12-2018

Strategic Behavior and Manipulation in Gender-Neutral Matching Algorithms

Sanjay Varma Rudraraju
sr2567@rit.edu

Strategic Behavior and Manipulation in Gender-Neutral Matching Algorithms

APPROVED BY

SUPERVISING COMMITTEE:

Dr. Hadi Hosseini and Dr. Ivona Bezáková, Advisor Date
Department of Computer Science
Rochester Institute of Technology

Dr. Stanislaw Radziszowski, Reader Date
Department of Computer Science
Rochester Institute of Technology

Dr. Peizhao Hu, Observer Date
Department of Computer Science
Rochester Institute of Technology

Strategic Behavior and Manipulation in Gender-Neutral Matching Algorithms

by

Sanjay Varma Rudraraju

THESIS

Presented to the Department of Computer Science
of the Golisano College of Computing and Information Sciences
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science in Computer Science

Rochester Institute of Technology

December 2018

Dedicated to my parents who are my greatest strength and biggest supporters, my thesis advisers who made sure that every step of mine is in the right direction, and Divya Pisal who always had my back, motivated me at my worst, and without whom I would have never finished this thesis.

Abstract

Strategic Behavior and Manipulation in Gender-Neutral Matching Algorithms

Sanjay Varma Rudraraju, M.S.
Rochester Institute of Technology, 2018

Supervisor: Dr. Hadi Hosseini and Dr. Ivona Bezáková

Within artificial intelligence, the sub-field of multi-agent systems studies the foundations of agent interactions and strategic behavior. Two-sided matching is one of the most fundamental problems in this field with applications in matching residents to hospitals, kidney donors to receivers and students to high schools. The earliest algorithm that solved this problem is the Gale-Shapley algorithm which guarantees a stable matching based on the preferences of both sides but has a drawback of favoring one side over the other, that is, proposers always get their most optimal stable partner.

We consider the design and analysis of gender-neutral stable matching algorithms where the proposing side from both sides is randomly chosen thereby giving an equal probability for both sides to get their most optimal stable partner (ex-ante). Later, we focus on investigating if an agent can exhibit strategic behavior i.e., whether it is possible for an agent to manipulate so that he/she improve the partner obtained when on the proposed side while retaining the partner obtained when on the proposing side.

The results obtained showed that for some manipulation algorithms, agents can still manipulate the outcome even when the decision of which side is proposing is unknown. Also, empirical evaluations were performed to understand and solidify the results.

Table of Contents

Chapter 1

Introduction

Two-sided matching is an area of interest for both computer scientists and economists due to the various real-world applications. A problem in the two-sided matching domain is the stable marriage problem which poses the following question: *How do we match two sides of a market who have preferences over the members of the other side?*. It is traditionally known as the stable matching problem because the two sides of a market are represented using men and women. The seminal algorithm that solved this problem is the Gale-Shapley algorithm. This algorithm has various applications such as matching residents to hospitals in the US - National Resident Matching Program (NRMP) [5] and several other nations such as Canada, Japan, and the UK [6]. The algorithm is also used for public school admissions in Boston and New York [1, 2], recruiting university faculty in France [5], online matrimony in India and auction mechanisms for sponsored search in Internet search engines [5].

The Gale-Shapley algorithm, hereafter referred to as the GS algorithm, works by one side proposing to the other and finds a stable matching where every man and woman on both sides are matched to someone on the other side and no man and woman prefer each other over their current partners. The GS algorithm favors one side over the other by assigning optimal stable partners for the proposing side and pessimal stable partners for the proposed side [4]. This creates a bias in the matching and is the basis for our first problem statement: *Can an algorithm be designed that will not have this bias and still result in a stable matching?*. In our investigation, we found that the design of such an algorithm is possible but made us question if an agent participating in the matching can manipulate to improve their outcome. This led us to another interesting problem that we decided to tackle i.e., *Can an agent manipulate his/her preferences to get a more preferred partner than their assigned partner if they are on the proposed side?*.

Chapter 2 introduces the stable marriage problem and also discusses the various notations along with the definitions properties used throughout the thesis. Chapter 3 talks about the various real world applications of the GS algorithm. Chapter 4 discusses the related work in the sub-domain of stable matching and unbiased matching algorithms. Chapter 5 has the design of the gender-neutral algorithm along with the theorems and proofs that have been written based on the gender-neutral algorithm manipulation results. Chapter

6 discusses the empirical evaluations performed to corroborate the results. Finally, Chapter 7 speaks about the conclusions that we arrived at and the future directions.

Chapter 2

Preliminaries

The Stable Marriage Problem (SMP) is the problem of finding a stable matching between two equal sized disjoint sets of elements given a strict ordering of preferences over the members of the opposite side. Let *men* M and *women* W be two equal sized disjoint sets where an agent from M be m_i and from W be w_j. Let $\succ_{a_i}$ denote agent a_i's strict ordering of preferences, referred hereafter as a preference list over the other set, and $\succ_{\mathrm{C}}$ denote a complete set of preference lists (referred to hereafter as a preference profile) for the SMP. For example, if agent m_i strictly prefers w_j over w_k we write $w_j \succ_{m_i} w_k$. A *Matching* is a one to one mapping $\mu : M \cup W \rightarrow M \cup W$ such that $\mu(m_i) = w_j$ and $\mu(w_j) = m_i$ if the agents (m_i, w_j) are matched to each other.

The seminal algorithm to solve the SMP is the Gale-Shapley [4] algorithm in which each unmatched man in a iterative manner proposes to each woman on his ordered preference list. If the woman is single, she immediately accepts the proposal and if not, she compares the proposer with her current partner and matches with her most preferred partner as submitted in her preference list. The pseudo-code for the Gale-Shapley algorithm is as follows:

Algorithm 1: The Gale-Shapley Algorithm

Data: M, W, $\succ_{\mathrm{C}}$

Result: μ - final matching

begin

  initialize all men and women on both sides to unmatched and set men to proposers

  while $\exists$ *some man* m_i *is unmatched and hasn't proposed to every woman on the other side* **do**

    $w_j \longleftarrow$ most preferred agent on proposer list to whom proposer hasn't yet proposed

    if w_j *is unmatched* **then**

      mark m_i and w_j as matched

    else if w_j *prefers* m_i *to current partner* m_k **then**

      mark m_i and w_j as matched and m_k as unmatched

    else

      w_j rejects m_i and mark w_j proposed to m_i

Example: Given an instance, $M = \{m_1, m_2, m_3\}$ and $W = \{w_1, w_2, w_3\}$.

$$
\begin{array}{llll}
\succ_{m_1}: & w_1 & w_2 & w_3 \\
\succ_{m_2}: & w_2 & w_1 & w_3 \\
\succ_{m_3}: & w_2 & w_3 & w_1
\end{array}
\qquad\qquad
\begin{array}{llll}
\succ_{w_1}: & m_2 & m_3 & m_1 \\
\succ_{w_2}: & m_1 & m_2 & m_3 \\
\succ_{w_3}: & m_1 & m_2 & m_3
\end{array}
$$

Applying the Gale-Shapley algorithm where men propose to the women, we have:

- Step 1: m_1 proposes to his top choice w_1 and since w_1 is unmatched, a pair (m_1, w_1) is formed.
- Step 2: m_2 proposes to his top choice w_2 and since w_2 is unmatched, a pair (m_2, w_2) is formed.
- Step 3: m_3 proposes to his top choice w_2 and since w_2 is matched to m_2, w_2 compares m_2 and m_3 based on her preferences and rejects m_3.
- Step 4: Next, m_3 proposes to his next choice w_3 and since w_3 is unmatched, a pair (m_3, w_3) is formed.
- Step 5: The algorithm terminates since all the agents have been matched.

The stable matching of the above instance obtained by using the GS algorithm is $\mu = \{(m_1,w_1),(m_2,w_2),(m_3,w_3)\}$

Stability is where no participant would leave their match to form a better matching along with another agent who he/she prefers more over the matched partner (blocking pair). The outcome should therefore be a desirable outcome for every participant such that no blocking pair is formed.

Definition 1. Given a matching pair μ, a pair (m_i,w_j) is called a *blocking pair* if $w_j \succ_{m_i} \mu(m_i)$ and $m_i \succ_{w_j} \mu(w_j)$. A matching is said to be *stable* if it contains no blocking pair.

Example: Given an instance

$M = \{m_1, m_2, m_3, m_4, m_5\}$ and $W = \{w_1, w_2, w_3, w_4, w_5\}$

$$
\begin{array}{llllll}
\succ_{m_1}: & w_1 & w_5 & w_4 & w_2 & w_3 \\
\succ_{m_2}: & w_5 & w_1 & w_2 & w_3 & w_4 \\
\succ_{m_3}: & w_2 & w_3 & w_1 & w_4 & w_5 \\
\succ_{m_4}: & w_5 & w_1 & w_2 & w_3 & w_4 \\
\succ_{m_5}: & w_4 & w_5 & w_1 & w_2 & w_3
\end{array}
\qquad
\begin{array}{llllll}
\succ_{w_1}: & m_5 & m_2 & m_3 & m_1 & m_4 \\
\succ_{w_2}: & m_4 & m_2 & m_3 & m_1 & m_5 \\
\succ_{w_3}: & m_5 & m_3 & m_2 & m_1 & m_4 \\
\succ_{w_4}: & m_4 & m_1 & m_2 & m_5 & m_3 \\
\succ_{w_5}: & m_1 & m_5 & m_2 & m_4 & m_3
\end{array}
$$

The stable matching obtained using the Gale-Shapley algorithm (which we will discuss later) for above example is $\mu_{\mathrm{m}} = \{(m_1,w_1), (m_2,w_5), (m_3,w_3), (m_4,w_2), (m_5,w_4)\}$.

In this instance, woman w_1 prefers m_5 over her partner m_1 but m_5 is matched to w_4 who according to w_5 is a better partner than w_1. Hence

(m_5,w_1) is not a blocking pair and similarly other pairs of men and women can also be verified to show that there is no blocking pair this matching, therefore the matching is stable.

In the stable marriage problem, strategy-proofness is very desired as there should be no incentive for any participant to act in a strategic manner such as by manipulating his/her preference list and improve his/her matching. If the matching algorithm is strategy-proof then every agent would be truthful in submitting their preference list and would not try to misreport his/her preference list.

Definition 2. A matching algorithm is *strategy-proof* when truthful reporting is the dominant strategy for every agent. That is, no man or woman can improve their matching by manipulating their preference ranking while the preference ranking stays the same for the other agents. A matching algorithm is said to be strategy-proof for men if any woman can manipulate her preference list to improve her partner outcome but no man can improve his partner outcome by manipulating. Similarly, a matching algorithm is said to be strategy-proof for women if there exists a man that can manipulate and improve his match but no woman can manipulate to improve partner outcome.

Roth proved that the Gale-Shapley algorithm is strategy-proof for men but not for women [10]. In the above example, woman w_1 submits her true preference list as $\succ_{w_1} = m_5 \succ m_2 \succ m_3 \succ m_1 \succ m_4$ and gets partner m_1 upon using the Gale-Shapley algorithm. If instead she submits a manipulated preference list $\succ'_{w_1} = m_5 \succ m_2 \succ m_4 \succ m_3 \succ m_1$ she would get the man m_2, who is better than m_1, as her partner upon using the Gale-Shapley algorithm. Hence, .

The Gale-Shapley algorithm is one of the most popular algorithms used to solve the stable matching problem and the matching that is produced using this algorithm can be shown stable for both sides and strategy-proof for men. The algorithm results in matching men to their best possible partner i.e., *man-optimal partner* and women to their worst partner i.e., *woman-pessimal partner* [4]. The man-optimal partner is a partner of man m_i such that the partner is the best woman he can receive in any possible stable matches that can be obtained in a given instance. Similarly, *woman-optimal partner* refers to the best man a woman can receive in any possible stable matches. On the contrary, *man-pessimal partner* and *woman-pessimal partner* refer to the worst partner that can be obtained by any man or woman respectively, participating in the stable marriage problem.

Strategy-proofness is another important property sought after in any matching algorithm. The Gale-Shapley algorithm isn't strategy-proof for both sides but is strategy-proof for men (to be specific, proposers) but not the women (proposed to). The side which is proposed in the Gale-Shapley algorithm can manipulate their preference list to improve their outcomes by changing the order of partners ranked in true preference list. Manipulation algorithms can be developed to find optimal manipulation strategy in order to get a better partner than the partner received while submitting true preference lists.

Definition 3. A matching algorithm is said to be *manipulable* by an agent a_i if there exists preference profiles $\succ'_{\mathrm{C}}$ and $\succ_{\mathrm{C}}$ which only differ in the preference list of the agent a_i such that $\mu'(a_i) \succ \mu(a_i)$ where μ and μ' are the matching before and after manipulation respectively obtained using the Gale-Shapley algorithm.

In the above example, woman w_1 submits her true preference list as $\succ_{w_1} = m_5 \succ m_2 \succ m_3 \succ m_1 \succ m_4$ and gets partner m_1 upon using the Gale-Shapley algorithm. If instead she submits a manipulated preference list $\succ'_{w_1}$ $= m_2 \succ m_4 \succ m_5 \succ m_3 \succ m_1$ she would get the man m_2 as her partner upon using the Gale-Shapley algorithm. This manipulation is obtained using a manipulation algorithm designed by Teo and Sethuraman, hereafter referred to as the TS manipulation algorithm, (which will be discussed in the Chapter 4) and shows that the Gale-Shapley algorithm is manipulable [13].

An algorithm is said to be *Gender-Neutral* if both men and women have equal probability of getting their optimal partner.

The matching obtained when men are the proposers in the gender-neutral algorithm is represented by μ_{m} and when women are the proposers by μ_{w}. We use μ'_{m} and μ'_{w} to denote the matching obtained using TS manipulated preference when men and women propose respectively. Similarly, μ''_{m} and μ''_{w} are used when the matching are obtained using inconspicuous manipulated preference list (which will be discussed in Chapter 4).

The term *rank of an agent* with respect to other agent's preferences is used extensively in this research which refers to the position of partner obtained by agent in the submitted preference list of the other agent. It can be simply viewed as, if there are a list of agents, the partner that is most liked is ranked 1 and then partners are ranked in the order of natural numbers.

The *rank* of the partner of agent a_i is the position of $\mu(a_i)$ in the preference list of a_i where if a_i ranks 'k' agents in its preference list the most preferred agent's rank is 1 and the least preferred agent's rank is k.

In the above previous example, man m_1 is matched to w_1 so rank of the partner of m_1 according to his preference list is 1 whereas the rank of the partner of w_1 according to her preference list if 4.

Any agent that is on the proposed to side might receive more than one proposal in the running of the Gale-Shapley algorithm or gender-neutral matching algorithm so the notation $Prop(a_i, \succ_{a_i}, k)$ is used to denote the k^{th} best proposer ranked according to $\succ_{a_i}$ i.e., the preference list of a_i.

Chapter 3

Applications

David Gale and Lloyd Shapley studied the problem of college admission where colleges have a quota of seats and students apply for them [4]. After evaluating their applications based on the preference rankings of students, colleges have to roll out admissions. Gale and Shapley developed an algorithm which results in stable matches for the SMP.

The New York City High School Match is one of the examples of the application of the Gale-Shapley algorithm. Abdulkadiroğlu et al., (2005) have proposed an algorithm to match students to schools [1]. This has been done by considering schools and students as two different sides of the market who provide their preference over the other side. The primary difference between the Gale-Shapley algorithm and NYC high school matching algorithm is that in the NYC high school matching algorithm no student would receive more than one offer. Also, the NYC high school match accommodates the rules set forth by NYC Department of Education (NYCDOE) such as forcing students to give preference lists for both specialized schools and non-specialized schools.

The algorithm goes as follows:

- Step 1: Every student ranks all the schools in the market and then applies to the most preferred school.
- Step 2: Every school rejects unranked applicants and ranks its most preferred students based on the number of available seats.
- Step 3: If at any step a student gets rejected to the school he applied to, then his application will automatically be sent to the next school on his application.
- Step 4: The algorithm ends when all the students are matched to some school and there are no more rejects.

Algorithm 2: The NYC High School Match Algorithm

Data: Students (S) , High Schools (H), $\succ_C$

Result: μ - final matching

begin

initialize all students to not rejected

while $\exists$ *student who is not matched to a school* **do**

every school rejects unranked applicants and ranks its most preferred students based on the number of available seats

if *student is rejected by a school he/she applied* **then**

student application sent to the next school on his/her application

Following, the success with NYC High School Match, the authors were approached by Boston City to modify their existing algorithm to match students. The early algorithm that was used by Boston to match their students to schools goes as follows:

- Step 1: For each school, the students who have listed it as their first choice, assign students based on the preference of the school until all the seats in the school are filled or there are no more students who have listed the school as their first choice.

.

.

.

- Step k: For each school that still have available seats, consider students who have listed the school as their k^{th} choice and accept the students until either they run out of seats or there are no more students who have listed the school as their k^{th} choice.

Abdulkadiroğlu et al., suggested the same algorithm used by NYC High School Match and also offered a variant of the Top Trading Cycle algorithm for them to implement. The Boston Public School system chose the later and the algorithm [2] goes as follows:

- Step 1: Every school submits the number of seats that are available. Each student points to their favorite school and schools point to their favorite student. For the algorithm to work, there must be at least one cycle like a student s_i points to school c_i, school c_i points to student s_j, student s_j points to school c_j and c_j points to s_i. The cycles are resolved by exchanging the assignments where students get the school they point to in the step. As each student is matched the school seat count is reduced and also the student is removed from the further rounds.

.

.

.

- Step k: Similar to the first step, students and schools point to their top available choice and if a cycle exists, it is resolved.

The algorithm terminates when there are no more cycles or all the students have been placed. The difference between the GS algorithm and Boston matching algorithm is that the GS algorithm results in a stable matching but matching is not *Pareto-efficient* whereas the Boston algorithm is Pareto-efficient but not stable. *Pareto-efficiency* is the state of allocation where no individual's outcome can be improved without worsening at-least one other individual outcome.

Algorithm 3: The Boston Public School Match Algorithm

Data: Students (S) , High Schools (H), $\succ_C$

Result: μ - final matching

begin

initialize all students to unmatched

while $\exists$ *cycles or unmatched student* **do**

schools and students point to their most preferred partner

if $\exists$ *cycle i.e., student* s_i *points to* c_i, c_i *points to student* s_j, s_j *points to school* c_j *and* c_j *points to* s_i **then**

resolve cycle by assigning students to the school they point to in cycle

The Gale-Shapley algorithm is the basis for the algorithmic matching of residents to hospitals - *National Resident Matching Program* [11]. The algorithm begins with matching applicants to the program most preferred according to the resident's preference list or Rank Order List (ROL) as called by NRMP. If a resident is not matched then the algorithm moves on to the second preferred program and finally ends when everyone is matched.

Chapter 4

Related Works

Unbiased Mechanisms are mechanisms that make sure there is no bias towards one side. Various researchers in this domain look at unbiasedness in different ways. While one researcher speaks of unbiasedness by finding a stable matching that is center of all the stable matchings when arranged as a distributive lattice, another researcher speaks of mechanisms as unbiased calling them gender neutral if the matching outcome doesn't change, no matter if you are on the proposing or proposed side. This chapter discusses some of the concepts of unbiasedness established by various prominent researchers.

4.1 Median and Center Stable Matching

Teo and Sethuraman came up with a concept of generalized median stable matching [12]. Let I be an instance of Stable Marriage and $\mathcal{M}$ be the set of its stable matchings where cardinality of $\mathcal{M}$ is n. For every agent in Men set, order his multiset of partners in $\mathcal{M}$ from his most preferred to least preferred. The i^{th} woman in the sorted list is denoted by $\mathcal{M}_{\mathrm{i}}(m)$ for each agent

m in M and set $\alpha_{\mathrm{i}} = \{(m,\mathcal{M}_{\mathrm{i}}(m))\}$ for each man. Similarly we set β_{i} for each woman. Teo and Sethuraman proved that for i = {1, 2,...,n} α_{i} and β_{i} are stable matchings and $\alpha_{\mathrm{i}} = \beta_{\mathrm{n\text{-}i+1}}$ and α_{i} is said to be the i^{th} generalized median stable matching of $\mathcal{I}$. The main important result is the stable matching in middle: $\alpha_{\frac{n+1}{2}}$ when n is odd and $\alpha_{\frac{n}{2}}$, $\alpha_{\frac{n}{2}+1}$ when n is even as all the agents are matched to their lower or upper median stable partners. They proved that this stable marriage is locally fair but Cheng proved that the stable marriages are also globally fair by imposing a graph structure for $\mathcal{M}$ [3]. In her work, Cheng constructs G(I) which is the cover graph of ($\mathcal{M}$,$\succeq$). The relation $\succeq$ on $\mathcal{M}$ is such that $\mu \succeq \mu'$ where μ and μ' are the two matchings in $\mathcal{M}$. Conway noticed that ($\mathcal{M}$,$\succeq$) forms a distributive lattice [7] so Cheng took the cover graph of ($\mathcal{M}$,$\succeq$) i.e., undirected version of its *Hasse diagram*, to prove that the median stable matching is globally fair but the drawback is that computing a median stable marriage is #P-hard [3].

Example: Given an instance $\mathcal{I}$, where M = $\{m_1, m_2, m_3, m_4\}$ and W = $\{w_1, w_2, w_3, w_4\}$.

$$\begin{array}{lllll} \succ_{m_1}: & w_1 & w_2 & w_3 & w_4 \\ \succ_{m_2}: & w_2 & w_1 & w_4 & w_3 \\ \succ_{m_3}: & w_3 & w_4 & w_1 & w_2 \\ \succ_{m_4}: & w_4 & w_3 & w_2 & w_1 \end{array} \qquad \begin{array}{lllll} \succ_{w_1}: & m_4 & m_3 & m_2 & m_1 \\ \succ_{w_2}: & m_3 & m_4 & m_1 & m_2 \\ \succ_{w_3}: & m_2 & m_1 & m_4 & m_3 \\ \succ_{w_4}: & m_1 & m_2 & m_3 & m_4 \end{array}$$

The set of stable matchings obtained using brute force $\mathcal{M}$ in this instance are:

$\mu_1 = \{(m_1,w_1),(m_2,w_2),(m_3,w_3), (m_4,w_4)\}$

$\mu_2 = \{(m_2,w_1),(m_1,w_2),(m_3,w_3), (m_4,w_4)\}$

$\mu_3 = \{(m_1,w_1),(m_2,w_2),(m_4,w_3), (m_3,w_4)\}$

$\mu_4 = \{(m_2,w_1),(m_1,w_2),(m_4,w_3), (m_3,w_4)\}$

$\mu_5 = \{(m_3,w_1),(m_1,w_2),(m_4,w_3), (m_2,w_4)\}$

$\mu_6 = \{(m_2,w_1),(m_4,w_2),(m_1,w_3), (m_3,w_4)\}$

$\mu_7 = \{(m_3,w_1),(m_4,w_2),(m_1,w_3), (m_2,w_4)\}$

$\mu_8 = \{(m_4,w_1),(m_3,w_2),(m_1,w_3), (m_2,w_4)\}$

$\mu_9 = \{(m_3,w_1),(m_4,w_2),(m_2,w_3), (m_1,w_4)\}$

$\mu_{10} = \{(m_4,w_1),(m_3,w_2),(m_2,w_3), (m_1,w_4)\}$

The lattice formed by ($\mathcal{M}$,$\succeq$) where 1, 2, 3, .., 10 correspond to the above stable matches is:

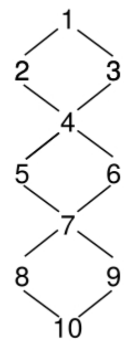

Figure 4.1: Lattice Ordered Set ($\mathcal{M}$,$\succeq$)

Cheng et al. inspired from the work on median stable matching studied another class of fair stable matchings called center stable matchings where the fairness is based on central vertices of $G(I)$. They identified center set of G, which has all the centers of the cover graph i.e., vertices whose distance to other vertices is the smallest.

4.2 Gender-Neutrality

Pini et al. (2011) define gender neutrality as mechanism in which the matching obtained is same even if we change the proposing side and proposed side [8]. In order to achieve this the authors add a preprocessing round to any given instance. This is achieved by computing signature for each gender where signature is a vector of numbers that is built by adding each of preference lists.

From the list of vectors, signature is the lexicographically smallest vector after reordering of the members of one gender and renumbering of the members of other. This vector can be computed in O(n^2). Later, the authors propose a rule called gn-rule [8] which is:

(i) If male signature is smaller than female then we swap men with women

(ii) Else no swap.

After the gn-rule, stable matching procedure can be applied to get a matching.

Example: Given an instance, $M = \{m_1, m_2, m_3\}$ and $W = \{w_1, w_2, w_3\}$.

$$
\begin{array}{llll}
\succ_{m_1}: & w_2 & w_1 & w_3 \\
\succ_{m_2}: & w_3 & w_2 & w_1 \\
\succ_{m_3}: & w_2 & w_1 & w_3
\end{array}
\qquad\qquad
\begin{array}{llll}
\succ_{w_1}: & m_1 & m_2 & m_3 \\
\succ_{w_2}: & m_3 & m_1 & m_2 \\
\succ_{w_3}: & m_2 & m_1 & m_3
\end{array}
$$

Applying the pre-processing round on the given instance, the smallest vector can be obtained by:

1. Swapping men m_2 and m_3

2. Swapping women w_1 and w_2

The signature of men is 123123312 and women is 123213312. According to the gn-rule, we swap men with women as male signature is smaller than female. The instance obtained is:

$\succ_{m_1}$:	w_1	w_2	w_3	$\succ_{w_1}$:	m_1	m_2	m_3
$\succ_{m_2}$:	w_2	w_1	w_3	$\succ_{w_2}$:	m_1	m_2	m_3
$\succ_{m_3}$:	w_3	w_1	w_3	$\succ_{w_3}$:	m_3	m_1	m_2

The above discussed is one of the many ways that researchers define gender-neutrality as in their work.

4.3 Manipulation

The Gale-Shapley algorithm has a proposing and a proposed side where the proposing side gets most optimal partners and proposed side gets most pessimal partners [9]. But the proposed side can manipulate their preference list to obtain a better partner than the partner obtained when they give their true preferences. Alvin E. Roth [10] showed that no mechanism exists in which truth-telling is a dominant strategy for all agents and the proposing side have no incentive in stating false preferences because they get no better than the most optimal solution that they get when they give their true preferences.

For simplicity, consider men are on the proposing side and women are on the proposed side. In this case, the partners of any agent are called the men-optimal partners but in the case that women are on the proposing side and men are on the proposed side, the partners of any agent are called the women-optimal partners.

Let us assume we have an instance I where woman w_i wants to manipulate in order to get a better partner. Let m_i be the men-optimal of w_i and m_j be her women-optimal partner. Pini et al., (2011) [8] established conditions to check if a woman can manipulate her preference list to obtain her women-optimal partner, the following conditions have to be satisfied:

1. In the men proposing Gale-Shapley algorithm she has to receive more than one proposal.

2. There exists a woman w_j whose men-optimal partner is m_j

3. w_j prefers m_i to m_j

4. m_j prefers w_j to w_i

5. m_i prefers w_i to w_j

Teo et al., (2001) [13] devised an algorithm for agents on the proposed side to manipulate to get a better partner (not necessarily partner obtained when on the proposing side) than her partner (if any). Let w_i be the woman who wants to manipulate with preference list $\succ_{w_i}$ and we use the TS manipulation algorithm to find a manipulation strategy in order to obtain a better men-optimal partner.

- Step 1. Run the men propose algorithm and keep a list of all the men who propose to w_i. Let m_i be the men-optimal partner of w_i.

- Step 2. Suppose m_j is a man who is one of the proposers. Obtain a modified preference list ${\succ_{w_i}}^{(1)}$ by moving m_j to the top of $\succ_{w_i}$.

- Step 3. Repeat Step 2 on modified preference list to next man who proposed in the proposer list except m_i. After this, we put m_i to a list called Exhausted Men List.

- Step 4. If m_j is not in the exhausted men list then run the men propose algorithm to add any new proposers as long as they are not in exhausted men list. Repeat Step 2 and Step 3 with m_j instead of m_i.

- Step 5. Repeat Step 4 till all proposers from the proposer list are moved to the exhausted men list.

- Step 6. Choose the modified preference list (if any) which results in a partner who is better than the men-optimal partner while using true preference list.

Algorithm 4: The TS Manipulation Algorithm

Data: M, W, $\succ_{\mathrm{C}} \longleftarrow$ complete set of preference lists of men and women

Result: $\succ'_{w_j}$ - optimal manipulated preference list

begin

$\succ_{w_j} \longleftarrow$ true preference list of w_j

$N \longleftarrow$ set of proposers

$E \longleftarrow$ set of exhausted

${\succ_{w_j}}^{\mathrm{C}} \longleftarrow$ set of manipulated preference lists

run the Gale-Shapley algorithm using $\succ_{w_j}$ add men who propose to w_j in N

repeat

Pick man m_i from N such that $m_i \not\ni E$

${\succ_{w_j}}^{(1)} \longleftarrow$ preference list obtained by moving man to top of w_j's last generated preference list

add ${\succ_{w_j}}^{(1)}$ to ${\succ_{w_j}}^{\mathrm{C}}$

add man to E

run the Gale-Shapley algorithm using ${\succ_{w_j}}^{(1)}$

add proposer to w_j in N such that proposer $\not\ni N$

until $N = E$

Set $\succ'_{w_j} \longleftarrow$ preference list where $\mu'(w_j)$ is most preferred partner according to $\succ'_{w_j}$

Example: Given an instance, $M = \{m_1, m_2, m_3\}$ and $W = \{w_1, w_2, w_3\}$

$\succ_{m_1}$:	w_4	w_1	w_3	w_2	$\succ_{w_1}$:	m_3	m_1	m_4	m_2
$\succ_{m_2}$:	w_1	w_3	w_2	w_4	$\succ_{w_2}$:	m_4	m_2	m_3	m_1
$\succ_{m_3}$:	w_3	w_4	w_1	w_2	$\succ_{w_3}$:	m_1	m_2	m_3	m_4
$\succ_{m_4}$:	w_4	w_2	w_1	w_3	$\succ_{w_4}$:	m_2	m_3	m_1	m_4

The woman w_4 in above example has more than one proposal while running the men propose algorithm so we apply the above algorithm to find a modified preference list that results in a better partner.

- Step 1. Run the men propose algorithm and keep a list of all the men who propose to w_i. Let m_i be the men-optimal partner of w_i.

  So, $\mu = (m_1,w_4)$, (m_2,w_1), (m_3,w_3), (m_4,w_2) and Proposer List = $\{m_3, m_4\}$

- Step 2. Taking the proposer m_4 we obtain a modified preference list ${\succ_{w_4}}^{(1)}$ by moving m_4 to the top of $\succ_{w_4}$. Therefore, ${\succ_{w_4}}^{(1)} = m_4 \succ m_2 \succ m_3 \succ m_1$

- Step 3. Run the men propose algorithm using ${\succ_{w_4}}^{(1)}$ and the new obtained partner of w_4 is m_4 and add the man m_1 and m_4 to the exhausted men list. Also, there are no new proposers added to the proposer list.

Exhausted Men List = $\{m_1, m_4\}$

Proposer List = $\{m_3\}$

- Step 4. Repeat Step 2 for man m_3 and obtain another modified preference list ${\succ_{w_4}}^{(2)}$ by moving m_3 to the top of $\succ_{w_4}$.

  ${\succ_{w_4}}^{(2)} = m_3 \succ m_4 \succ m_2 \succ m_1$

- Step 5. Run the men propose algorithm using ${\succ_{w_4}}^{(2)}$ and the new obtained partner of w_3 is m_3 and add m_3 to the exhausted men list. Also, there are no new proposers added to the proposer list.

  Exhausted Men List = $\{m_1, m_4, m_3\}$

  Proposer List = $\{\}$

- Step 6. From the above two strategies available ${\succ_{w_4}}^{(2)}$ is the best strategy since $m_3 \succ m_1 \succ m_4$ according to $\succ_{w_4}$.

Hence, manipulated preference list of w_4 is $m_3 \succ m_4 \succ m_2 \succ m_1$

Although the manipulation algorithm returns a modified preference list it is completely different from the true preference list. Vaish and Garg (2017) [14] proposed an algorithm that rearranges the modified preference list in order to look like the true preference with at most only one man moved and the authors call this *inconspicuous manipulation*. The algorithm is as follows:

- Step 1. Identify p and q: Let ${\succ_{w_i}}^{(1)}$ be the optimal manipulated preference list of woman w_i. Obtain p and q who are the top 2 proposers from the proposer list that is ordered according to the ${\succ_{w_i}}^{(1)}$ i.e., modified preference list of w_i.

- Step 2. Promoting q: Obtain preference list ${\succ_{w_i}}^{(2)}$ from ${\succ_{w_i}}^{(1)}$ by placing q right after p.

- Step 3. Fixing part above p in the list: Identify the non-proposers above p in $\succ_{w_i}$ and place them above p in ${\succ_{w_i}}^{(2)}$ to obtain ${\succ_{w_i}}^{(3)}$.

- Step 4. Fixing part below q in the list below q: Final step where a list of preference lists $\{{\succ_{w_i}}^{(4)}, {\succ_{w_i}}^{(5)}, ...\}$ are created. The preference list ${\succ_{w_i}}^{(\mathrm{k+1})}$ is obtained from ${\succ_{w_i}}^{(\mathrm{k})}$ by swapping two men $\{m_i, m_j\}$ if
  (i) q $\succ m_i$ and m_j in ${\succ_{w_i}}^{(\mathrm{k})}$
  (ii) $m_i \succ m_j$ in $\succ_{w_i}$ and $m_j \succ m_i$ in ${\succ_{w_i}}^{(\mathrm{k})}$

Algorithm 5: Inconspicuous Manipulation

Data: M, W, $\succ_{\mathrm{C}} \longleftarrow$ complete set of preference lists of men and women, $\succ'_{w_j} \longleftarrow$ manipulated preference list

Result: $\succ''_{w_j}$ - inconspicuous manipulated preference list

begin

$\succ_{w_j} \longleftarrow$ true preference list of w_j

$\succ'_{w_j} \longleftarrow$ manipulated preference list of w_j

Run the Gale-Shapley algorithm when men propose using $\succ'_{w_j}$ and record set P of proposals made to w_j

Set Prop(w_j, $\succ'_{w_j}$, 1) = 'p' and Prop(w_j, $\succ'_{w_j}$, 2) = 'q'

Create ${\succ_{w_j}}^{(1)}$ by moving 'q' to position right after 'p' in $\succ'_{w_j}$

Place M-P agents above 'p' in same order as $\succ_{w_j}$ to create ${\succ_{w_j}}^{(2)}$

repeat

Take a pair of adjacent men (m_i,m_j) below q in ${\succ_{w_j}}^{(2)}$

if *Prop(*w_j*,* ${\succ'_{w_j}}^{(k)}$*, 2)* ${\succ'_{w_j}}^{(k)}$ m_i *and Prop(*w_j*,* ${\succ'_{w_j}}^{(k)}$*,2)* ${\succ'_{w_j}}^{(k)}$ m_j **then**

if $m_i \succ_{w_j} m_j$ *and* m_j ${\succ'_{w_j}}^{(k)}$ m_i **then**

Create ${\succ_{w_j}}^{(\mathrm{k+1})}$ from ${\succ_{w_j}}^{(\mathrm{k})}$ by swapping (m_i,m_j)

until ${\succ_{w_j}}^{(k)} = {\succ_{w_j}}^{(k+1)}$

Set ${\succ_{w_j}}^{(\mathrm{k})}$ as $\succ''_{w_j}$

Example: Given an instance, $M = \{m_1, m_2, m_3, m_4\}$ and $W = \{w_1, w_2, w_3, w_4\}$

$$
\begin{array}{lllll}
\succ_{m_1}: & w_4 & w_1 & w_3 & w_2 \\
\succ_{m_2}: & w_1 & w_3 & w_2 & w_4 \\
\succ_{m_3}: & w_3 & w_4 & w_1 & w_2 \\
\succ_{m_4}: & w_4 & w_2 & w_1 & w_3
\end{array}
\qquad
\begin{array}{lllll}
\succ_{w_1}: & m_3 & m_1 & m_4 & m_2 \\
\succ_{w_2}: & m_4 & m_2 & m_3 & m_1 \\
\succ_{w_3}: & m_1 & m_2 & m_3 & m_4 \\
\succ_{w_4}: & m_2 & m_3 & m_1 & m_4
\end{array}
$$

In the above instance,

True preference list of w_4 is $\succ_{w_4} = m_2 \succ m_3 \succ m_1 \succ m_4$

men-optimal partner of m_4 is $\mu(w_4) = m_1$

Manipulated preference list of w_4 is ${\succ_{w_4}}^{(1)} = m_3 \succ m_4 \succ m_2 \succ m_1$

New men-optimal partner of w_4 is $\mu(w_4) = m_3$

Applying the inconspicuous manipulation algorithm to the above manipulated preference:

- Step 1. Identifying p and q: We obtain proposer list ordered according to ${\succ_{w_4}}^{(1)}$

  Proposer List = $\{m_3, m_4, m_1\}$

  p = m_3

q = m_4

- Step 2. Promoting q: In this example q is right after p, so ${\succ_{w_4}}^{(2)} = m_3 \succ m_4 \succ m_2 \succ m_1$

- Step 3. Fixing part above p in the list: Identify non-proposers above p in $\succ_{w_4}$ and place them above p in ${\succ_{w_4}}^{(2)}$.

  Non-Proposers = $\{m_2\}$

  ${\succ_{w_4}}^{(3)} = m_2 \succ m_3 \succ m_4 \succ m_1$

- Step 4. Fixing part below q in the list below q: Final step where a list of preference lists $\{{\succ_{w_4}}^{(4)}, {\succ_{w_4}}^{(5)}, ...\}$ are created. The preference list ${\succ_{w_4}}^{(\mathrm{k+1})}$ is obtained from ${\succ_{w_4}}^{(\mathrm{k})}$ by swapping two men $\{m_i, m_j\}$ if

  (i) q $\succ m_i$ and m_j in ${\succ_{w_i}}^{(\mathrm{k})}$

  (ii) $m_i \succ m_j$ in $\succ_{w_4}$ and $m_j \succ m_i$ in ${\succ_{w_4}}^{(\mathrm{k})}$

There are no men to swap in this example but post swapping the obtained inconspicuous preference list is

$${\succ_{w_4}}^{(4)} = m_2 \succ m_3 \succ m_4 \succ m_1$$

Comparing $\succ_{w_4}$ and ${\succ_{w_4}}^{(3)}$, it can be seen that only man m_4 has been moved and the men-optimal partner doesn't change from the optimal manipulated preference list.

Chapter 5

Results

This chapter will discuss the design and implementation of the gender-neutral matching algorithm for two-sided matching. Later, the algorithm is analyzed to see if agents participating in the algorithm can manipulate it to obtain a better outcome.

5.1 Gender-Neutral Algorithm

Let *men* M and *women* W be the two disjoint sets of the stable marriage problem and $\succ_{\mathrm{C}}$ be the complete set of preference lists of all agents participating in the problem. The gender-neutral algorithm begins with using the proposer function $P(M, W)$, which is a fair coin toss with $p = 0.5$, that decides which side gets to be the proposing side. For the sake of convenience, let us assume that the proposer function decides to put the women on the proposing side which in turn implies that the men are on the proposed side. Then in the first round, each unmatched woman proposes to the man she prefers most and if the man is unmatched, he is matched to that woman.

But if the man is matched to another woman, he evaluates the woman he is matched to currently, with the woman who proposed to him and matches according to his preference list to the most preferred partner. In each subsequent round, the unmatched women repeat the process of proposing to her most preferred man whom she has not yet proposed. The process is repeated until everyone is matched to a partner from the other side. The distinction between the Gale-Shapley algorithm and the gender-neutral algorithm is that in the Gale-Shapley algorithm the men are always proposers whereas in the gender-neutral algorithm both sides have an equal probability of being the proposers.

Algorithm 6: The Gender-Neutral Algorithm

Data: M, W, $\succ_{\mathrm{C}}$

Result: μ - final matching

begin

$P(M, W) \longleftarrow$ proposer function that decides proposing side

```
/* Let us assume the proposer function outputs women
   as the proposers                                  */
```

initialize all men and women on both sides to unmatched and set women to proposers

while $\exists$ *some woman* w_j *is unmatched and hasn't proposed to every man on other side* **do**

$m_i \longleftarrow$ most preferred agent on proposer list to whom proposer hasn't yet proposed

if m_i *is unmatched* **then**

mark m_i and w_j as matched

else if m_i *prefers* w_j *to current partner* w_k **then**

mark m_i and w_j as matched and w_k as unmatched

else

m_i rejects w_j and mark w_j proposed to m_i

```
/* If the men are shown as proposers by the proposer
   function then flip men and women in the algorithm
   */
```

In this research, the proposer function ensures that both the men and women have equal probability of being on the proposing side. This gender-neutral algorithm fixes the problem of bias while returning the stable matches as required.

Example: Given an instance, $M = \{m_1,\ m_2,\ m_3,\ m_4,\ m_5\}$ and $W = \{w_1,\ w_2,\ w_3,\ w_4,\ w_5\}$

$$
\begin{array}{llllll}
\succ_{m_1}: & w_1 & w_5 & w_4 & w_2 & w_3 \\
\succ_{m_2}: & w_5 & w_1 & w_2 & w_3 & w_4 \\
\succ_{m_3}: & w_2 & w_3 & w_1 & w_4 & w_5 \\
\succ_{m_4}: & w_5 & w_1 & w_2 & w_3 & w_4 \\
\succ_{m_5}: & w_4 & w_5 & w_1 & w_2 & w_3
\end{array}
\qquad
\begin{array}{llllll}
\succ_{w_1}: & m_5 & m_2 & m_3 & m_1 & m_4 \\
\succ_{w_2}: & m_4 & m_2 & m_3 & m_1 & m_5 \\
\succ_{w_3}: & m_5 & m_3 & m_2 & m_1 & m_4 \\
\succ_{w_4}: & m_4 & m_1 & m_2 & m_5 & m_3 \\
\succ_{w_5}: & m_1 & m_5 & m_2 & m_4 & m_3
\end{array}
$$

In order to find the stable matches in the given instance, we apply the gender-neutral algorithm.

- Step 1: Run the proposer function and decide the proposer. For now assume the proposer function decides that the women are proposers i.e., $P(M, W) = W$
- Step 2: Take an unmatched woman from W and propose to most preferred agent i.e., w_1 proposes to m_5 and since m_5 is unmatched (m_5,w_1) are matched and w_1 is set to matched.

- Step 3: Next, w_2 proposes to m_4 and since m_4 is unmatched (m_4,w_2) are matched and w_2 is set to matched.

- Step 4: w_3 proposes to m_5 but since m_5 is matched he evaluates between his current partner and proposer i.e., w_1 and w_3. In the preference list of m_5 since w_1 ranks better than w_3 the proposal by w_3 is rejected.

- Step 5: w_3 now proposes to next man m_3 and since m_3 is unmatched (m_3,w_3) are matched and w_3 is set to matched.

- Step 6: w_4 proposes to m_4 and since m_4 is matched and m_4 prefers w_2 over w_4 the proposal is rejected.

- Step 7. w_4 proposes to next man m_1 and since m_1 is unmatched (m_1,w_4) are matched and w_4 is set to matched.

- Step 8. w_5 proposes to m_1 and m_1 prefers w_5 over current partner w_4 so (m_1,w_5) are matched and w_4 is set back to unmatched.

- Step 9. w_4 proposes to most preferred man who she hasn't proposed yet i.e., m_2 and since w_2 is unmatched (m_2,w_4) are matched and w_4 is set to matched.

- Step 10. The algorithm terminates as all women are matched.

The matching obtained when women propose above is $\mu_{\mathrm{w}} = \{(w_1,m_5)$, (w_2,m_4), (w_3,m_3), (w_4,m_2), $(w_5,m_1)\}$.

In the above example if the proposer function decides that men are the proposers then the matching obtained is $\mu_{\mathrm{m}} = \{(m_1,w_1)$, (m_2,w_5), (m_3,w_3), (m_4,w_2), $(m_5,w_4)\}$.

The following is a comparison of the properties of the Gale-Shapley algorithm and the gender-neutral algorithm:

Table 5.1: Comparison of the Gale-Shapley (Men Propose) and the Gender-Neutral algorithm

	Gale-Shapley Algorithm	Gender-Neutral Algorithm
Stable Matching	Yes	Yes
Men Optimal	Yes	Yes (If Proposer)
Women Optimal	No	Yes (If Proposer)
Men Manipulable	No	?
Women Manipulable	Yes	?

It is known that in the Gale-Shapley algorithm, women can manipulate to receive better outcomes so we sought after the question "Although no agent from either set knows which side they are going to be on, is it possible for an agent to manipulate their preferences such that he/she improves their outcome if on the proposed side since the proposing side always receives their optimal partner?"

5.2 Manipulation using TS Manipulation

In order to investigate the above problem, the works of Teo et al., (Refer 4.3 - TS manipulation algorithm) have been applied to the gender-neutral algorithm [13]. In the TS manipulation algorithm, the agent who wishes to manipulate must have more than one proposal in the series of proposals made during the gender-neutral algorithm. Since both men and women have an equal probability of being on the proposed side, we investigate the possibility of manipulation, assuming women as the manipulators for convenience.

In the TS manipulation, it is assumed that the manipulator is on the proposed side when using the Gale-Shapley algorithm and the rank of the partner obtained is improved but there is an equal probability that the manipulator can also be on the proposing side when using the gender-neutral algorithm. So the newly obtained preference list has been used to obtain the matching when manipulator is on the proposing side and this showed that although the partner of manipulator improves if, on the proposed side, the partner of manipulator worsens if on the proposing side. Also, the partner obtained when on proposing and proposed side is same for the manipulator.

The matching obtained when men are the proposers in the gender-neutral algorithm is represented by μ_{m} and when women are the proposers by μ_{w}. The notations μ'_{m} and μ'_{w} are used to denote the matchings obtained using the gender-neutral algorithm when manipulators use TS manipulation when men and women propose respectively.

Theorem 1 (Teo et al., 2001). Let w_j be a manipulator using TS manipulation. If w_j is on the proposed side in the gender-neutral algorithm, her partner improves compared to using her true preferences. Formally, $\mu'_m(w_j) \succeq \mu_m(w_j)$.

Later, we observed that the partner obtained by manipulator when he/she is on proposing side or proposed side is same and we prove this in Theorem 2.

Theorem 2. Let w_j be a manipulator using TS manipulation. Let m_r be the partner obtained by w_j if women are on the proposed side and m_s be the partner obtained by w_j if women are on the proposing side then, $m_r = m_s$. Formally, $\mu'_m(w_j) = \mu'_w(w_j)$

Proof. Gale and Shapley have proven that an agent receives better partner if on the proposing side than if on the proposed side [4]. So,

(i) $\mu_w(w_j) \succeq \mu_m(w_j)$

(ii) $\mu'_w(w_j) \succeq \mu'_m(w_j)$ or $m_s \succeq_{w_j} m_r$

The TS manipulation algorithm iterates the manipulator preference list such that the partner obtained when on the proposed side is at the top of the manipulator preference list, in other words, m_r is placed at the top of the manipulator preference list.

(iii) $\succ'_{w_j} = m_r \succ (M$ - $m_r)$

(iv) $r(m_r) = 1$

From (ii) we know that, $r(m_s)$ should be lower than or equal to r(m_r) but a rank lower than 1 is not possible therefore, $m_s = m_r$. $\square$

Based on the above result, it can be inferred that the partner might worsen if the manipulator is on the proposing side in the gender-neutral algorithm and we prove this result in Theorem 3.

Theorem 3. Let w_j be manipulator using TS manipulation, if w_j is on the proposing side in the gender-neutral algorithm, her partner might worsen compared to using her true preferences. Formally, $\mu_w(w_j) \succeq \mu'_w(w_j)$.

Proof. It is known that the Gale-Shapley algorithm is strategy-proof for proposers and the gender-neutral algorithm has same properties as Gale-Shapley as the matching iterations are the same [4,10]. So woman w_j can't improve her partner by manipulating as proposer receives his/her most optimal partner. Therefore, $\mu_w(w_j) \succeq \mu'_w(w_j)$. $\square$

Example: Given an instance

$M = \{m_1, m_2, m_3, m_4, m_5\}$ and $W = \{w_1, w_2, w_3, w_4, w_5\}$

$$
\begin{array}{llllll}
\succ_{m_1}: & w_1 & w_5 & w_4 & w_2 & w_3 \\
\succ_{m_2}: & w_5 & w_1 & w_2 & w_3 & w_4 \\
\succ_{m_3}: & w_2 & w_3 & w_1 & w_4 & w_5 \\
\succ_{m_4}: & w_5 & w_1 & w_2 & w_3 & w_4 \\
\succ_{m_5}: & w_4 & w_5 & w_1 & w_2 & w_3
\end{array}
\qquad
\begin{array}{llllll}
\succ_{w_1}: & m_5 & m_2 & m_3 & m_1 & m_4 \\
\succ_{w_2}: & m_4 & m_2 & m_3 & m_1 & m_5 \\
\succ_{w_3}: & m_5 & m_3 & m_2 & m_1 & m_4 \\
\succ_{w_4}: & m_4 & m_1 & m_2 & m_5 & m_3 \\
\succ_{w_5}: & m_1 & m_5 & m_2 & m_4 & m_3
\end{array}
$$

The matching obtained when men propose is $\mu_{\mathrm{m}} = \{(m_1,w_1), (m_2,w_5), (m_3,w_3), (m_4,w_2), (m_5,w_4)\}$

The matching obtained when women propose is $\mu_{\mathrm{w}} = \{(w_1,m_5), (w_2,m_4), (w_3,m_3), (w_4,m_2), (w_5,m_1)\}$

Let us assume w_1 is the woman manipulating using the TS manipulation algorithm.

- Step 1. Run the men propose gender-neutral algorithm. The matching obtained is $\{(m_1,w_1), (m_2,w_5), (m_3,w_3), (m_4,w_2), (m_5,w_4)\}$.

- Step 2. The men proposing to w_1 in the matching are m_1 and m_4.Add m_4 and m_1 to N and since the preference list already matched m_1 to w_1 add man m_1 to E.

- Step 3. The men in N-E are $\{m_4\}$ so move man m_4 to the top of $\succ_{w_j}$ to generate ${\succ_{w_j}}^{(1)} =$ and add ${\succ_{w_j}}^{(1)}$ to ${\succ_{w_j}}^{\mathrm{C}}$.

- Step 4. Run the men propose gender-neutral algorithm. The matching obtained is $\{(m_1,w_5), (m_2,w_3), (m_3,w_2), (m_4,w_1), (m_5,w_4)\}$.

- Step 5. The men proposing to w_1 in the above step are m_1, m_4 and m_2 so add m_2 to N and m_4 to E.

- Step 6. Next move man m_2 to the top of ${\succ_{w_j}}^{(1)}$ to generate ${\succ_{w_j}}^{(2)} = \{m_2 \succ m_4 \succ m_5 \succ m_3 \succ m_1\}$ and add ${\succ_{w_j}}^{(2)}$ to ${\succ_{w_j}}^{\mathrm{C}}$.

- Step 7. Run the men propose gender-neutral algorithm. The matching obtained is $\{(m_1,w_5), (m_2,w_1), (m_3,w_3), (m_4,w_2), (m_5,w_4)\}$.

- Step 8. The men proposing to w_1 in the above step are m_1, m_4 and m_2 so add m_2 to E.

- Step 9. Since $N = E$ the process is stopped.

- Step 10. Set ${\succ_{w_j}}^{(2)}$ as $\succ'_{w_j}$ since w_2 is the most preferred partner according to $\succ_{w_j}$ and the algorithm terminates.

The optimal manipulated preference list obtained using the TS manipulation algorithm is $\succ'_{w_1} = m_2 \succ m_4 \succ m_5 \succ m_3 \succ m_1$.

The matching obtained when men propose when w_1 manipulates is $\mu'_{\mathrm{m}} = \{(m_1,w_5), (m_2,w_1), (m_3,w_3), (m_4,w_2), (m_5,w_4)\}$

The matching obtained when women propose is $\mu'_{\mathrm{w}} = \{(w_1,m_2), (w_2,m_3), (w_3,m_5), (w_4,m_5), (w_5,m_1)\}$

The partner obtained by w_1 when men propose with true preference list is m_1 so rank of partner $r(\mu_{\mathrm{m}}(w_1)) = 4$ and the partner obtained by w_1 when women propose with true preference list is m_5 so $r(\mu_{\mathrm{w}}(w_1)) = 1$. But the partner obtained by w_1 when men propose with the manipulated preference list is m_2 so $r(\mu'_{\mathrm{m}}(w_1))$ and the partner obtained by w_1 when women propose with the manipulated preference list is w_2 so $r(\mu'_{\mathrm{w}}(w_1)) = 2$. So, if $p \times$ (manipulated partner - true partner, proposing) + $(1-p) \times$ (manipulated partner - true partner, proposed) is greater that zero, expected gain of manipulation is positive, meaning that it's beneficial to manipulation. If negative, means that expected gain from manipulation is negative...So here, you would say expected gain = .5 * 1 - .5*2 = -0.5, meaning that manipulation is not beneficial.

Table 5.2: Rank of the partners obtained using the Gender-Neutral Algorithm with true preference list and TS manipulated preference list

Manipulator : w_1	Partner Rank (Men Propose)	Partner Rank (Women Propose)
True Preference List ($\succ_{w_1}$)	4	1
TS Manipulated Preference List ($\succ'_{w_1}$)	2	2

Note: The rank of the partner obtained by w_1 under various cases is based on the true preference list of w_1.

The stable matching obtained by the use of TS manipulation in the gender-neutral algorithm did improve the partner when participant is proposed to but there is a risk of getting a worse partner using manipulated preference list than the partner obtained using true preference list when proposing. So the question in front of us now is *"Can a participant in this gender-neutral algorithm manipulate such that he/she improves their partner when on proposed side but doesn't want to get a worse partner if they happen to be on the proposing side?"*.

5.3 Matching using Inconspicuous Manipulation

To answer the above question, the works of Vaish and Garg (2017) [14] (Refer 4.3 - Inconspicuous manipulation algorithm) has been applied to manipulate the gender-neutral algorithm.

The preference list ${\succ''}_{w_j}$ obtained using the inconspicuous manipulation algorithm is set as the manipulated preference of manipulator and upon running the gender-neutral matching algorithm it was seen that when the manipulator is on the proposed side the partner obtained is improved whereas the partner obtained if on the proposing side is the same partner obtained using true preference list by manipulator when proposing. The matching obtained when men are the proposers in the gender-neutral algorithm is represented by μ_{m} and when women are the proposers by μ_{w}. The notations ${\mu''}_{\mathrm{m}}$ and ${\mu''}_{\mathrm{w}}$ are used to denote the matching obtained using gender-neutral algorithm when manipulators use inconspicuous manipulation when men and women propose respectively.

Theorem 4 (Vaish and Garg 2017). Let w_j be a manipulator using inconspicuous manipulation. If w_j is on the proposed side in the gender-neutral algorithm, her parter improves compared to her true preferences and is the same partner obtained when using TS manipulation. Formally, ${\mu''}_m(w_j) \succeq$

$\mu_m(w_j)$ and $\mu''_m(w_j) \succeq \mu'_m(w_j)$

Theorem 5. Let w_j be a manipulator using inconspicuous manipulation. If w_j is on the proposing side in the gender-neutral algorithm, the partner obtained by manipulating is same as partner obtained using true preferences. Formally, $\mu''_w(w_j) = \mu_w(w_j)$.

Proof. Let m_q be w_j partner when on proposed side and m_p partner when on proposing side obtained using true preferences in the gender-neutral algorithm. Let m_s be w_j partner when on proposed side and m_r partner when on proposing side obtained using inconspicuous manipulation in the gender-neutral algorithm.

According, to the inconspicuous manipulation algorithm, the non proposers above m_s are placed in same order as true preference list.

(i) $\succ_{w_j} = ...\ m_p\ ....\ m_q\ ....$

(ii) $\succ''_{w_j} = ...\ m_p\ ..\ m_s\ ....$

In the gender-neutral algorithm w_j doesn't propose to men beyond m_p as he is the most optimal partner of w_j under true preferences. All the men until m_p are non proposers in both $\succ_{w_j}$ and $\succ''_{w_j}$. So, m_p is the partner obtained by w_j using manipulated preference list and $m_p = m_r$. Therefore, $\mu''_w(w_j) = \mu_w(w_j)$. $\square$

Example: Given same instance as above

$M = \{m_1, m_2, m_3, m_4, m_5\}$ and $W = \{w_1, w_2, w_3, w_4, w_5\}$

$\succ_{m_1}$:	w_1	w_5	w_4	w_2	w_3	$\succ_{w_1}$:	m_5	m_2	m_3	m_1	m_4
$\succ_{m_2}$:	w_5	w_1	w_2	w_3	w_4	$\succ_{w_2}$:	m_4	m_2	m_3	m_1	m_5
$\succ_{m_3}$:	w_2	w_3	w_1	w_4	w_5	$\succ_{w_3}$:	m_5	m_3	m_2	m_1	m_4
$\succ_{m_4}$:	w_5	w_1	w_2	w_3	w_4	$\succ_{w_4}$:	m_4	m_1	m_2	m_5	m_3
$\succ_{m_5}$:	w_4	w_5	w_1	w_2	w_3	$\succ_{w_5}$:	m_1	m_5	m_2	m_4	m_3

The matching obtained when men propose is $\mu_{\mathrm{m}} = \{(m_1,w_1), (m_2,w_5), (m_3,w_3), (m_4,w_2), (m_5,w_4)\}$

The matching obtained when women propose is $\mu_{\mathrm{w}} = \{(w_1,m_5), (w_2,m_4), (w_3,m_3), (w_4,m_2), (w_5,m_1)\}$

Since w_1 is the woman who manipulated using the TS manipulation algorithm, we take the optimal manipulated preference list obtained and apply the inconspicuous algorithm.

- Step 1. Run the men propose gender-neutral algorithm using $\succ'_{w_1}$ and record the proposals made to w_1 in P. $P = \{m_1, m_2, m_4\}$.

- Step 2. Identify $Prop(w_1, \succ'_{w_1}, 1)$ and $Prop(w_1, \succ'_{w_1}, 2)$ as 'p' and 'q' respectively. Therefore, 'p' $= m_2$ and 'q' $= m_1$.

- Step 3. Create ${\succ_{w_1}}^{(1)}$ by moving 'q' right after 'p' in $\succ'_{w_1}$. ${\succ_{w_1}}^{(1)} = \{m_2 \succ m_1 \succ m_4 \succ m_5 \succ m_3 \}$

- Step 5. Place M-P agents above 'p' in same order as $\succ_{w_1}$ to create ${\succ_{w_1}}^{(2)} = \{m_5 \succ m_3 \succ m_2 \succ m_1 \succ m_4 \}$

- Step 6. Take a pair of adjacent men below 'q' i.e., $\{m_1,m_4\}$ and check to see if they need to be swapped but since they don't satisfy the conditions, there will be no swapping done.

- Step 7. Set ${\succ_{w_1}}^{(2)}$ as $\succ''_{w_1}$ and the algorithm terminates.

The preference list that is obtained after applying the inconspicuous algorithm is $\succ''_{w_1} = m_5 \succ m_2 \succ m_4 \succ m_3 \succ m_1$.

The matching obtained when men propose when w_1 manipulates is $\mu''_{\mathrm{m}} = \{(m_1,w_5), (m_2,w_1), (m_3,w_3), (m_4,w_2), (m_5,w_4)\}$

The matching obtained when women propose is $\mu''_{\mathrm{w}} = \{(w_1,m_5), (w_2,m_4), (w_3,m_3), (w_4,m_2), (w_5,m_1)\}$

Here, μ_{w} and μ''_{w} are the same matching outcome which means that the partner obtained by w_1 upon using the inconspicuous manipulation algorithm is same as partner obtained using true preference list so the rank of partner doesn't get any worse when she manipulates whereas in the case of TS manipulation it did get worse. In the case of men proposing when woman w_1 doesn't manipulate, the rank of partner obtained is $r(\mu_{\mathrm{m}}(w_1)) = 4$ but when she manipulates using the inconspicuous manipulation the rank of partner obtained is $r(\mu''_{\mathrm{m}}(w_1)) = 2$. Hence, the woman w_1 is better off if she manipulates using the inconspicuous algorithm.

Table 5.3: Ranks of partner obtained using the Gender-Neutral Algorithm with true preference list, TS manipulated and inconspicuous manipulated preference list

Manipulator : w_1	Partner Rank (Men Propose)	Partner Rank (Women Propose)
True Preference List ($\succ_{w_1}$)	4	1
TS Manipulated Preference List ($\succ'_{w_1}$)	2	2
Inconspicuous Manipulated Preference List ($\succ''_{w_1}$)	2	1

Hence, when using the gender-neutral algorithm, it is beneficial to use the inconspicuous manipulation algorithm unlike the TS manipulation algorithm.

Chapter 6

Empirical Evaluations

We conducted empirical evaluations in order to corroborate our results. We generated a preference list for each agent in the stable marriage problem by initializing the list of agents on other side and shuffling them using the *shuffle* function in Python. The instance size is equal to the total number of agents on both sides. For each size, we took 1,000 example instances in order to verify the results accurately.

6.1 Manipulators

Initially, we ran evaluations to check the manipulators who can manipulate using inconspicuous manipulation in every instant size. For each of the instance size, we generated 1,000 instances and then ran the gender-neutral algorithm to find the stable matching for each instance. Later, we used the inconspicuous manipulation algorithm to find all the individual manipulators across the instance. It is important to note that we assume that manipulator believes that he/she is the sole manipulator and manipulates his/her preference list. Finally, we averaged the total number of manipulators per instance size and plotted the average percentage of manipulators per instance size (Figure 6.1) and the average number of manipulators per instance size (Figure 6.2).

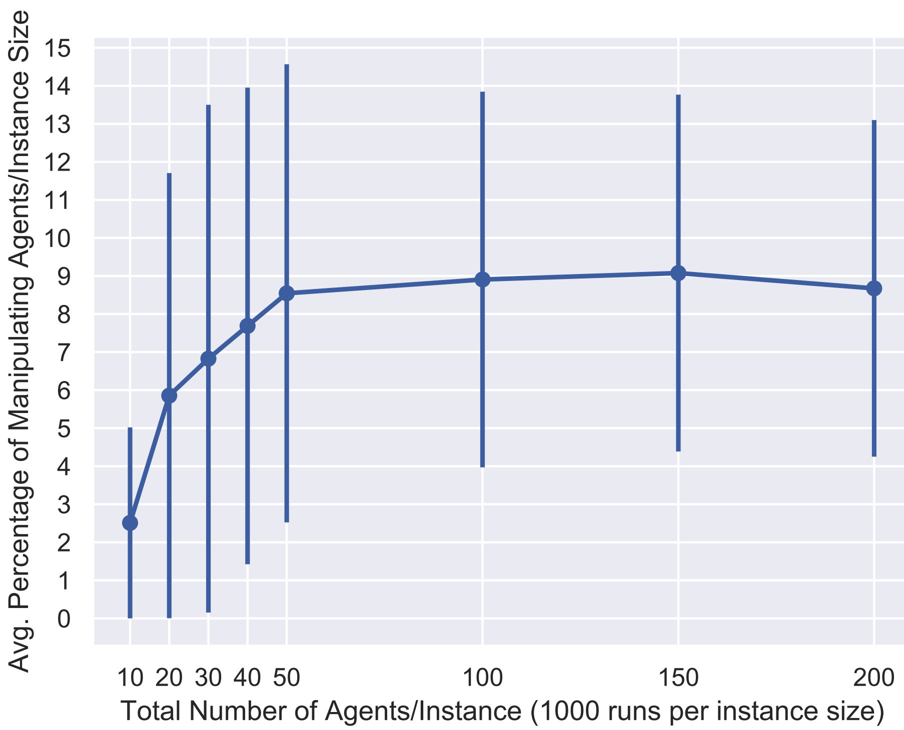

Figure 6.1: Avg. percentage of manipulators per instance size

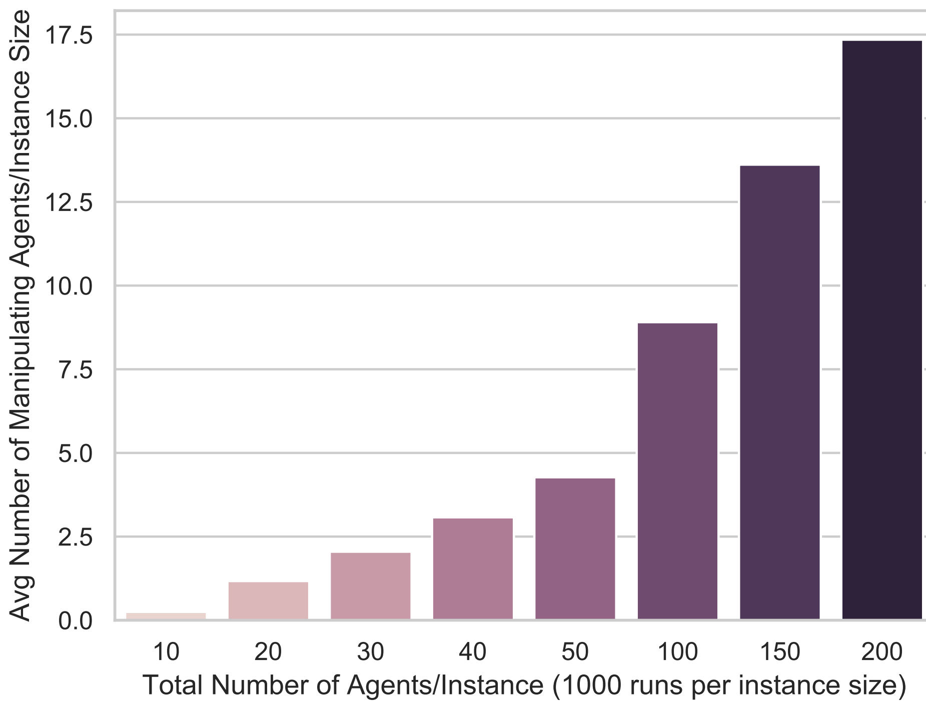

Figure 6.2: Avg. number of manipulators per instant size

It can be seen that, as the instance size increases from 10 to 50, the average percentage of manipulators increases rapidly but from 50 to 200 is around 9%. This shows that there are agents who can successfully manipulate the gender-neutral algorithm to improve their outcome. But it has to be noted that if more than one agent manipulates at the same time, the outcome of manipulation might not remain same as each manipulator improves his outcome based on the true preference list of the other manipulator.

6.2 Expected Rank Gain (ERG)

In this section, we evaluated the expected rank gain for manipulators using the TS manipulation algorithm. The expected rank gain for manipulator has been computed by multiplying the probability of agent being on the proposing side with the difference between partner rank obtained from manipulation and partner rank obtained without manipulation and then multiplying the probability of agents being on the proposed side with the difference between partner rank from manipulation and partner rank obtained without manipulation. The probabilities have been set from 0 to 1 with an increment of 0.25.

$ERG = p \times$ (manipulated partner - true partner, proposing) $+ (1 - p) \times$ (manipulated partner - true partner, proposed)

where $p =$ probability of being on the proposing side

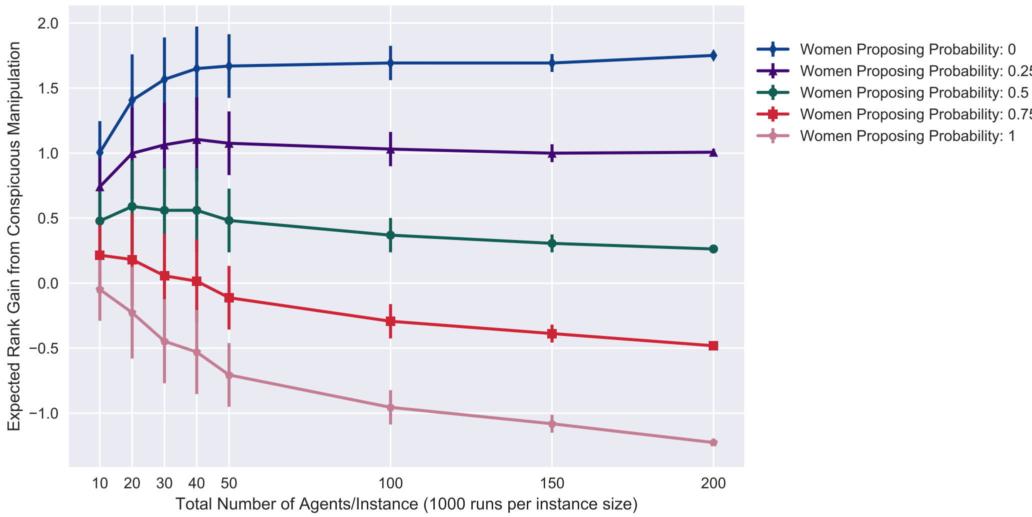

Figure 6.3: Expected rank gain for manipulators

It can be seen that as the proposing probability of women increases from 0 to 1 the expected rank gain reduces drastically. This implies that upon manipulation, the agent rank improves if he/she is on the proposed side but worsens if on the proposing side. This corroborates Theorem 3 which has been proved in Chapter 5.

Chapter 7

Conclusion

The results that have been obtained, help us understand that adding a probability in deciding the proposers will never truly make the Gale-Shapley algorithm fair as agents can still manipulate. It can be clearly seen that although agents could either be on the proposing or proposed side, they can always manipulate in such a way that they improve their partner if on the proposed side while retaining the same partner if on the proposing side. This proves there is an incentive for agents to manipulate which leads to strategic behavior of the agents. The future work lies in taking multiple mechanisms in two-sided matching and applying the different manipulation algorithms if there is a mechanism which is not manipulable and if there is such a mechanism it would be interesting to design a manipulation algorithm for it. Also, another direction would be to design a matching algorithm where instead of setting all agents on one side to proposers, we could flip a coin for every agent to decide whether he/she is a proposer or not.

Bibliography

[1] Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. The New York City High School Match. *American Economic Review*, 95(2):364–367, April 2005.

[2] Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. The Boston Public School Match. *American Economic Review*, 95(2):368–371, April 2005.

[3] Christine T. Cheng. Understanding the Generalized Median Stable Matchings. *Algorithmica*, 58(1):34–51, September 2010.

[4] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[5] Yuichiro Kamada and Fuhito Kojima. Improving Efficiency in Matching Markets with Regional Caps: The Case of the Japan Residency Matching Program. *Discussion Papers, Stanford Institute for Economic Policy Research*, page 55, 2010.

[6] Yuichiro Kamada and Fuhito Kojima. Efficient Matching under Distributional Constraints: Theory and Applications. *The American Economic Review*, 105(1):67–99, 2015.

[7] Donald Ervin Knuth. *Mariages stables et leurs relations avec d&autres problèmes combinatoires.* Presses de l'Université de Montréal, 1976.

[8] Maria Silvia Pini, Francesca Rossi, K. Brent Venable, and Toby Walsh. Manipulation complexity and gender neutrality in stable marriage procedures. *Autonomous Agents and Multi-Agent Systems*, 22(1):183–199, January 2011.

[9] Alvin E Roth. Deferred Acceptance Algorithms: History, Theory, Practice, and Open Questions. page 39.

[10] Alvin E. Roth. The Economics of Matching: Stability and Incentives. *Mathematics of Operations Research*, 7(4):617–628, 1982.

[11] 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(4):33, 1999.

[12] Chung-Piaw Teo and Jay Sethuraman. The Geometry of Fractional Stable Matchings and Its Applications. *Mathematics of Operations Research*, 23(4):874–891, November 1998.

[13] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. Gale-Shapley Stable Marriage Problem Revisited: Strategic Issues and Applications. *Management Science*, 47(9):1252–1267, September 2001.

[14] Rohit Vaish and Dinesh Garg. Manipulating Gale-Shapley Algorithm: Preserving Stability and Remaining Inconspicuous. pages 437–443. Inter-

national Joint Conferences on Artificial Intelligence Organization, August 2017.