Finding a Fair Stable Marriage

Cara A. Leavens
Oregon State University

REU Program
1991

Finding a Fair Stable Marriage

I. Abstract

This paper will review the results of Gale and Shapley using the book The Stable Marriage Problem, by Gusfield and Irving as a source. It will then explore the possibilities for more "fair" stable matchings than the male and female optimal stable marriages. The search leads to graph theory, where matching algorithms already exist for network flows. The concept of stable marriages is also applied to matching hospitals and interns which is less romantic, but much more practical. It is this application of stable marriages that is the motivation for this paper.

II. Introduction

A **matching** is a one-to-one correspondence between two groups, where each member has a preference list, a ranking of the members of the opposite group. In the stable marriage problem, this idea is applied to men and women, where the town matchmaker wants to pair them up. The matching must be **stable**, meaning two people from different couples may not prefer each other over their own partners. This couple, if one exists, is called a **blocking pair**. If no blocking pairs exist, then a matching is stable. Only cases will be considered where the number of men equals the number of women, denoted by n. Here is an example of preference lists where n = 5. Man 1 prefers woman 3 first, woman 5 second, and so on. Similarly, woman 1 prefers man 1 first, man 2 second, and so on.

Men:	Women:
1 - 3 5 4 1 2	1 - 1 2 3 4 5
2 - 4 5 3 2 1	2 - 5 3 2 1 4
3 - 1 2 5 4 3	3 - 4 1 2 3 5
4 - 2 5 3 4 1	4 - 4 5 3 1 2
5 - 5 1 2 3 4	5 - 2 1 3 4 5

An algorithm for finding a stable marriage was developed by Gale and Shapley in 1962. The **Gale-Shapley algorithm** starts with one group, for convenience we will start with the men. Then, while there exists an unengaged man, he proposes to the first woman on his list who has not rejected him. If the woman is unengaged, then they are matched. Otherwise, she chooses the higher man on her list and the lower man becomes unengaged. This algorithm terminates when all the men and women are matched, there exists at least one stable matching for any preference lists. The resulting matching does not depend on the order in which the men propose. If the women propose, however, then a different matching results.

When the men propose, the matching is called **male optimal**, because each man has the best partner he can have in any stable marriage. But in the male optimal matching, the women have the worst partners they can have in any stable matching. Similarly, when the women do the proposing, the outcome is the **female optimal** marriage because the women have the best partners they can have in any stable marriage, and the men have the worst partners. There may also be more stable marriages in between the male and female optimal matchings.

There is a modified version of the Gale-Shapley algorithm which will be illustrated by an example using the previous preference lists. When a man (or woman) is proposed to, all of the women (or men) who are preferred less than the proposer are deleted from the preference lists, and all the men (women) are deleted from the would be proposers' lists. In this example, let the men propose to the women. Man 1 proposes to woman 3, so woman 3 crosses off men 2, 3, and 5 because she would never accept a proposal from anyone below man 1. Then men 2, 3, and 5 cross 3 off their lists since they do not need to bother proposing. After the first proposal, the lists look as follows:

Men:							Women:						
1	-	3	5	4	1	2	1	-	1	2	3	4	5
2	-	4	5		2	1	2	-	5	3	2	1	4
3	-	1	2	5	4		3	-	4	1			
4	-	2	5	3	4	1	4	-	4	5	3	1	2
5	-	5	1	2		4	5	-	2	1	3	4	5

Now let the rest of the men propose to get the final lists:

Men:							Women:						
1	-	3	5	4	1	2	1	-	1	2	3		
2	-	4	5		2	1	2	-	5	3	2	1	4
3	-	1	2	5	4		3	-	4	1			
4	-	2	5	3	4		4	-	4	5	3	1	2
5	-	5		2		4	5	-	2	1	3	4	5

To get the G-S list, which contains all of the stable marriages, the process is repeated with the women proposing:

Men:							Women:						
1	-	3	5	4	1		1	-	1		3		
2	-	4	5				2	-	5	3			4
3	-	1	2	5	4		3	-	4	1			
4	-	2	5	3			4	-			3	1	2
5	-	5		2			5	-	2	1	3	4	5

This is the final G-S list (either one may be used). It is important to realize that all stable marriages are contained in this list, but the converse is not true. There may be one or more matchings in the G-S list that are not stable. From the G-S list we can list all the possible stable matchings. For convenience, we will represent a matching by an ordered n-tuple where the position of the number stands for the man and the number represents the woman. For example, (1,5,4,3,2) denotes the matching 1-1, 2-5, 3-4, 4-3, 5-2, where the first numbers correspond to the men and the second numbers correspond to the women. In the example, there are seven marriages contained in the G-S list: (1,4,2,3,5), (1,4,5,3,2), (1,5,4,3,2), (3,4,1,2,5), (4,5,1,3,2), (5,4,1,3,2), and (3,4,1,5,2). As it turns out, two of the marriages are not stable. Stability can be easily checked using the preference lists. For example, look at the matching (1,4,2,3,5). It is only necessary to check the women that each man prefers over his partner, to see if someone he prefers also prefers him over her partner. This couple would then be a blocking pair. Below, the couples in the matching are marked by bold numbers:

Men:	Women:
1 - 3 5 4 **1** 2	1 - **1** 2 3 4 5
2 - **4** 5 3 2 1	2 - 5 **3** 2 1 4
3 - 1 **2** 5 4 3	3 - **4** 1 2 3 5
4 - 2 5 **3** 4 1	4 - 4 5 3 1 **2**
5 - **5** 1 2 3 4	5 - 2 1 3 4 **5**

Since man 1 prefers women 3, 5, and 4 over his partner, whether or not any of them prefer man 1 over their partner needs to be checked. Since both women 4 and 5 prefer man 1 over their partners, 1-4 and 1-5 are blocking pairs for this matching, and the matching is not stable. Couple 4-5 is also a blocking pair for this matching. Similarly, matching (1,4,5,3,2) is blocked by pairs 1-4 and 1-5. Now it is necessary to know which one of the remaining five stable marriages is the most fair.

III. Weight Functions

In the search for a fair stable marriage, the concept of a weight function must be introduced. This function defines what is meant by "fair". For a couple, let m be the rank of the woman on the man's list and w be the rank of the man on the woman's list. There are many functions that could be used, but we want one that has properties that would make the least weighted matching be the most fair matching. Some weight functions might add the sums of the m's and w's, or take their difference. Continuing the previous example, here are some different weight functions applied to the five different stable matchings:

Matchings		Weights			
Men		m	w	$\lvert m - w \rvert$	m + w
Women	(1,5,4,3,2)	16	7	9	23
	(4,5,1,3,2)	12	10	2	22
	(5,4,1,3,2)	10	12	2	22
	(3,4,1,5,2)	8	15	7	23
	(3,4,1,2,5)	5	20	15	25

The first matching listed is the female optimal matching, because the total of the women's weights is the smallest of all the matchings. The least possible weight is 5, if everyone has their first choice, which occurs in the last matching, the male optimal matching.

A small sum of weights is desirable, because that means everyone has a high choice on their list, but this is not enough. The difference between the weights also needs to be represented. Consider the function $m^2 + w^2 + (m - w)^2$. Dividing by two, this function reduces to $m^2 + w^2 - mw$. The table below shows the behavior of the function for combinations of m and w for n up to 10.

$m^2 + w^2 - mw$

	1	2	3	4	5	6	7	8	9	10
1	1	3	7	13	21	31	43	57	73	91
2		4	7	12	19	28	39	52	67	94
3			9	13	19	27	37	49	63	79
4				16	21	28	37	48	61	76
5					25	31	39	49	61	75
6						36	43	52	63	76
7							49	57	67	79
8								64	73	84
9									81	91
10										100

The table is read as follows: for any couple, if the man has his third choice and the woman has her first choice, the weight for that couple is in the first row and third column (or the third row and first column), 7 in this case 8. The table is symmetrical, so only half of it is shown. When the weights for each couple are added, that gives the total weight for the matching. Although this may be arguably a fair weight function, It does not have the property that the columns are non-increasing. For example, the entries in row 1 column 6 and row 5 column 6 are equal, meaning a matching of ranks 1 and 6 is as fair as a matching of ranks 5 and 6. A new weight function is needed with the property of the columns being non-increasing so this problem does not occur. Consider $m + w + (m - w)^2$, which still takes into account both the sum and the difference of the individual weights, but places more emphasis on the difference.

$m + w + (m - w)^2$

	1	2	3	4	5	6	7	8	9	10
1	2	4	8	14	22	32	44	58	74	92
2		4	6	10	16	24	34	46	60	76
3			6	8	12	18	26	36	48	62
4				8	10	14	20	28	38	50
5					10	12	16	22	30	40
6						12	14	18	24	32
7							14	16	20	26
8								16	18	22
9									18	20
10										20

This function is more desirable because it has the property that as the rankings get further apart, the weight gets larger, and also as both the rankings get larger, the weight gets larger. Let's go back to the previous weight table and see how this function looks.

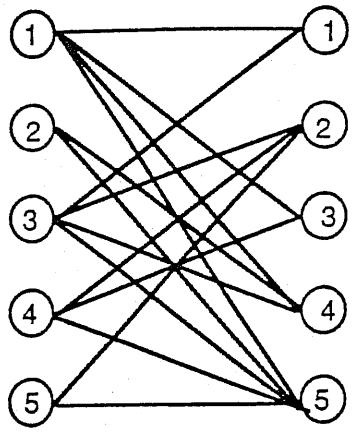
Matchings		Weights				
Men		m	w	$\|m - w\|$	$m + w$	$m+w + (m-w)^2$
Women	(1,5,4,3,2)	16	7	9	23	42
	(4,5,1,3,2)	12	10	2	22	36
	(5,4,1,3,2)	10	12	2	22	50
	(3,4,1,5,2)	8	15	7	23	52
	(3,4,1,2,5)	5	20	15	25	78

The matching with the smallest weight is one of the matchings with the smallest sum, and smallest difference, so it appears to be the most fair. Notice that the new weight function also distinguishes between the two matchings with the same sum and difference.

Given any preference lists, to find the most fair matching, one could compute the weight of every stable marriage and then choose the lowest weight, but that is not at all efficient as n gets large. In the next section some introductory material will be presented for an algorithm that is more efficient.

IV. Networks

A matching may be represented by a **bipartite graph**, since the men and women are two disjoint groups. The vertices on the left represent the men, and the vertices on the right represent the women. There is an edge between a man and a woman if they are a couple in any stable matching. In other words, if the couple is in the G-S list, their vertices are connected. The graph for the above example is shown below:



The graph may be expanded, so it has the form of a network, then existing network flow algorithms may be used. A **network** is a graph with directed, weighted edges and a unique source and sink. Each edge from a man to a woman has a capacity and a weight. An edge is weighted according to the weight function, using the preference lists. The capacity for edges between two vertices is infinite. A source (S) and sink (T) are added to make the graph in the form of a network. The capacity for the edges between the source and the vertices and between the sink and the vertices is one, and the cost is zero. Now network flow algorithms may be used to find the minimum weight matching.

Consider another example with the following preference lists:

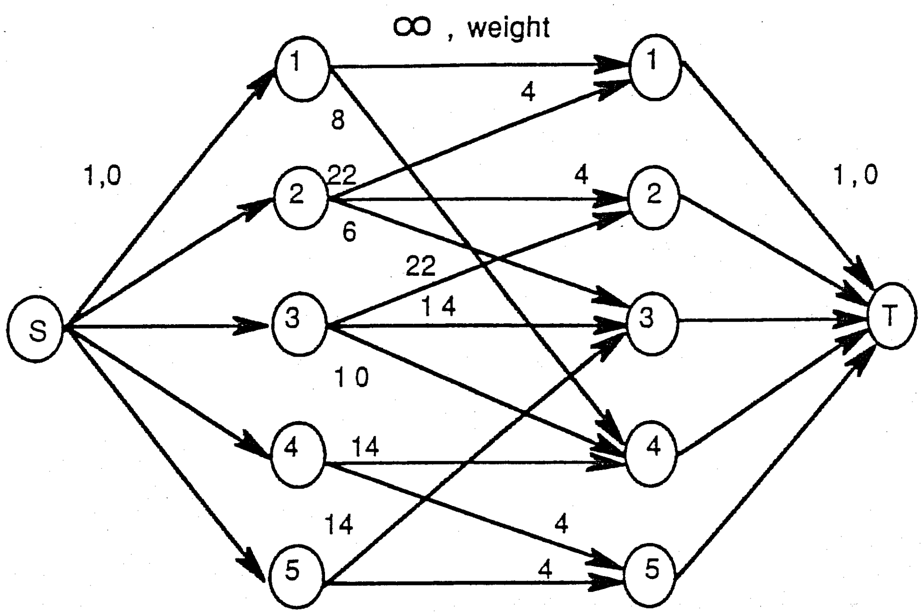
Men:
1 - 1 3 4 2 5
2 - 2 4 3 5 1
3 - 3 1 5 4 2
4 - 4 5 2 1 3
5 - 5 2 1 3 4

Women:
1 - 2 1 3 5 4
2 - 3 2 4 1 5
3 - 5 4 2 3 1
4 - 1 3 5 4 2
5 - 4 5 1 2 3

The reader may verify that the G-S list is as follows (only the men's list is shown because both lists contain the same information):

Men:
1 - 1 4
2 - 2 3 1
3 - 3 4 2
4 - 4 5
5 - 5 3

From the G-S list, we can construct the network below, by adding a source, sink, capacities and weights to the bipartite graph. The first number on the edges is the capacity, followed by the weight. For readability, only the weights are shown on the edges of the network.



The G-S list contains four matchings in this case, all of which are stable. The weights for each matching are listed below.

Matchings				Weights
Men		m	w	$m+w + (m-w)^2$
Women	(1,2,3,4,5)	5	14	40
	(1,2,4,5,3)	12	8	36
	(1,3,2,4,5)	11	12	50
	(4,1,2,5,3)	19	5	70

V. Algorithm and Efficiency

An important thing to note about this algorithm is that it is independent of the weight function chosen. If one has a different opinion of what is fair, a different function may be chosen.

1. Using the Gale-Shapley modified algorithm, generate the G-S list from the preference lists.

2. Construct a network flow using only edges in the G-S list, adding weights according to the preference lists.

3. Use a network flow minimum cost-maximum flow algorithm to find the matching with the least weight.

Because the Gale-Shapley algorithm can be executed in $O(n^2)$ time, and the most efficient network flow algorithms can be executed in $O(n^3)$ time, this algorithm is still fairly efficient. The problem with the algorithm is that the least weight matching may not be stable. Whether or not this can happen with the weight function chosen here has not been proved or disproved. If it can happen, it is not known with what frequency it will happen, so these problems need further attention.

VI. References

Gusfield, Dan and Robert W. Irving, The Stable Marriage Problem, MIT Press, 1989.

Lawler, Eugene L., Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.