2016

Stable Marriage Problem

Angelica Ventura
Long Island University, angventu@gmail.com

Stable Marriage Problem

An Honors College Thesis

by

Angelica Ventura

Fall 2016

Department of Mathematics

Dr. Jozsef Losonczy, Advisor

Dr. Sheldon Rothman, Reader

Date

Contents

Abstract

Every instance of the Stable Marriage Problem involves two finite sets of equal size. We can think of one of the sets as containing men and the other as containing women. Each person must rank the members of the opposite gender in their order of preference. The goal is then to create a set of man-woman couples with the following stability property: it is impossible to find a man and a woman who prefer each other over their respective partners in the set of couples. A set of couples having this property is called a stable matching. Such matchings can be found using the Gale-Shapley algorithm. In this thesis, we discuss the history of the Gale-Shapley algorithm. We also state and prove some theorems which establish the most important properties of the algorithm. We provide many examples in order to demonstrate how the algorithm works.

Chapter 1

Introduction

There is a lot of history that leads up to the Gale-Shapley Algorithm. From the early 1900's up until 1945, the labor market for medical interns suffered a Prisoner's Dilemma problem in which there was a competition amongst hospitals for interns. It was a race to hire medical students earlier and earlier in their medical school career. There were many problems with how hospitals were looking to get medical students.

For starters, hospitals had more positions open than the number of students graduating. This was a big reason for why hospitals would compete for students. However, allowing students in earlier would be more expensive. The hospital would not be aware of the student's final grades in their courses and thus the students could stop caring about their grades since they already had a job. To fix this, it was agreed that student information would not be released until a set date. In this way, all hospitals would be given a fair chance. By 1944, appointments for interviews would take place during a student's junior year.

Another issue was the waiting period between the time when offers of internships were made and the time students were required to accept. If a student was waitlisted at one's first hospital choice, but accepted at one's second choice, one would wait to hear back from one's first choice. Students would be waiting too long to respond to hospitals.

The solution to this was students would have only 10 days to respond to hospitals after being offered a position. However, while both of the above fixes seemed like good solutions, it was quickly realized that it was not good enough to ultimately fix the whole problem.

Thus, the solution was the trial-run algorithm. The students would rank hospital programs in order of preference, which they applied to and hospitals would rank their applicants. Both parties would submit these ranking to a bureau, which would then use this information to arrange a matching of students to hospitals and inform the parties of the results. Thus a specific algorithm was proposed to produce a matching from the submitted ranks. This ultimately became known as the National Intern Matching Program in 1953.

Many years later it was discovered that this was essentially the Gale-Shapley algorithm, which was published in 1962 [1]. The only difference is that the latter finds an optimal matching for the hospital rather than the residents. We know from the Gale-Shapley algorithm, that there will exist at least one stable matching in an instance of the stable marriage problem. This algorithm always finds a stable matching.

Chapter 2

Definitions and Examples

In this chapter a few important key definitions to better understand the *Stable Marriage Problem* will be given. I will provide an example for each definition in order to better comprehend what the definition is stating. I will also include a few theorems which will be proven in Chapter 4.

Definition 1. An instance of the Stable Marriage Problem of size n involves two disjoint sets, one containing n men and the other containing n women. Each man and each woman has a strictly ordered preference list, ranking all of the people of the opposite sex. We say that person p prefers q to r (q and r are of the opposite sex from p) if q is ranked higher than r on p's preference list.

Example 1. Consider the situation where the set of men is

$$\{Cole, Jack, Ken, Larry\},$$

and the set of women is

$$\{Gail, Heather, Jane, Maggie\}.$$

Preference lists for men and women are given in the two tables below.

Cole	Jack	Ken	Larry
Heather	Gail	Gail	Maggie
Jane	Jane	Maggie	Heather
Maggie	Heather	Jane	Gail
Gail	Maggie	Heather	Jane

Table 2.1: Men's Preferences

Gail	Heather	Jane	Maggie
Cole	Jack	Ken	Ken
Larry	Cole	Jack	Larry
Ken	Larry	Larry	Jack
Jack	Ken	Cole	Cole

Table 2.2: Women's Preferences

We see that Cole's first choice is Heather, his second choice is Jane, his third is Maggie, and his fourth is Gail. The preferences of the others are read in the same way.

The above is an instance of the Stable Marriage Problem of size 4.

Definition 2. A *matching* for an instance of the stable marriage problem of size n is a set M consisting of exactly n man-woman pairs (m, w), with each man appearing in exactly one pair, and likewise for each woman. If man m and woman w are matched in M, then m and w are called *partners* in M, and we write $m = P_M(w)$, $w = P_M(m)$. Thus, $P_M(m)$ is the M-partner of m, and $P_M(w)$ the M-partner of w.

Example 2. Looking back at Example 1, one matching would be

$$\{(Cole, Heather), (Jack, Gail), (Ken, Maggie), (Larry, Jane)\}.$$

Definition 3. Suppose that M is a *matching* for an instance of the stable marriage problem. A *blocking pair* for M consists of a man m and woman w such that m and w are not partners in M, but m prefers w to $P_M(m)$ and w prefers m to $P_M(w)$. We say that M is *stable* if there is no blocking pair for M.

The basic problem we address in the thesis is the following:

> For each instance of the Stable Marriage Problem, is it possible to find a stable matching?

In Chapter 3, we answer the above question in the affirmative. A procedure called the Gale-Shapley Algorithm will be presented and used to find stable matchings.

Determining whether or not a given matching M is stable turns out to be easy. We simply check each member of one of the genders to see if that person can belong to a blocking pair. Suppose that we check the men. Then for each man m we look to see if any woman w he prefers to $P_M(m)$ also prefers him to $P_M(w)$.

Example 3. Referring back to Example 1, the matching

$$\{(Cole, Heather), (Jack, Gail), (Ken, Maggie), (Larry, Jane)\}$$

is not stable because Gail prefers Ken over Jack and Ken's first choice was also Gail. Therefore (Ken, Gail) is a blocking pair for the matching.

Definition 4. A man m and woman w are said to form a *stable pair* if m and w are partners in some stable matching. If m and w form a stable pair, we also say that m and w are *stable partners* of each other.

Example 4. Referring back to Example 1, a stable matching would be

$$\{(Cole, Heather), (Jack, Jane), (Ken, Gail), (Larry, Maggie)\}$$

This is a stable matching, because there are no blocking pairs.

Definition 5. A stable matching is said to be *man-optimal* if every man is paired with his favorite stable partner. Similarly a *woman-optimal* matching occurs when every woman is paired with her favorite stable partner. We will always denote a man-optimal stable matching by M_0 and women-optimal by M_z.

We prove the existence of man-optimal and woman-optimal matchings in Chapter 4.

Example 5. Refer to the two tables below:

Joe	Derek	Zack	Dave
Karen	Judy	Penny	Beatrice
Judy	Karen	Beatrice	Penny
Penny	Penny	Karen	Judy
Beatrice	Beatrice	Judy	Karen

Table 2.3: Men's Preferences

Beatrice	Judy	Karen	Penny
Derek	Zack	Dave	Joe
Joe	Joe	Derek	Zack
Dave	Dave	Joe	Derek
Zack	Derek	Zack	Dave

Table 2.4: Women's Preferences

For the instance of the Stable Marriage Problem described by the above tables, man-optimal and woman-optimal matchings are given below:

$$M_0 = \{(Joe, Karen), (Derek, Judy), (Zack, Penny), (Dave, Beatrice)\}$$

$$M_z = \{(Joe, Penny), (Derek, Beatrice), (Zack, Judy), (Dave, Karen)\}$$

Notice, in M_0 every man is paired with his first choice, and likewise every woman is paired with her first choice in M_z. We cannot expect that this will always occur, as the example below shows.

Example 6. The two tables below show another instance of the Stable Marriage Problem.

Steve	Will	Anthony	Donald
Emma	Ava	Ava	Harper
Harper	Harper	Olivia	Ava
Ava	Emma	Emma	Olivia
Olivia	Olivia	Harper	Emma

Table 2.5: Men's Preferences

Olivia	Ava	Harper	Emma
Will	Steve	Anthony	Anthony
Donald	Donald	Will	Steve
Steve	Anthony	Donald	Will
Anthony	Will	Steve	Donald

Table 2.6: Woman's Preferences

Carrying out the Gale-Shapley algorithm (which we describe in the next chapter), we obtain the following matchings:

$$M_0 = \{(Steve, Emma), (Will, Harper), (Anthony, Olivia), (Donald, Ava)\}$$

$$M_z = \{(Steve, Ava), (Will, Harper), (Anthony, Emma), (Donald, Olivia)\}$$

We see that some individuals end up with their first choice, but not everyone does.

Example 7. In this example, we discuss the College Admissions Problem. A college is considering a set of n applicants of which it can admit a quota of only q. The admissions office evaluates the qualifications of the applicants and decides which ones to admit. Offering only the q best qualified applicants will most likely not work, because it cannot be assumed that all who are offered admission will accept. For a college to receive q acceptances, it will have to offer admission to more than q applicants. This creates some issues because there are many factors that are unknown. The colleges are unaware of whether the applicants have applied elsewhere, how they rank their colleges and which other colleges are offering them admission. Due to all this uncertainty, colleges can only expect that the entering class will be close to the desired quota.

The usual admissions procedure not only creates problems for the colleges, but for the students as well. For example, asking an applicant to put one's order of colleges in a preference list could affect what college one gets into. A college might see that it is the third choice on the preference list and this could hurt one's chances of even getting into the college. A solution that was offered is the "waiting list." Applicants are informed that they are not admitted into the college at the time, but may gain admission later. However, this solution itself still has some problems. Students must decide if they want to wait for the college to respond to see if they will get admitted later on or if they should just accept another college that has already offered acceptance.

An alternate approach to the above would be to proceed as follows. Each applicant would be required to make a preference list in which the colleges are ranked in order of preference, excluding colleges he or she would never accept. Similarly, the colleges would rank the applicants in order of preference, excluding those whom they would never accept. The goal would then be to find an assignment of applicants to colleges which is stable in the following sense: it is impossible to find an applicant x and a college y such that x is unassigned or prefers y to the college he or she is assigned to, and y has an opening or prefers x to one of the applicants who have been assigned to y. Thus the problem becomes a generalized version of the Stable Marriage Problem.

Chapter 3

Gale-Shapley Algorithm

Gale's and Shapley's results show that in every instance of the stable marriage problem there exists at least one stable matching. They did so by giving an algorithm that is guaranteed to find such a matching. They showed that their algorithm finds a stable matching that simultaneously gives all men (or women, if the gender roles are reversed) the best possible partner that they can have in any stable matching.

The algorithm can be expressed as a sequence of "proposals" from men to women. Through the execution, each person is either engaged or free; the man can be free or engaged. However, once a woman is engaged, she can no longer be free again. That does not mean that her fiancé's marital status cannot change. A man who is engaged more than once obtains fiancées who are less desirable to him, while each engagement brings the woman to a more favorable partner.

A free woman must accept the first proposal she receives and becomes engaged to whoever proposes to her. When an engaged woman is proposed to, she compares the proposer and her current fiancé, then rejects the one she finds less desirable. If she becomes engaged with the new proposer, her ex partner will now be free again. Each man proposes to the woman on his preference list in order of appearance until he becomes engaged and the algorithm terminates when everyone is engaged.

Each man proposes to the women on his preference list, in the order in which they appear. He does this until he becomes engaged. However, if a woman decides to break off the engagement, he becomes free again. When this happens, he goes back to his sequences of proposals and proposes to the next woman. The algorithm terminates when everyone is engaged. Upon termination, the engaged couples form a stable matching. This will be proved in Chapter 4. We denote the set of all men by M and the set of all women by W. Below we give the basic Gale-Shapley algorithm, in pseudocode form.

assign each person to be free.
while some m is free do
begin
  w := the first woman on m's list to whom m has not yet proposed ;
  if w is free then
    assign m and w to be engaged { to each other }
  else
    if w prefers m to her fiancé m' then
      assign m and w to be engaged and m' to be free
    else
      w rejects m and m remains free
end ; output the stable matching of the n engaged pairs

The above is taken from [2]. Since the order in which the free men propose is not specified, the algorithm has an element of nondeterminism. Later on, in Chapter 4, we will see that the nondeterminism is not a problem. The order in which the free men propose does not affect the outcome. We will prove that the above algorithm terminates and yields a stable matching in Chapter 4 (see Theorem 1).

1	2	3	4
3	3	1	2
2	4	4	1
1	1	2	4
4	2	3	3

Table 3.1: Men's Preferences

1	2	3	4
1	2	1	4
4	3	3	1
3	4	2	2
2	1	4	3

Table 3.2: Women's Preferences

Example 8. Consider the instance of the Stable Marriage Problem described by Tables 3.1 and 3.2. A possible execution of the algorithm results in the following sequence of proposals: man 1 to woman 3 (accepted); man 2 to woman 3 (not accepted); man 3 to woman 1 (accepted); man 4 to woman 2 (accepted). Based off this, woman 4 does not have a match which means there is something wrong. As one can see, women 3 was proposed to twice. We have to view woman 3's list to see who she prefers out of the two men. Her first choice is man 1, therefore, she stays with man 1. Then we view man 2's second choice which is woman 4. Woman 4 has not yet been proposed to so her match will be with man 2. Thus, the stable matching generated by the man-oriented version of the algorithm is $\{(1,3),(2,4),(3,1),(4,2)\}$.

All possible executions of the Gale-Shapley algorithm (men being the proposers) lead to the same stable matching and every man obtains from it the best partner that he can possibly have in any stable matching. This follows from Theorem 2 in Chapter 4. Although the men are competing for the women, they can all agree on a stable matching that is optimal for all of them.

Notice that, according to Theorem 2, if each man is independently given his best stable partner, then the result is a stable matching. This is surprising, because there is no prior reason as to why this should even be a matching. The stable matchings that we attain by the man-oriented version of the Gale-Shapley algorithm are man-optimal (recall Definition 5). If the roles are changed and we apply the woman-oriented version of the algorithm, then we obtain the woman-optimal stable matching. In some cases, the man and woman-optimal stable matchings will be identical, but in general, they will not be. As before, we will denote the man-optimal stable matching by M_0 and the woman-optimal by M_z.

When each man ends up with his best stable partner, it is no surprise that this comes with some sort of consequence. However, it is the opposite gender that faces the problem. In the man-optimal stable matching, each woman has the worst possible partner that she can have in any stable matching (see Theorem 3 in Chapter 4). Therefore, a man-optimal matching is also *woman-"pessimal"* (the term for each woman ending up with worst possible stable partner); likewise the woman-optimal stable matching is *man-pessimal.*

Example 9. Consider the situation described by Tables 3.3 and 3.4

1	2	3	4
1	3	2	4
2	4	4	1
3	1	1	2
4	2	3	3

Table 3.3: Men's Preferences

1	2	3	4
2	1	4	3
4	3	2	1
3	4	1	2
1	2	3	4

Table 3.4: Women's Preferences

If we apply the Gale-Shapley algorithm with men and then women as proposers, we obtain the man-optimal and woman-optimal stable matchings below:

$$M_0 = \{(1,1),(2,3),(3,2),(4,4)\}$$

and

$$M_z = \{(1,3),(2,1),(3,4),(4,2)\}.$$

Definition 6. If the Gale-Shapley algorithm is being carried out with the men doing the proposing, then we call this the *man-oriented* version of the algorithm. Likewise, when women are doing the proposing, we call this the *woman-oriented* version.

We will see in Chapter 4 that the man-oriented version of the algorithm leads to a man-optimal stable matching, and the woman-oriented version leads to a woman-optimal stable matching.

Chapter 4

Theorems and Proofs

In this chapter, I will provide proofs of the theorems that were discussed in earlier chapters. When the man-oriented version of the Gale-Shapley algorithm is executed, each man proposes to the women on his preference list, starting with his first choice. He does this until he is engaged. If a woman breaks the engagement, then he is free again. He begins the sequence of proposals again starting with the next woman on his preference list. The algorithm will terminate when everyone is engaged. It turns out that the order in which the free men propose does not determine the outcome. We will now show that upon termination, the engaged couples form a stable matching.

Theorem 1. *In any given instance of the stable marriage problem, the Gale-Shapley Algorithm terminates and once it terminates, the engaged pairs form a stable matching.*

Proof. We use contradiction to prove that no man can be rejected by all women. Assume that a man can in fact be rejected by all women. Once a woman is engaged, she can no longer be free again (although she can become engaged to a different man). Therefore the only way a woman can reject a proposal is if she's already engaged. It follows that if a man is rejected by the last woman on his list, then all of the women have already been engaged. However, there are an equal number of men and women, and no man can have two partners. Thus, all of the men would have also been engaged, a contradiction. No man can propose to the same woman twice. Hence, the total number of proposals cannot exceed n^2 (in a case involving n men and n women). We conclude that the algorithm does terminate.

After termination, the engaged pairs form a matching, which we denote by M. If man m prefers woman w to $P_M(m)$, then w must have rejected m at some point. This means that w was, either at the time of m's proposal or at some later time, engaged to a man she prefers to m. Any further changes in w's engagement status will result in an even better partner. Thus, w cannot prefer m to $P_M(w)$ and (m, w) cannot block M. This means that the matching M is stable. □

Recall, the concept of stable partner was introduced in Definition 4.

As we have mentioned before, all executions of the Gale-Shapley algorithm lead to the same stable matching. This stable matching has the property that every man ends up with his best possible stable partner, as we prove below.

Theorem 2. *After any execution of the man-oriented version of the Gale-Shapley algorithm, every man is paired with his favorite stable partner.*

Proof. Suppose towards a contradiction that some execution E of the Gale-Shapley algorithm results in a matching M with some man m not paired with his favorite stable partner. Then, in some other stable matching M', man m is paired with a woman he likes more than his partner in M. Let's denote m's partners in M and M' by w and w', respectively. Recall, in any execution of the Gale Shapley algorithm, each man proposes first to the woman he likes best. If he is rejected, then he just proposes to the women he likes second best, and so on. Since m winds up with w in M, and not with w', it follows that w' rejected m at some point during E. We may assume that this was the first time during E that a man was rejected by a stable partner. Now, w' rejected m for someone else; let's call him m'. Since w' rejected m for m', it must be the case that w' prefers m' over m.

We claim that w' is the favorite stable partner of m'. To see this, we assume for a contradiction that the favorite stable partner of m' is a different woman, let's call her w''. Observe that m' must have proposed to w'' during E before he proposed to w' (since he prefers w'' to w). Furthermore, he must have been rejected by w'' at some point (if not, he would have never proposed to w'). Thus during E, m' is rejected by w'' before he proposes to w'. However, recall that m is rejected by w' because of the engagement of w' with m'. This means that m is rejected by w' after m' is rejected by w''. This contradicts our assumption that the rejection of m by w' is the first time during E that a man is rejected by a stable partner. The claim is established.

So now we have that w' prefers m' over m and m' prefers w' over all of his other stable partners. This means that the pair (m', w') blocks M'. But M' was suppose to be stable. This contradiction completes the proof of the theorem. □

The above theorem implies that if each man is independently given his best stable partner, then the result is a stable matching.

Having examined the output of the man-oriented version of the Gale-Shapley algorithm from the perspective of the men, we now consider just how good the output is for the women. Before reading the theorem and proof below, the reader may wish to review Definition 5 in Chapter 2.

Theorem 3. *In the man-optimal stable matching, each woman has the worst possible partner that she can have in any stable matching.*

Proof. As always, we denote the man-optimal stable matching by M_0. We are going to prove this theorem by contradiction. Assume that there is a woman w whose partner in M_0 is not her least favorite stable partner. This means that there exists a stable matching M' such that w prefers her partner in M_0 over her partner in M'. We represent w's partner in M_0 and M' as m and m', respectively. Based off of this, we have that m can't have the same partner in M' that he has in M_0; let's denote his partner in M' by w'. Since m is paired with w in the man-optimal matching, then m must prefer w to w'. It now follows that M' is blocked by (m, w). This contradicts the fact that M' is stable. □

The man-optimal (and woman-optimal) stable matchings turn out to exhibit another interesting kind of optimality, which we describe in the theorem below.

Theorem 4. *For any instance of the stable marriage problem, there is no matching, stable or otherwise, in which every man has a partner whom he strictly prefers to his partner in the man-optimal stable matching M_0.*

Proof. Clearly, there cannot be any stable matching with the property that was just stated. We will prove this by using contradiction. Therefore, we assume that there exists an unstable matching M' with this property. During an execution of the man-oriented Gale-Shapley algorithm, if w is the last woman to become engaged then no man was rejected by w, since the algorithm terminates when the last woman receives her first proposal. But if w's partners in M_0 and M' are m and m' respectively, then m' prefers w to his partner in M_0, so w must have rejected m'. Thus, a contraction. □

Chapter 5

Examples

In this chapter, we will carry out the Gale-Shapley algorithm using different versions of the algorithm (man-oriented vs woman-oriented). In some of the examples, we give a detailed description of how the algorithm proceeds when there are rejections.

Carl	Ethan	Jon	Ryan
Grace	Madison	Addison	Riley
Madison	Addison	Riley	Grace
Riley	Grace	Madison	Addison
Addison	Riley	Grace	Madison

Table 5.1: Men's Preferences

Addison	Grace	Madison	Riley
Ethan	Ethan	Jon	Ryan
Carl	Jon	Carl	Ethan
Ryan	Carl	Ethan	Jon
Jon	Ryan	Ryan	Carl

Table 5.2: Women's Preferences

We read the table as follows: Carl's first choice is Grace, his second choice is Madison, his third is Riley and his fourth is Addison. The preferences of the others are read in the same way.

A possible execution of the Gale-Shapley algorithm results in the following sequences of proposals: Carl to Grace (accepted); Ethan to Madison (accepted); Jon to Addison (accepted); Ryan to Riley (accepted). Thus, the stable matching generated by the man-oriented version of the algorithm is

$$\{(Carl, Grace), (Ethan, Madison), (Jon, Addison), (Ryan, Riley\}.$$

Note that in this example, every man ends up with his first choice. This of course cannot happen in situations where the same woman is listed first by two or more men.

Bill	Clint	Kevin	Ron
Sally	Tammy	Sally	Anna
Anna	Anna	Jane	Sally
Jane	Sally	Tammy	Tammy
Tammy	Jane	Anna	Jane

Table 5.3: Men's Preferences

Anna	Jane	Sally	Tammy
Clint	Kevin	Bill	Ron
Bill	Ron	Ron	Bill
Ron	Bill	Kevin	Clint
Kevin	Clint	Clint	Kevin

Table 5.4: Women's Preferences

For the situation described by Tables 5.3 and 5.4, a possible execution of the Gale-Shapley algorithm results in the following sequences of proposals: Bill to Sally (accepted); Clint to Tammy (accepted); Kevin to Sally (not accepted); Kevin to Jane (accepted); Ron to Anna (accepted). Kevin and Sally were not paired together because Sally's first choice was Bill. Thus, the stable matching generated by the man-oriented version of the algorithm is

$$\{(Bill, Sally), (Clint, Tammy), (Kevin, Jane), (Ron, Anna)\}.$$

Note that this matching is man-optimal, by Theorem 2.

Dave	Frank	Greg	Hank	Luke
Maria	Erica	Kasey	Nancy	Nancy
Erica	Irene	Erica	Maria	Irene
Kasey	Kasey	Nancy	Irene	Maria
Irene	Maria	Maria	Erica	Kasey
Nancy	Nancy	Irene	Kasey	Erica

Table 5.5: Men's Preferences

Erica	Irene	Kasey	Maria	Nancy
Hank	Dave	Frank	Hank	Frank
Frank	Luke	Dave	Dave	Greg
Luke	Greg	Greg	Luke	Dave
Dave	Frank	Luke	Greg	Luke
Greg	Hank	Hank	Frank	Hank

Table 5.6: Women's Preferences

For the situation described by Tables 5.5 and 5.6, a possible execution of the woman-oriented version of the Gale-Shapley algorithm results in the following sequence: Erica to Hank (accepted); Irene to Dave (accepted); Kasey to Frank (accepted); Maria to Hank (accepted); Erica to Frank(accepted); Kasey to Dave(accepted); Irene to Luke (accepted); Nancy to Frank (not accepted); Nancy to Greg (accepted).When Maria proposes to Hank, we take a look at Hank's preference list and see that he prefers Maria over Erica so he breaks off his engagement with Erica and goes with Maria. Now we go back to Erica who must propose to her second choice. She proposes to Frank who is already engaged to Kasey. However, Frank prefers Erica to Kasey so he leaves Kasey to be with Erica. Then Kasey is single again and proposes to her second choice, Dave. Dave must accept her proposal, because he has not yet been proposed to. Next, Nancy proposes to Frank, but Frank prefers Erica over Nancy so he stays with Erica. Then Nancy proposes to her second choice Greg, who must accept because he is unengaged. Thus, the stable matching generated by the woman-oriented version of the algorithm is

$$\{(Dave, Kasey), (Frank, Erica), (Greg, Nancy), (Hank, Maria), (Luke, Irene)\}.$$

This matching is woman-optimal.

Brian	Dan	Larry	Rob
Fran	Fran	Taylor	Leslie
Leslie	Taylor	June	June
June	June	Fran	Fran
Taylor	Leslie	Leslie	Taylor

Table 5.7: Men's Preferences

Fran	June	Leslie	Taylor
Brian	Brian	Dan	Rob
Rob	Larry	Brian	Larry
Larry	Rob	Larry	Dan
Dan	Dan	Rob	Brian

Table 5.8: Women's Preferences

Consider now the situation described by Tables 5.7 and 5.8. A possible execution of the man-oriented version of the Gale-Shapley algorithm results in the following sequence of proposals: Brian to Fran (accepted); Dan to Fran (not accepted); Dan to Taylor (accepted); Larry to Taylor (accepted); Dan to June (accepted); Rob to Leslie (accepted). Dan proposes to Fran, but she rejects her proposal, because her first choice is Brian. Then Dan proposes to Taylor who has not been proposed to yet so she accepts his proposal. However, right after that, Larry proposes to Taylor and Taylor prefers Larry over Dan. Therefore she leaves her engagement with Dan to be engaged to Larry. Dan is once again single and must propose to his next choice. His third choice is June. June is single so she accepts Dan's proposal. No other rejections occur throughout the rest of the algorithm. Thus, the stable matching generated by the man-oriented version of the algorithm is

$$\{(Brian, Fran), (Dan, June), (Larry, Taylor), (Rob, Leslie)\}.$$

A possible execution of the Gale-Shapley algorithm using the woman-oriented version results in the following sequence of proposals: Fran to Brian (accepted); June to Brian (not accepted); June to Larry (accepted); Leslie to Dan (accepted); Taylor to Rob (accepted). When June proposes to Brian, he turns down her proposal, because he prefers Fran to June. Therefore, June must ask her second choice, which is Larry. Larry has not been proposed to yet so he must accept her proposal. No other rejections occur throughout the rest of the algorithm. Thus, the stable matching generated by the woman-oriented version of the algorithm is

$$\{(Brian, Fran), (Dan, Leslie), (Larry, June), (Rob, Taylor)\}.$$

After carrying out the two different versions of the algorithm, we see that the results vary. The only couple that remains together is Brian and Fran. This occurs because they are both each other's first choice.

Doug	James	Liam	Noah
Mia	Sophia	Zoe	Zoe
Sophia	Zoe	Lily	Lily
Lily	Lily	Sophia	Sophia
Zoe	Mia	Mia	Mia

Table 5.9: Men's Preferences

Lily	Mia	Sophia	Zoe
Noah	Doug	Liam	Doug
Liam	James	James	James
James	Liam	Doug	Liam
Doug	Noah	Noah	Noah

Table 5.10: Women's Preferences

For the situation described by the above two tables, we will show two possible executions of the Gale-Shapley algorithm. One will be a man-oriented version and the other will be a woman-oriented version. Then we will compare the outcomes.

A possible execution of the algorithm using the man-oriented version results in the following sequence of proposals: Doug to Mia (accept); James to Sophia (accept); Liam to Zoe (accept); Noah to Zoe (not accept); Noah to Lily (accept). Zoe prefers Liam to Noah therefore she will stay with Liam. Thus, the stable matching generated by the man-oriented version of the algorithm is

$$\{(Doug, Mia), (James, Sophia), (Liam, Zoe), (Noah, Lily)\}.$$

A possible execution of the Gale-Shapley algorithm using the woman-oriented version results in the following sequence of proposals: Lily to Noah (accept); Mia to Doug (accept); Sophia to Liam (accept); Zoe to Doug (not accept), Zoe to James (accept). Doug prefers Mia over Zoe so Zoe must ask her second choice. She proposes to James, who is single, and is now engaged to James. Thus, the stable matching generated by the woman-oriented version of the algorithm is

$$\{(Doug, Mia), (James, Zoe), (Liam, Sophia), (Noah, Lily)\}.$$

While two couples did not change from the previous matching, James and Liam switched partners with Zoe and Sophia.

Bibliography

[1] Gale, D. and Shapley, L.S. *College Admissions and the stability of marriage*, American Mathematical Monthly, **69** (1962), 9-15.

[2] Gusfield, Dan and Irving, Robert W. *The Stable Marriage Problem Structure and Algorithms*, The MIT Press, Cambridge, Massachusetts, 1989.

[3] Manlove, David F. *Algorithmics of Matching Under Preferences*, World Scientific Publishing Company, Singapore, 2013.

[4] Roth, Alvin E. *The Evolution Of The Labor Market For Medical Interns And Residents: A Case Study In Game Theory*, Journal Of Political Economy 92.6 (1984): 991. Business Source Complete. Web. 15 Mar. 2016.