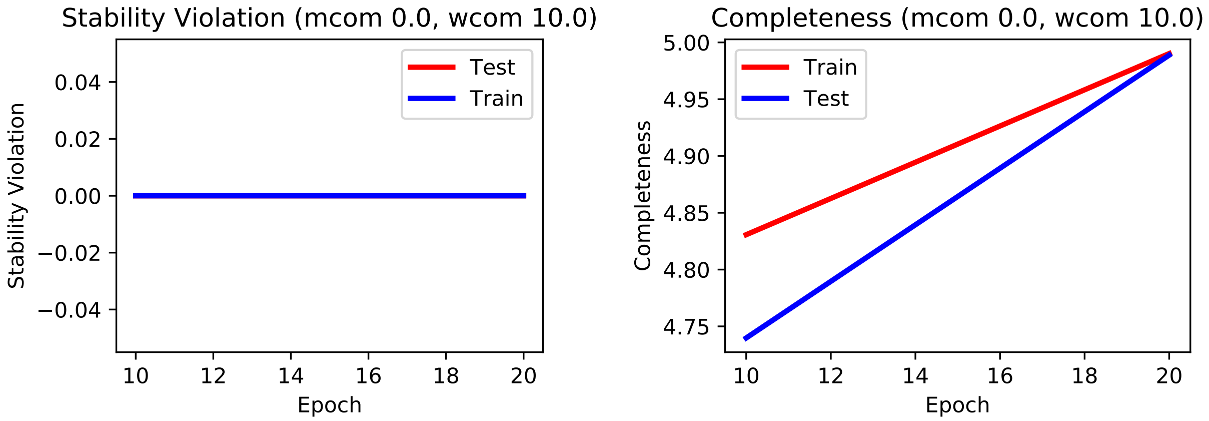
Given this mathematical framework, we perform the following experiments.

1. Vary wcom_val in 0.1 increments from 0 to 1, while keeping mcom_val = 0
2. Vary mcom_val in 0.1 increments from 0 to 1, while keeping wcom_val = 0
3. Vary mcom_val, wcom_val together in 0.1 increments from 0 to 1, with mcom_val = wcom_val

When discussing the results of these experiments in terms of stability violation, completeness, and match convergence, we focus on the first set of experiments, where men's preferences are completely uncorrelated and women's preference correlation varies. We see that the results of the second and third set of experiments produce similarly successful results and thus, omit their analysis for brevity.

6.4.1 One-Sided Full Correlation

To begin, we look at the most extreme case of one-sided correlation where men's preferences remain completely uncorrelated, as above, and women's preferences are completely correlated. In the framework that we discussed above, this means that every woman has the same common value preference list over all men. In this case, we see that the stability violation starts off, and remains extremely small through the entire training in Figure 6.3a, which lasts for about 20 epochs before reaching optimal conditions below 0.05. Similarly, we see in Figure 6.3b that the completeness increases linearly over the 20 epochs to 4.9882 out of the maximum of 5. This implies that if we evaluate our model on a large set of preferences, 25% of the time, only one agent would would receive a matching that violates their preferences and leads to a blocking pair, and over 99% of the time, the matching is complete. We see that in the case of maximal preference correlation on one side of the market, we are able to quickly find a matching between men and women that minimizes the stability violation and maximizes the completeness of the final match. As desired, our results in the test set also track the results in the training set. In addition, we see symmetric results for the case where men's preferences are completely correlated and women's preferences remain completely uncorrelated.



(a) Stability Violation

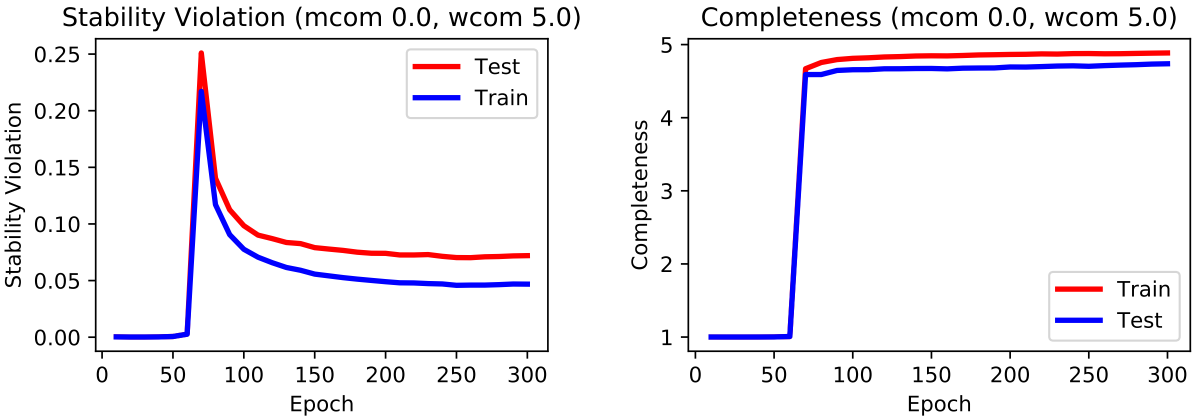
(b) Completeness

Figure 6.3: Stability violation and completeness plots when women's preferences are completely correlated and men's preferences are completely uncorrelated.

We omit the match convergence plots for this case because for fully correlated preferences on one side of the market, a deterministic matching is learned almost immediately, and there is no visible evolution.

6.4.2 One-Sided Partial Correlation

To examine a more interesting, intermediate example of one-sided correlation, we now look at the case where men's preferences remain completely uncorrelated and women's preferences are correlated with wcom_val = 0.5. This means that each woman has a 50% chance of maintaining their individual preferences and a 50% chance of adopting the common value preferences. After 300 epochs, the final stability violation was 0.072 and the completeness of the match was 4.7376. Similar to the pattern in the case of uncorrelated preferences, we again see a similar trend where the first 60 epochs strongly favor minimizing the stability violation, and then after a dramatic spike in both, succeed in optimizing for both equally well.



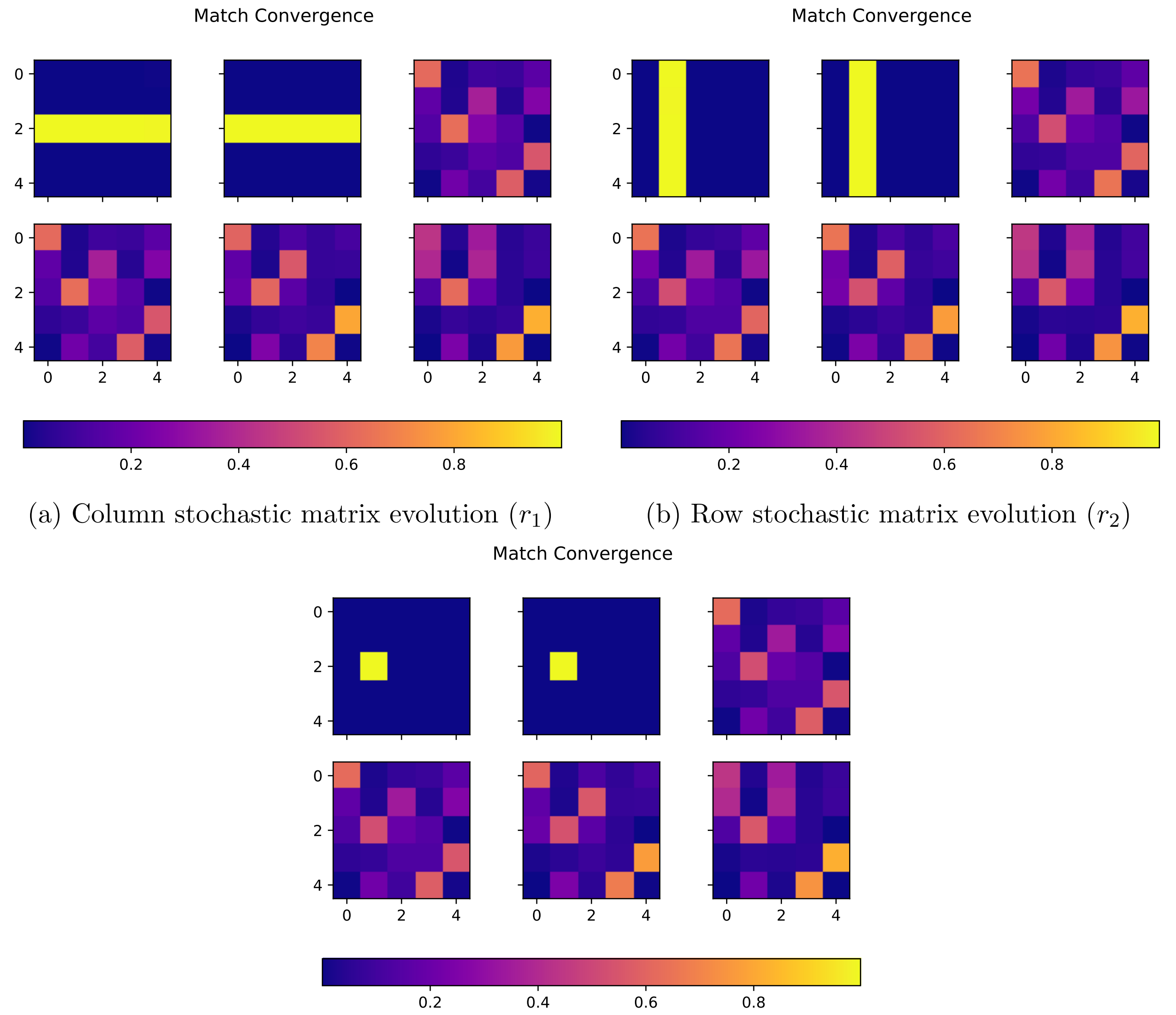
(a) Stability Violation

(b) Completeness

Figure 6.4: Stability violation and completeness plots when women's preferences are 50% correlated and men's preferences are completely uncorrelated.

In this intermediate case, we can also analyze the allocation matrices as seen in Figure 6.5. In contrast to the prior case of uncorrelated preferences, we can see that the matches learned here are closer to looking like a deterministic matching. Indexing the men and women each from 0 to 4, in the bottom right heat map, we see that the we have close to deterministic matchings for (m_4, w_3) and (m_3, w_4). We also have a slightly weaker matching of (m_2, w_1) and a relatively even split for the remaining matchings between m_0, m_1 and w_0, w_2. Given that correlated preferences make the data sets more homogeneous, we believe that an increase in preference correlation is likely to lead to stable matchings that look more deterministic.

Generally speaking, as we increase wcom_val from 0 to 1 and make women's preferences more homogeneous, the stability violation becomes smaller, the completeness of the final matching gets closer to 5. Furthermore, the visualization of match convergence depicts more deterministic final matchings and the model takes fewer epochs to train before converging to an optimum. Due to a decrease in the variation of both the training and test data sets as women's preferences become more homogeneous, there are fewer patterns for our deep learning methods to discover. Therefore, it makes sense that the final matchings improve on metrics such stability violation and completeness and more closely approximate stable deterministic matchings as wcom_val grows.



(a) Column stochastic matrix evolution (r_1)

(b) Row stochastic matrix evolution (r_2)

(c) Final allocation matrix evolution ($\min(r_1, r_2)$)

Figure 6.5: Evolution of matching when women's preferences are 50% correlated and men's preferences are uncorrelated. In each subfigure moving horizontally and then down, each 5×5 matrix heat map (generated every 60 epochs) shows the current matching at that epoch. Brighter colors indicate higher probability matchings and darker colors indicate lower probability matchings.

6.5 Welfare Comparisons

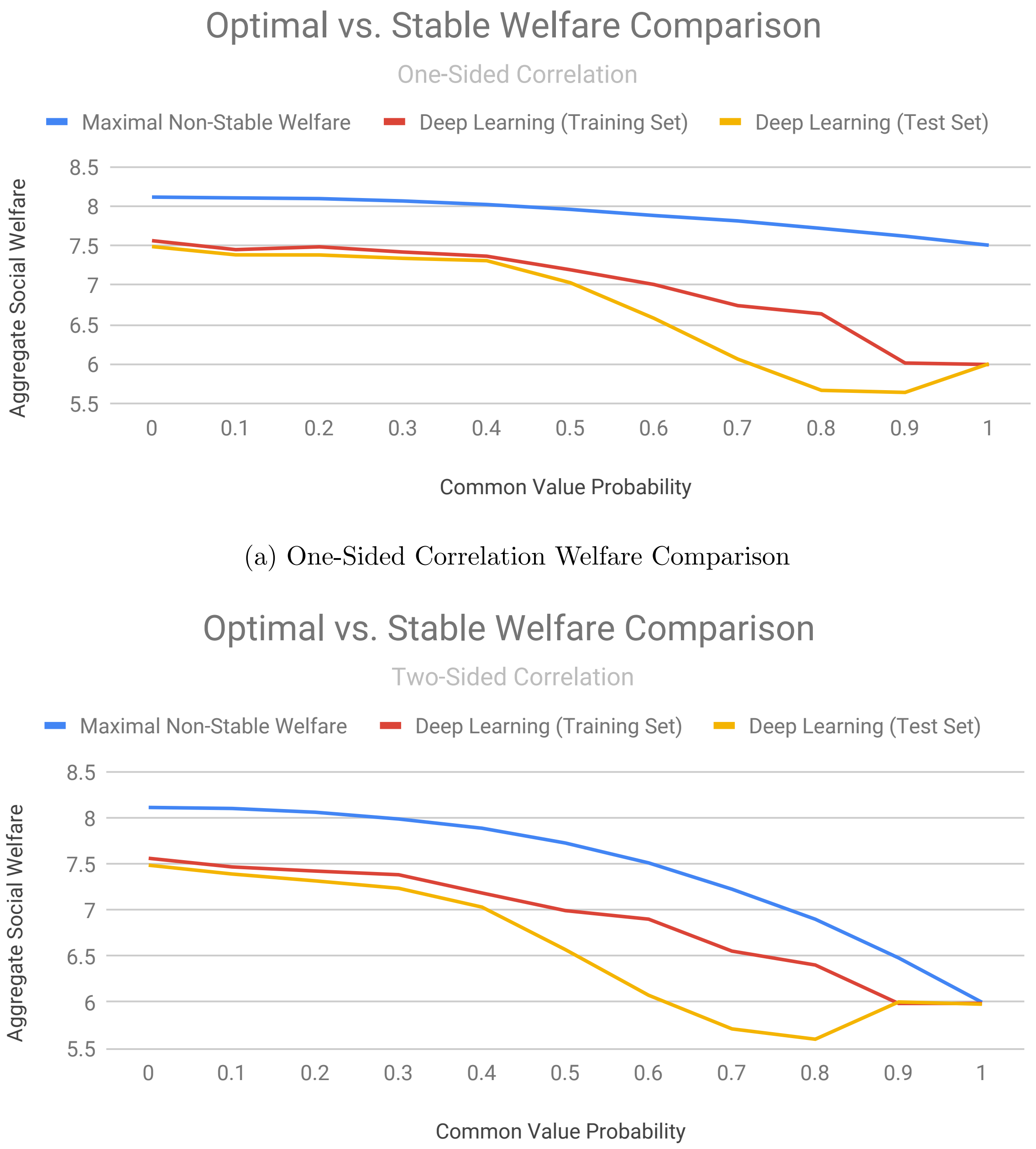
Finally, we know that the welfare-maximizing matching is not necessarily contained in the set of stable matchings. Thus, we would like to compare the aggregate welfare attained in the stable matching found via deep learning methods to the maximal expected welfare of any matching in our market (an upper bound). More specifically, we compare these two types of welfare in two cases: one-sided and two-sided correlation, where we vary the correlation in increments of 0.1 from 0 to 1.

To find the expected maximal welfare, we recast our stable matching problem as a linear sum assignment or minimum weighted matching problem on a bipartite graph. Using the Hungarian algorithm [15] in `scipy.optimize`, we compare the maximal welfare for non-stable matchings to the aggregate welfare of approximately stable matchings found with our deep learning methods on the training and test sets. In Figure 6.6, we see three key findings.

First, regardless of one-sided or two-sided correlation and regardless of the correlation probability, the expected maximal welfare on the training and test sets is higher than the welfare of the learned stable matchings. This tells us that there is a trade-off between stability and welfare maximization, and that imposing the stability constraint does aggregate a loss in overall welfare for the individuals. However, this tradeoff is rather small for each of the ten agents, given that the decrease in aggregate welfare is on average, 0.5, and at most, 1.5.

Additionally, we see that as the common value probability increases, meaning the preferences become more homogeneous, the overall social welfare decreases in both the welfare-maximizing and stable matching. This can be explained by the fact that when agents have greater heterogeneity in preferences, it is more likely that each agent can be matched to one of their top choices, whereas if everyone on one side of the market has the same set of common value preferences, only one of them can receive their top-ranked match, and more agents will receive lower-ranked matches.

Lastly, we see that the expected maximal welfare decreases much more dramatically in Figure 6.6b with two-sided correlation than in Figure 6.6a with one-sided correlation. Whereas in Figure 6.6a the expected maximal welfare doesn't dip below 7.5, in Figure 6.6b the expected maximal welfare for fully correlated preferences on both sides is 6, at the same level as the average aggregate welfare for the stable matchings in both the training and test sets. Similar to what was argued in the previous finding, we see that severe homogeneity in preferences leads to a sharp decrease in expected maximal welfare, due to the fact that more agents will be receiving lower-ranked matches when competing for the same matches.



(a) One-Sided Correlation Welfare Comparison

(b) Two-Sided Correlation Welfare Comparison

Figure 6.6: Comparison of expected maximal welfare to expected aggregate welfare in stable matchings as correlation varies in 0.1 increments between 0 and 1 for both one-sided and two-sided correlation.

Chapter 7

Conclusions

In Part I, we provide exposition on the main results from the matching literature in one-to-one and many-to-one settings. Heavily relying on the work of Roth and Sotomayor, we prove the existence of stable matchings, discuss the structure of the set of stable matchings, and illuminate the strategic questions that naturally arise in stable matching mechanisms. Furthermore, we provide real-world context at the end of each chapter to help the reader better understand how these mathematical findings play out in practice.

In Part II, we begin by providing the fundamentals of machine learning, explain the experimental setup of our deep learning framework, and conclusively demonstrate that this novel computational method is able to replicate the theoretical results from Part I and learn stable matchings given preference inputs. Specifically, we demonstrate the following:

1. Our deep learning framework is able to learn an approximately stable matching when preferences on both sides of the market are uncorrelated.
2. Our deep learning framework is able to learn an approximately stable matching when preferences on one or both sides of the market are correlated at varying levels.
3. Our deep learning framework is able to provide a computational comparison of the maximal welfare in a non-stable matching with the welfare of stable matchings at various preference correlation levels.

The success of these experiments suggest that the deep learning framework has great potential to further the study of matching markets by analytically exploring difficult theoretical questions.

7.1 Next Steps and Future Work

In Chapter 6, we succeeded in encoding the notion of stability into a loss function and given individual preferences of each agent in the market, were able to design a neural net to learn an approximately stable matching between men and women. However, we also saw in Theorem 4.2 that no stable matching mechanism is also DSIC.

Although a matching mechanism cannot be simultaneously stable and DSIC, it is not well understood how bad this tradeoff is in practice. If we were able to create an DSIC matching mechanism, what would the stability of the resulting matches look like? Would there be a singular blocking pair (close to stable) or would there be many blocking pairs (very unstable)? We would like to computationally understand the tension between stability and IC by imposing an additional DSIC constraint on the deep learning framework built in Chapter 6 and studying to what magnitude the stability of the resulting matches differs from the stability of the matches learned in Chapter 6. To impose the DSIC constraint, we use a similar implementation of RegretNet, as developed by Dütting et al. [6], but adapted for the matching setting.

7.1.1 Quantifying Regret

To enforce the DSIC constraint, we must define an individual's *ex post regret* for reporting truthful preferences. There are two ways in which individual agents can mis-report their preferences.

1. Agents can mis-report by truncating their preferences and stating that they would prefer to be single rather than be matched with someone low on their preference list.
2. Agents can mis-report by re-ordering their preferences and stating the ranking of their possible matches in a non-truthful order.

Given submitted preferences p'_i, whether truthful or not, the resulting match r is returned, with r_i representing the probability vector of matchings for agent i. The utility of agent i is then the dot product of r_i with their true preferences p_i. To measure the ex post regret rgt_i for each agent i, we need to sample from the possible preference mis-reports for each agent, compute the expected utility gain (if any) from these mis-reports, and sum over all agents to quantify the total regret for submitting truthful preferences to the mechanism.

In order to impose the DSIC constraint, we need $rgt_i = 0$ for all $i \in M \cup W$. Given that this is again a constrained optimization problem as we had in Chapter 6, but with $|M| + |W|$

more constraints, we can create the corresponding hyper-parameters λ_{rgt_i} for each agent and a ρ_{rgt} such that $|M| + |W|$ more Lagrangian penalty terms can be added to the loss function $\mathcal{L}$. Given this, our new optimization problem is as follows:

$$\min\ stv(r_1) + stv(r_2) \text{ s.t. } rgt_i = 0\ \forall i \in M \cup W \text{ and } d(r_1, r_2) = 0$$

Mirroring our work in Chapter 6, we can begin by finding optimal initial hyper-parameters to train our model. Then, beginning with the case of uncorrelated preferences, we can compare the magnitude of the stability violation and completeness before and after imposing the DSIC constraint. We can also do the same comparison of stability violation and completeness for correlated preferences. Lastly, we can compare the overall social welfare of these approximately stable matchings before and after imposing the DSIC constraint. By providing quantitative metrics to measure the trade-off between stability and DSIC constraints, this work would be able to provide great insight and nuance into the theoretical understanding of this impossibility theorem.

7.1.2 Policy Impacts

Moreover, if it is discovered that imposing DSIC constraints does not drastically increase the stability violation, these deep learning methods have the potential to be used in practice in areas where methods like deferred acceptance are being used now. As discussed prior in Section 4.3, matching mechanisms like the original Boston public school mechanism allowed strategic manipulations that caused great inequity in school assignments. Even though deferred acceptance is now implemented, the matching mechanism, though stable, still remains DSIC for only one side of the market.

However, if a DSIC, approximately stable matching mechanism could be created, it would be feasible for Boston public schools to implement this deep learning method as opposed to deferred acceptance. By doing so, both schools and parents would have incentive to truthfully report their preferences, and the resulting matching would still be very close to stable. Thus, future work in this area would not only be of theoretical interest to better understand the trade-offs of the impossibility theorem, but also have real-world ramifications in areas where matching algorithms are currently implemented.

Bibliography

[1] Atila Abdulkadiroglu, Parag Pathak, and Alvin Roth, *Strategy-proofness versus efficiency in matching with indifferences: Redesigning the new york city high school match*, (2009).

[2] Atila Abdulkadiroglu, Parag Pathak, Alvin Roth, and Tayfun Sonmez, *Changing the boston school choice mechanism*, (2006).

[3] Itai Ashlagi, Yash Kanoria, and Jacob D. Leshno, *Unbalanced random matching markets: The stark effect of competition*, Journal of Political Economy **125** (2017), 69–98.

[4] Charles Blair, *Every finite distributive lattice is a set of stable matchings*, Journal of Combinatorial Theory **37** (1984), no. 3, 353356.

[5] Gabrielle Demange and David Gale, *The strategy structure of two-sided matching markets*, Econometrica **53** (1985), no. 4, 873.

[6] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, and David C. Parkes, *Optimal auctions through deep learning*, (2018).

[7] Tams Fleiner, *A fixed-point approach to stable matchings and some applications*, Mathematics of Operations Research **28** (2003), no. 1, 103126.

[8] David Gale and Lloyd Shapley, *College admissions and the stability of marriage*, The American Mathematical Monthly **69** (1962), no. 1, 915.

[9] David Gale and Marilda Sotomayor, *Some remarks on the stable matching problem*, Discrete Applied Mathematics **11** (1985), no. 3, 223232.

[10] Xavier Glorot and Yoshua Bengio, *Understanding the difficulty of training deep feedforward neural networks*, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics **9** (2010), 249256.

[11] John William Hatfield and Paul R Milgrom, *Matching with contracts*, American Economic Review **95** (2005), no. 4, 913935.

[12] Onur Kesten and M. Utku Ünver, *A theory of school choice lotteries*, Theoretical Economics **10** (2017), 543–595.

[13] Donald Knuth, *Marriages stables*, Les Presses de l'Universite de Montreal.

[14] Scott Duke Kominers, *Introduction to matching and allocation problems*, Mar 2019.

[15] Harold W. Kuhn, *The hungarian method for the assignment problem*, Naval Research Logistics Quarterly **2** (1955), 8397.

[16] SangMok Lee, *Incentive compatability of large centralized matching markets*, The Review of Economic Studies **84** (2015), 444–463.

[17] James Liang, *An introduction to deep learning*, Oct 2018.

[18] D. G. Mcvitie and L. B. Wilson, *Stable marriage assignment for unequal sets*, Bit **10** (1970), no. 3, 295309.

[19] Parag A Pathak and Tayfun Snmez, *Leveling the playing field: Sincere and sophisticated players in the boston mechanism*, American Economic Review **98** (2008), no. 4, 16361652.

[20] Alvin Roth, Tayfun Sonmez, and M. Utku Unver, *Kidney exchange*, Quarterly Journal of Economics (2004), 457488.

[21] Alvin E. Roth, *The economics of matching: stability and incentives*, Mathematics of Operations Research **7**, 617628.

[22] ______, *The Evolution of the Labor Market for Medical Interns and Residents : A Case Study in Game Theory*, 1984.

[23] ______, *Misrepresentation and stability in the marriage problem*, Journal of Economic Theory **34** (1984), no. 2, 383387.

[24] ______, *The college admissions problem is not equivalent to the marriage problem*, Journal of Economic Theory **36** (1985), no. 2, 277288.

[25] ———, *On the allocation of residents to rural hospitals: A general property of two-sided matching markets*, Econometrica **54** (1986), no. 2, 425.

[26] ———, *Deferred acceptance algorithms: History, theory, practice, and open questions*, International Journal of Game Theory **36** (2008), 537569.

[27] Alvin E. Roth and Elliott Peranson, *The redesign of the matching market for american physicians: Some engineering aspects of economic design*, The American Economic Review **89** (1999), no. 4, 748780.

[28] Alvin E. Roth and Marilda A. Oliveira Sotomayor, *Two-sided matching*, Cambridge University Press, 1992.