Stable Marriage Problem

Jennie Premer
Fort Calhoun, NE

Math in the Middle Institute Partnership
Research and Expository Report

in partial fulfillment of the MAT Degree
Department of Mathematics
University of Nebraska-Lincoln
Summer 2011

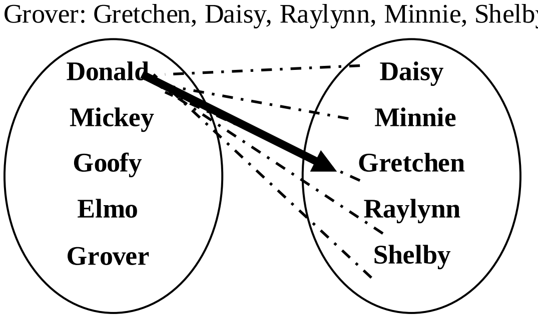
Introduction

In the stable marriage problem[1], we wish to pair off, or marry, *n* men and *n* women, to make a set of *n* matching pairs such that everyone is reasonably happy. When setting up our pairs, we need to create a situation where no man and no woman prefer each other to their current partner. If this is true, then we have created *n* stable marriages. If this is not true, then we have a blocking pair, namely two people in relationships who prefer each other to their current spouses. To start the pairing process, each man and woman rank the members of the opposite sex in order of their preferences. This means that each woman will give a complete list of men between 1 and *n* ordered by her preferences: first choice, second choice, etc. Each man will do the same, ordering the women. Mathematically we would say that there are two disjoint sets A and B. For now, we assume the set A has the same number of elements as set B. We will find a set Z of *n* pairs (a,b) such that a is an element of A and b is an element of B.

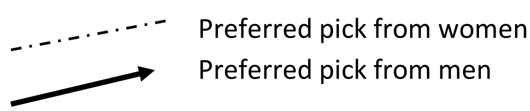
Assume we own an Internet matching company and have been given 5 profiles about men and 5 profiles about women. Our male and female customers have ranked each other 1-5 in order of preference. We need to find a stable matching of the 5 couples.

Men :	Picks
Donald:	Gretchen, Daisy, Minnie, Raylynn, Shelby
Mickey:	Minnie, Shelby, Daisy, Gretchen, Raylynn
Goofy:	Shelby, Daisy, Gretchen, Raylynn, Minnie
Elmo:	Gretchen, Daisy, Raylynn, Minnie, Shelby
Grover:	Gretchen, Daisy, Raylynn, Minnie, Shelby

Women :	Picks
Minnie:	Donald, Mickey, Elmo, Grover, Goofy
Daisy:	Donald, Grover, Elmo, Goofy, Mickey
Gretchen:	Donald, Grover, Goofy, Mickey, Elmo
Raylynn:	Donald, Mickey, Grover, Elmo, Goofy
Shelby:	Donald, Mickey, Elmo, Goofy, Grover



*Notice that all the women picked Donald as their first choice.



[1] *Disclaimer: In this paper we will consider a marriage to be between a man and a woman. This is done for mathematical purposes only.

In this example, each woman chooses Donald as their first pick. This poses a problem. There is only one Donald, thus only one woman will be able to have Donald as her husband. All the other women will be left with a different choice. So we are not able to guarantee everyone's first choice. We then look at trying to find stable marriage engagements.

We begin with all the men proposing to their first choice. Below is a list of the proposals in the first round:

Donald – Gretchen
Mickey – Minnie
Goofy – Shelby
Elmo – Gretchen
Grover – Gretchen

We know that only one man can marry Gretchen, and of course, Gretchen picks Donald. Elmo and Grover will have to pick again as they both proposed to Gretchen, but Donald was Gretchen's first choice.

Once Donald and Gretchen are identified as preferring each other they are paired and we can return to the original rankings with Donald and Gretchen removed from the rankings to conduct a second round of proposals. In the chart below, note that now Mickey and Minnie prefer each other and Grover and Daisy prefer each other.

Men :	Picks
Mickey:	**Minnie**, Shelby, Daisy, Raylynn
Goofy:	Shelby, Daisy, Raylynn, Minnie
Elmo:	Daisy, Raylynn, Minnie, Shelby
Grover:	**Daisy**, Raylynn, Minnie, Shelby

Women :	Picks
Minnie:	**Mickey**, Elmo, Grover, Goofy
Daisy:	**Grover**, Elmo, Goofy, Mickey
Raylynn:	Mickey, Grover, Elmo, Goofy
Shelby:	Mickey, Elmo, Goofy, Grover

At this stage, Minnie should accept Mickey's proposal and Daisy should accept Grover's proposal. We are left with the following men and women and their preferences.

Men :	Picks
Goofy:	Shelby, Raylynn
Elmo:	Raylynn, Shelby

Women :	Picks
Raylynn:	Elmo, Goofy
Shelby:	Elmo, Goofy

At this point, Goofy proposes to Shelby and Elmo proposes to Raylynn. Raylynn should accept Elmo's proposal and Shelby is left to accept Goofy's proposal. Our stable marriage plan would marry Donald & Gretchen, Mickey & Minnie, Grover & Daisy, Elmo & Raylynn, and Goofy & Shelby.

Gale-Shapely Algorithm

The Gale-Shapely algorithm can streamline the process. David Gale and Lloyd Shapely proved in 1962 that with an equal number of men and women it is possible to solve this problem using an algorithm that will result in all marriages being stable. Let us start by looking at the following sample preference lists:

Woman Preferences

	Joe	Brian	George	Matt	Jim
Amy	1	2	4	3	5
Sarah	3	5	1	2	4
Susan	5	4	2	1	3
Kelly	1	3	5	4	2
Dianne	4	2	3	5	1

Men Preferences

	Amy	Sarah	Susan	Kelly	Dianne
Joe	5	1	2	4	3
Brian	4	1	3	2	5
George	5	3	2	4	1
Matt	1	5	4	3	2
Jim	4	3	2	1	5

We will assume that men always propose to women; however, we could have assumed that women proposed to men. In addition we will want to think of these proposals in rounds. In the beginning, no one is engaged and the end of the process is determined when everyone is engaged. Also a woman once engaged will always be engaged but she will have the opportunity to "trade up" if she receives a proposal from a man she prefers, and every man must propose to every woman until one accepts. This assures that there will never be an end where there is a man and woman not engaged.

The men will each propose to the first woman on their preference list at the same time. The women will follow the following guidelines:

- If the woman is not engaged and has received no proposals, then the woman has to wait.

- If the woman is not engaged and has received one proposal, then the woman will accept the proposal.

- If the woman is not engaged and has received more than one proposal, then the woman will accept the proposal from the man who is highest on her preference list.

- If the woman is already engaged and she receives another proposal, then the woman will accept the proposal of the man whom is highest on her preference list. In essence, she can "trade up."

Round 1:

Men: Each man will propose to the woman he loves most based on his preferences.

Women: The women will then respond based on the above listed conditions in every round.

Let us see how this works in our example:

Men:	Joe proposes to Sarah	Women:	Sarah accepts Joe's proposal
	Brian proposes to Sarah		Sarah does not take Brian's proposal
	George proposes to Dianne		Dianne accepts George's proposal
	Matt proposes to Amy		Amy accepts Matt's proposal
	Jim proposes to Kelly		Kelly accepts Jim's proposal

We now have four engaged men and four engaged women. Brian and Susan are not paired with anyone. Since not everyone is engaged, we must perform another round of proposals.

Round 2:

Men: Any man who is not engaged will propose to his next choice.

Women: Follow the same conditions as before.

In our example:

Men:	Brian proposes to Kelly	Women:	Kelly doesn't change (Jim is higher)

Note: Kelly was always engaged, we still have 4 men and 4 women who are engaged. Brian and Susan still do not have a matching. Since not everyone is engaged, we must do another round.

Round 3: Same as round 2

Men: Brian proposes to Susan   Women: Susan accepts Brian's proposal

Now we have five stable couples.

Joe	Sarah
Brian	Susan
George	Dianne
Matt	Amy
Jim	Kelly

We would have obtained a very different result by starting with the women proposing. In two rounds, we would have established the five stable couples listed below:

Amy	Brian
Sarah	George
Susan	Matt
Kelly	Joe
Dianne	Jim

This example shows us that there is not a unique stable marriage.

We now want to use the algorithm to pair up two sets, not necessarily males and females, so that there are no two elements that prefer each other to their current pairings. In order to do this, we need to abstract the mathematics behind the stable marriage problem. We replace the names in our previous tables with representatives for the elements of two arbitrary sets. Let A denote one set and B denote the other. The elements of the sets will be $\{a_1, a_2, \ldots, a_5\}$ and $\{b_1, b_2, \ldots, b_5\}$ respectively. We denote by S the set of stable matchings. The elements of S are pairs of elements from A and B. Below are the tables from the previous example written in terms of the arbitrary sets A and B.

Set B Preferences

	a_1	a_2	a_3	a_4	a_5
b_1	1	2	4	3	5
b_2	3	5	1	2	4
b_3	5	4	2	1	3
b_4	1	3	5	4	2
b_5	4	2	3	5	1

Set A Preferences

	b_1	b_2	b_3	b_4	b_5
a_1	5	1	2	4	3
a_2	4	1	3	2	5
a_3	5	3	2	4	1
a_4	1	5	4	3	2
a_5	4	3	2	1	5

Our solution using the Gale-Shapely algorithm, with proposers from Set A listed first, can be represented as S= $\{(a_1, b_2), (a_2,b_3), (a_3,b_5), (a_4,b_1), (a_5,b_4)\}$. If instead the Set B members proposed using the Gale-Shapely algorithm, proposers still listed first, our results would have been S= $\{(b_1,a_1), (b_2,a_3), (b_3,a_4), (b_4,a_5), (b_5,a_2)\}$. We leave it to the reader to check these details except to say that if the proposers come from Set A, four matches are established in the first round but it takes three rounds for a_2 to be matched with b_3.

We have learned that when using the Gale-Shapely algorithm we can have two stable solutions. Consider the following example, with two sets of people consisting of three people. We will start with the men and women setting up their preference list.

Men's Preferences

M_a	X	Y	Z
M_b	Z	Y	X
M_c	X	Z	Y

Women's Preferences

W_x	B	A	C
W_y	C	B	A
W_z	A	C	B

If the men do the proposing, we can find the following stable pairings: S={(A,X),(B,Y),(C,Z)}. Using the same preference lists, when the women do the proposing we can find the following stable pairings: S={(A,Z),(B,X),(C,Y)}. However we can also find the stable pairing of S={(A,Y), (B,Z),(C,X)}. I.e. this stable marriage problem has three solutions. Thus, there can be more solutions than the ones obtained using the Gale-Shapely algorithm.

As we have demonstrated with our examples, using the Gale-Shapely algorithm we sometimes find more than one stable solution: one when the members of Set A propose or one when the members of Set B propose. However, these sometimes produce the same result. When you come across a case when you get the same solution whether the members of either set are doing the proposing, this is called a unique stable marriage pairing. In this situation, we would not care which group is proposing, Set A or Set B: the pairing is the same. Consider the following example in which we have been given the preference lists below:

Men Preferences

	b_1	b_2	b_3	b_4	b_5
a_1	**1**	3	**5**	4	2
a_2	2	**1**	3	**5**	4
a_3	**4**	2	**1**	3	**5**
a_4	5	**4**	2	**1**	3
a_5	3	5	**4**	2	**1**

Women Preferences

	a_1	a_2	a_3	a_4	a_5
b_1	**1**	2	**4**	5	3
b_2	3	**1**	2	**4**	5
b_3	**5**	3	**1**	2	**4**
b_4	4	**5**	3	**1**	2
b_5	2	4	**5**	3	**1**

In one round, with men proposing or women proposing we will end up with the same pairings

$$S = \{(a_1,b_1),(a_2,b_2),(a_3,b_3),(a_4,b_4),(a_5,b_5)\}.$$

This is called a unique stable pairing.

In general the Gale-Shapely algorithm does not depend on how many elements are in each set, we could have any number of elements in each set. If we have *n* elements in each set, the preference lists would have the following form:

Set B Preferences

	a_1	a_2	a_3	.....	a_n
b_1	1	n	2	◊	3
b_2	n	3	◊	◊	1
b_3	3	2	◊	◊	n
....	1	3	n	◊	2
b_n	2	1	n	◊	◊

Set A Preferences

	b_1	b_2	b_3	…..	b_n
a_1	3	n	1	◊	2
a_2	n	2	3	◊	1
a_3	3	1	◊	◊	n
….	◊	2	n	◊	1
a_n	1	3	n	2	◊

It Works!

In this section we will show that this is true and that the Gale-Shapely algorithm will always end. Also, we will see that the Gale-Shapely algorithm will always find stable marriage pairings. First we show that the algorithm will end. Remember that a woman will accept her first proposal, and will always remain engaged from that point on although she might “trade up” for a better husband. Also, every man will propose to a woman, or will currently be engaged to a woman in each round. Since we have the same number of men and women, we know that we cannot have a single male without having a single female. If these two single individuals did exist, then the man would have to propose to the woman at some point as she is listed on his preference list. Thus, they would become engaged. This ensures the algorithm will end.

At this point, if one considers all the possible stable marriage problems, we might ask, how many proposals and how many rounds are possible when we use the Gale-Shapely algorithm? We can easily say that the minimum number of proposals would be n and the minimum number of rounds would be one, as there might be a perfect, stable match of n pairs on the very first round. This would happen if each man chooses a different woman as his first choice. What are the maximums? For example, does the algorithm always end with the number of rounds being no more than the number of proposers?

Consider an example with 20 men and 20 women. Each man must make a proposal to each woman on his list until his proposal is accepted. We know that each man can make a maximum of 20 proposals, one to each woman. Thus, it is not possible to have more than $20^2 = 400$ proposals by which time every woman has received and accepted a proposal.

In order to obtain a better upper bound on the number of proposals that can take place, assume we have n men. Of course, each man can make n proposals. We thus know that the most proposals that we can have are n^2 proposals. Looking at this more closely however, we can show a better upper bound on the number of proposals that can take place based on the number of people involved. If we consider a scenario where every man has proposed to n -1 women, then either every woman has received a proposal and thus every woman has accepted a proposal and the game is over, or there is one woman who has never received a proposal. Since there are n men, the maximum number of possible proposals must be $n(n-1)+1$, where we have added one to account for the last woman to whom a proposal might be extended.

Is this relatively large number of proposals a best possible upper bound? For example, in our example above, we would have 20(20-1) +1 = 381 proposals. This number of proposals seems surprisingly high.

We are going to refer back to a previous example when we used three men and three women. Listed below are new preferences choices for a new stable marriage problem.

Men's Preferences	Women's Preferences
$\mathbf{M_a}$**:** X,Z,Y	$\mathbf{W_x}$**:**C,B,A
$\mathbf{M_b}$**:** X,Z,Y	$\mathbf{W_y}$**:**C,B,A
$\mathbf{M_c}$**:** Z,X, Y	$\mathbf{W_z}$**:** A,C,B

Using the Gale-Shapely Algorithm observe:

Round 1:

A proposes to X		
B proposes to X	X accepts B	
C proposes to Z	Z accepts C	Pairs are (B,X) and (C,Z)

Round 2:

A proposes to Z	Z switches to A	Pairs are (B,X) and (A,Z)

Round 3:

C proposes to X	X switches to C	Pairs are (A,Z) and (C,X)

Round 4:

B proposes to Z	Z rejects B	Pairs remain the same

Round 5:

B proposes to Y	Y accepts B	Pairs are (A,Z), (B,Y) and (C,X)

Notice we had 3 people in each group, and it took us 7 proposals and 5 rounds to find our set of stable matching. Since this works for 3-person sets, it is reasonable to assume that similarly large numbers of proposals and rounds (relative to the size of n) are possible when more people are in a set.

Thus, if $P(n)$ represents the number of proposals made by using the algorithm for n women and n men, we know $n \le P(n) \le n(n-1)+1$ and we believe it is likely the case that the upper bound can be realized for all n.

The maximum number of rounds according to Gale and Shapely is $n^2 - 2n + 2$. (Gossett, p.225)

Looking back at our last example, with three people in each of our two sets, you can see that not only did it take 7 proposals but it did indeed take 5 rounds. Note also that this is the maximum as proved by Gale and Shapely, i.e. $R(3) = 3^2 - 2(3) + 2 = 5$ rounds. While a proof for all n is beyond the level of this paper, we do believe that if $R(n)$ represents the number of rounds made by using the algorithm for n women and n men, the best possible lower and upper bounds are given by $1 \le R(n) \le n^2 - 2n + 2$.

The solution for the stable marriage problem given by the Gale-Shapley algorithm will always end with stable pairings. To see that the algorithm always results in a stable pairing, consider a man and woman who are engaged to separate people, but prefer each other. Since the women will always accept the proposal from the man higher on her preference list, one of three things might have happened. One is that the man has not proposed to the woman, in which case the woman he is with is higher on his preference list. There is also the possibility that the man has proposed to the woman. Since they are not together, she either broke off an engagement with him for a man higher on her preference list or said no because she is currently with a man higher on her list. In either of these cases, both cannot prefer each other to their current partners.

<u>Men or Women</u>

The Gale-Shapely algorithm favors the group doing the proposing. Thinking back to the first 3-person example, we found three solutions. Whoever did the proposing ended up with their higher choice for a spouse. Consider the following example in which we have two men and two women. The preference lists for this example are listed below:

Men preferences	Women preferences
a_1: b_1, b_2	b_1: a_2, a_1
a_2: b_2, b_1	b_2: a_1, a_2

Here a_1 would propose to b_1, and b_1 would accept. Similarly a_2 would propose to b_2, and b_2 would accept. Our pairs are then S= {(a_1, b_1), (a_2, b_2)}. In this example, the men get their first choice; however, the women have their second choice. This is called a male-optimal and female pessimal outcome. A person's optimal mate is that person's favorite from the list of possibilities. A person's pessimal mate is that person's least favorite from the list of their possibilities.

Let us consider what the outcome would be if the women would have been the proposers. In this scenario, b_1 would propose to a_2, and a_2 would accept. Also, b_2 would propose to a_1, and a_1 would accept. Our pairs are then S= {(b_1, a_2),(b_2, a_1)}. If the women were the ones proposing first, they would have their first choice, and the men would have their second choice.

<u>Does it pay to cheat?</u>

We ask, does it pay to cheat? It might seem reasonable that there is no reason for men or women to lie when reporting their preference lists when using the Gale-Shapley Algorithm. We will see, however, that although the proposers should be honest, there are situations where the people receiving proposals (in our scenario, the women) can benefit themselves by lying.

Case 1:

We will start with men proposing and show they do not benefit from lying. There is no reason for a man to lie when making his preference list. He should always make the list with his top choice. Since men have to propose by going down their list, they should tell the truth so as not to end up proposing to someone who is not their highest available preference. This may result in ending up with a woman who has the man high on her list, so she would accept the proposal. The man then could not ask another woman.

Case 2:

Now we will consider the case when men are proposing and see if women can benefit from lying. Women do have a strategy they can use to change the outcome. A woman can create false preference lists to try to get her best choice. Then the proposals made by the men, may work out to the woman's advantage that the man she wanted will propose to her. However the woman could possibly end up with a less than desirable choice.

Consider the following example:

Men Preferences:	Women's Preferences:
$\mathbf{M_a}$**:** X,Y,Z	$\mathbf{W_x}$**:** B,A,C
$\mathbf{M_b}$**:** Y,X,Z	$\mathbf{W_y}$**:** A,B,C
$\mathbf{M_c}$**:** X,Y,Z	$\mathbf{W_z}$**:** A,B,C

If the men are the proposers, this example would end with a set of stable matchings of:

$$S=\{(A,X),(B,Y),(C,Z)\}$$

Note: Man A is paired with Woman X, this is her second choice.

If woman X changes her preference list (as a lie) to the following:

$\mathbf{W_x}$**:** B,C,A

Again the men do the proposing, this example would end with a set of stable matchings of:

S={(A,Y),(B,X),(C,Z)}

Note: Man B is now paired with Woman X, which is her first choice.

Woman X ended up with her better choice by lying. Our example shows that women can benefit from lying; however, this is not easily done and may not work out.

Unequal men and women

When there are unequal numbers of men or women, we will not end up with everyone in a pairing, but we can still end with stable marriages. We would still have everyone make their ranked preference lists. In this scenario, the preference lists of the smaller group will be longer than the preference group of the other larger group. In the example we consider, it is smaller group that does the proposing.

We slightly alter the Gale-Shapely algorithm so that a member of the larger set may say no if he or she does not find a member of the smaller set who is acceptable. We will still be able to find stable pairings. The proposers will end up with the highest ranking member of the other set who accepts them, so they will be happy. Those receiving proposals have a chance to change engagements if someone higher on their preference lists proposes in a later round. The Gale-Shapely algorithm pairings will end when every man has proposed to everyone on his list.

Suppose we have 4 men and 6 women with the following preference lists:

Men Preferences

	b_1	b2	b_3	b_4	b_5	b_6
a_1	3	1	2	4	5	6
a_2	3	1	4	2	6	5
a_3	1	5	4	2	3	6
a_4	1	2	6	5	4	3

Women Preferences

	a_1	a_2	a_3	a_4
b_1	no	1	no	2
b_2	1	2	no	no
b_3	3	2	1	4
b_4	2	1	3	4
b_5	1	2	no	no
b_6	2	3	no	no

The table below represents the proposals and outcomes of the Gale-Shapely algorithm to this example:

Men Proposing:	Round1	Round2	Round3	Round4 :	Outcome
a_1	b_2				(a_1,b_2)
a_2	b_2	b_4			(a_2,b_4)
a_3	b_1	b_4	b_5	b_3	(a_3,b_3)
a_4	b_1				(a_4,b_1)

Thus the outcome is represented by S= $\{(a_1,b_2),(a_2,b_4),(a_3,b_3),(a_4,b_1)\}$. Notice the pairings are still stable.

A slight variation of the stable marriage problem is the stable roommates problem. This problem is different from the stable marriage problem in that there is only one group of people each of whom would make their preference lists from 1 to *n*-1. This could be rephrased as the same-sex stable marriage problem. As an example will show, it is possible that a stable matching does not exist.

Similarly to the Gale-Shapley algorithm, we might permit people to propose to their first choice all at once. If their proposal is rejected, they would continue to propose to the next person on their list. If a person has a roommate or receives a proposal from someone they prefer the most, then he or she will reject other proposals. Since each proposer is also receiving proposals, it is not clear how proposals are accepted. One person may “accept” a proposal from a person who is in the process of accepting a proposal from someone else. As we will see, it is possible that no stable matching can take place, so there is no modification of the algorithm that works. We look at two examples of the roommate problem, one that gives a set of stable pairings and one that does not.

Case 1:

In this first case we will look at when we are not able to find a stable matching in the roommate problem.

Suppose we have 4 roommates who are to be paired into two separate rooms:

Roommate preferences:

Chase:	Travis	Nick	Roman
Travis:	Nick	Chase	Roman
Nick:	Chase	Travis	Roman
Roman:	Chase	Travis	Nick

Possible Pairings:
(Chase, Travis) and (Nick, Roman) Travis and Nick would rather be together so not stable.
(Chase, Nick) and (Travis, Roman) Travis and Chase would rather be together so not stable.
(Chase, Roman) and (Travis, Nick) Chase and Nick would rather be together so not stable.

In this case we cannot find a stable roommate pair. Note that Roman is everyone's third choice. Because Chase prefers Travis, Travis prefers Nick and Nick prefers Chase, no matter who is paired with Roman would prefer to be paired with one of the other two who would also prefer to be paired with that person than the person they are paired with.

Case 2:

Now we will look at an example that the stable matching will work for to find stable roommates.

Given 6 roommates paired into three separate rooms:

Roommate preferences:

a_1:	a_3	a_4	a_2	a_6	a_5
a_2:	a_6	a_5	a_4	a_1	a_3
a_3:	a_2	a_4	a_5	a_1	a_6
a_4:	a_5	a_2	a_3	a_6	a_1
a_5:	a_3	a_1	a_2	a_4	a_6
a_6:	a_5	a_1	a_3	a_4	a_2

After about 3 rounds and some adjusting, we can find the following stable matching pairs

$S = \{(a_1,a_6), (a_2,a_4), (a_3,a_5), (a_4,a_2), (a_5,a_3), (a_6,a_1)\}$.

<u>Real-World applications</u>

One of the most popular applications for the stable marriage problem involves fourth year medical students and hospitals. We refer to this as the National Resident Matching program.

Students go to hospitals and are interviewed. Students and hospitals then make their own preference lists. In this scenario, the hospitals and students are two disjoint groups, where the hospitals are to be considered the proposers. By the end of the algorithm, the hospitals get their better picks as they are the proposers, and the students might end up with their lesser pick. One difference between this problem and the stable marriage problem is that the hospitals are picking more than one student.

The Gale-Shapeley algorithm has also been used to match sailors to ships, and pupils to specialized trade schools. In its one gender form, it is used to help to pair roommates in colleges. Also it can probably be used in online dating services.

Conclusion

Stable marriage pairings are created using the Gale-Shapely algorithm or modifications of the algorithm, whether we have the same number of people in two groups of the opposite sex, unequal numbers of people, or single groups of people. Everyone involved will create a preference list. The proposers will begin proposing and ultimately have their best choices in the pairings. Those being proposed to must accept their first proposal, and then the proposal of anyone who is higher on their preference list than their current proposer. When we are working with two sets of equal size, this process will always create a stable pairing without any blocking pairs. A version of the Gale-Shapely algorithm has been used to help in pairing hospital interns with hospitals and roommates in colleges.

Implications

I may try this as a demonstration in my 8th grade math classroom. I was thinking also of seeing if it would work to assign students as project partners. I would have them try it more as a roommate problem than as split in two groups.

References:

Gossett, E. (2009)Discrete Mathematics with Proof. (2nd ed). Hoboken, NJ. Wiley &Sons, Inc. p. 225

Hirata, J. Notes on Matching. Retrieved on June 7th, 2011. http://www-math.mit.edu/~djk/18.310/Lecture-Notes/MatchingProblem.pdf

Hunt, W. The Stable Marriage Problem, Lane Department of Computer Science and Electrical Engineering, West Virginia University, WV Retrieved on June 6th 2011. http://www.csee.wvu.edu/~ksmani/courses/fa01/random/lecnotes/lecture5.pdf

Mairson, H. (1992) The Stable Marriage Problem. Retrieved on June 6th, 2011 http://www.cs.columbia.edu/~evs/intro/stable/writeup.html

National Resident Matching Program. Wikipedia. Retrieved June 8th, 2011 http://en.wikipedia.org/wiki/National_Resident_Matching_Program