

Genetic Algorithm for the Stable Marriage Problem (SMP)

Isah Bello¹, Shi Lianshuan²

Tianjin University of Technology and Education, School of Information Technology and Engineering, Tianjin, P.R. China.

Abstract: *The Genetic Algorithm is one of the most important techniques used to solve many combinatorial optimization problems. This paper proposed a Genetic Algorithm for the Stable Marriage Problem. The goal of Stable Marriage Problem is to find a stable matching between two sets (men and women) with various preferences of each other. The Genetic Algorithm results for the Stable Marriage Problem were then compared to Gale-Shapley algorithm's results. The result of the comparison shows that Genetic Algorithm has better results for happiness and egalitarian happiness. There is however a small drop in the number of stable pairs with the Genetic Algorithm as compared to the Gale-Shapley result. This goes in accordance with the theory since every person tries to get as best matching as possible and that is at the expense of stability (Caldarelli and Capocci, 2000).*

Keywords: Genetic Algorithm, Stable Marriage Problem, Fitness function

1. Introduction

1.1 Stable Marriage problem (SMP)

The Stable Marriage Problem (SMP) is the best known stable matching problem which is a classic mathematics problem. The goal is to find a stable matching's between two sets (men and women) with various preferences of each other.

The problem of finding a stable matching started in a paper published in 1962 by Gale and Shapley (1962, p.11) and it can be defined as follows

"A certain community consists of n men and women. Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner: we seek a satisfactory way of marrying off all members in the community. We call a set of marriages unstable if under it there are a man and a woman who are not married to each other but prefers each other to their actual mates."

The Stable Marriage Problem (SMP) has been studied and researched by a large number of scholars in the previous decades. The first ones to discover a stable solution for the problem were D.Gale and G.S.Shapley (1962) by the introduction of their algorithm, called the GSS algorithm, in their paper "College Admissions and the Stability of Marriage". Since their findings, many other aspects and variations of the problem have been examined and solved.

The GSS algorithm works like this:

In GSS algorithm, each man proposes to his most preferred mate that he has not already proposed to. Each woman considers all proposals she has received and rejects all but her most preferred who she keeps on hold. Each man who has been rejected proposes to the next woman on his list. This process continues until either no men are rejected or all men who have rejected have proposed to every woman on their lists. All women are paired with the man they currently have on held. A woman who has not received a proposal or a man who has had all of his proposals rejected remains single.

1. assign each person to be free;
2. while some man m is free and m has a nonempty list loop
3. $w :=$ first woman on m 's list ; m proposes to w
4. if m is not on w 's preference list then
5. delete w from m 's preference list;
6. go to line 3
7. end if
8. if some man p is engaged to w then
9. assign p to be free;
10. end if
11. assign m and w to be engaged to each other;
12. for each successor p of m on w 's list loop
13. delete p from w 's list;
14. delete w from p 's list;
15. end loop;
16. end loop;

Figure 1: The GSS algorithm

The GSS algorithm always finds a stable matching. In stable matching, there is no man-woman pair that would prefer each other compared to their current matchings. However, there are some problems with the original algorithm. First of all, it produces either the man optimal or the woman optimal result, depending on which group is proposing."A stable marriage is called optimal if every applicant is at least as well off under it as under any other stable assignment."(Gale and Shapley, 1962, p.10).

The problem with the man optimal result is that, it is also the result where every woman gets her worst choice partner, meaning marriages are stable but all the women are highly unsatisfied with their partners, and vice versa for the woman optimal matching. A great amount of articles and books have been written about the stable marriage problem. Nowadays, there are variations of the SMP which have been proven to be NP-hard. Therefore, approximation algorithms were suggested to deal with the NP-hard variations of the problem. An approximation algorithm is an algorithm that because no polynomial time solution can be found, it settles for an optimal solution, of a specific optimality of a subset of the problem. Genetic Algorithm is the best known approximation algorithm.

The Stable Marriage Problem (SMP) has many real-world applications. Some applications of the SMP can be found in College Admission, the sailors-boats problem, the stable roommates' problem, application in router technology, e.t.c.

1.2 Genetic Algorithm (GA)

The basic principles of Genetic Algorithms (GAs) were introduced by John Holland in 1975. The GAs are an optimization and search techniques which emerged from a study of biological evolution. Genetic Algorithms operate on a "population" of potential solutions, which are usually referred to as "chromosomes". Each chromosome represents a set of parameters for a given problem. The chromosomes evaluate to represent the best solutions for a recombination process, which produces new chromosomes. The new improved chromosomes replace those with poorer solutions. In this way, each generation becomes closer to the optimal solution. This continues for many generations until the termination condition is met. Mutations and different combining strategies ensure that a large range of search space is discovered.

2. Related Work

In Roth's and Vande Vate's work (1990) a sequence of steps was described for finding a stable matching that is similar to the process the GA uses. They used the following strategy, themselves following the example of Knuth (1976).

Beginning from a random matching, Roth and Vande Vate (1990) located the first blocking pair for the matching.

A blocking pair is a pair of people that prefer each other to their current partners, and because of them the matching is unstable (Roth and Vande Vate, 1990). A way to measure how unstable a matching is, is by counting the blocking pairs. By altering the matching in a way that the two people are matched together one can come to a new matching, which might be stable or might contain more blocking pairs. By continuing in the same way, a stable matching will eventually be found. The GA basically follows the same pattern, but it changes more than one pair in every generation.

Aldershof and Carducci (1999), modeled a coding of the SMP and the hospital couples problem, through the use of a GA they created, and their goal was to solve the hospital/residents couples problem. They achieved that goal by representing the problem in the form of a bit matrix X of dimensions h and p , where h is the number of hospital positions, and p the number of students. If a student and a hospital are matched then the value of $X(h, p) = 1$, otherwise it is 0. That matrix was then translated into a string where every position represented a hospital, and the number located in that position represented the student that

was assigned to that hospital. They represented the rest of the problem in the form of inequalities. If all those inequalities are satisfied by a chromosome X then that chromosome constitutes a stable marriage. Moreover, in their initialization and reproduction functions they ensured that their population produces acceptable pairs (Aldershof and Carducci, 1999) that is pairs that have each other on their preference lists.

As far as the couple's problem is concerned, they separated the applicants set into three categories. Those who are single applicants and two sets of men and women for the couples that creates a married couple, which then is matched to a hospital. More inequalities (Aldershof and Carducci, 1999) were then introduced for this problem.

For fitness function, they simply created a function that adds the number of inequalities that were satisfied. This helped them locate solutions for the couple's problem where a stable matching might not exist, but it is the best available solution might be located anyway, even though it might lack stability. For a mating/crossover function they used cyclic crossover.

As far as mutation was concerned, they chose to use a function that randomly or after finding an unstable pair in a chromosome, performs the change. Mutation is of importance, because simply changing a random number in the string may result in an illegal chromosome.

Their results were finding all matches in the SMP and a student-optimal matching but no hospital-optimal matching in the couples's problem. Also, it appeared that singles get more satisfactory results than couples, and couples had a larger probability of being unmatched in the end of the algorithm, which can simply be explained by the fact that, it is easier to satisfy the pre-requisites of a single than of two persons. They also emphasized on the need for an algorithm with specific criteria for deciding among matchings.

3. Methodology

3.1 Encoding

The standard bit-string representation of simple Genetic Algorithm is not easy to be use in Stable Marriage Problem (SMP). In this work, the choice of chromosome type used is vector of positive integers that represent men matched to the women. For example, for 4 men and 5 women, the vector [4, 3, 1, 5, 2] can be translated as the 1st woman is matched to the 4th man, the 2nd woman is matched to the 3rd man, the 3rd woman is matched to the 1st man, the 4th woman is matched to the 5th man, the 5th woman is matched to the 2nd man.

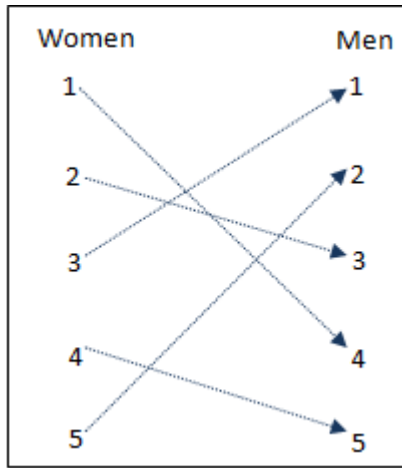


Figure 2: Women matched to men

3.2 Population Initialization: In order to initialize population in Genetic Algorithm for the SMP, there are two parameters that the GA must have; the encoded chromosome which in this research is the population vector and the preference lists. The preference lists are created once and remain the same throughout the iteration. Both population vector and the preference lists are created randomly. For a matching between 4 men and 4 women, the preference lists look like this.

Table 1: Preference lists

Rank	1	2	3	4
M1	W1	W4	W3	W2
M2	W1	W2	W3	W4
M3	W2	W4	W3	W1
M4	W3	W1	W2	W4
W1	M3	M1	M2	M4
W2	M1	M4	M3	M2
W3	M2	M4	M3	M1
W4	M4	M2	M1	M3

3.3 The Fitness Function

The fitness function typically represents the objective function that we want to optimize in the problem. In this research, the target is to get a matching with stable pairs, happiness for every person, and the egalitarian happiness. If M is a stable matching between n men and women, the position of a woman w in a man's preference list is $P_m(w)$, and for woman $P_w(m)$ respectively. We can define the happiness cost $h(M)$ and the egalitarian happiness $eh(M)$, for any instance of stable marriage problem with the formulas (i) and (ii) below (Iwama and Miyazaki, 2008).

Happiness cost

$$h(M) = \sum P_m(w) + \sum P_w(m) \quad 1$$

Egalitarian cost

$$eh(M) = \sum P_m(w) - \sum P_w(m) \quad 2$$

We can define the happiness per person if we divide the above numbers with the number of participants. Formulas (iii), (iv), and (v) gives us the happiness per person (hpp) for men or women as the case may be, the happiness per couple

(hpc) and the egalitarian happiness per couple (ehc) (Caldarelli and Capocci, 2000).

Happiness per person

$$hpp(M) = \frac{1}{N} \left(\sum P_m(w) \right) \quad 3$$

Happiness per couple

$$hpc(M) = \frac{1}{N} \left(\sum P_m(w) + \sum P_w(m) \right) \quad 4$$

Egalitarian happiness per couple

$$ehc(M) = \frac{1}{N} \left(\sum P_m(w) - \sum P_w(m) \right) \quad 5$$

Where N is the number of men/women.

For individual say x , the fitness function formula is given as follows:

$$F = \text{stable pairs}(x) + \text{hap}(x) + \text{ehap}(x) \quad 6$$

To weight the contributions of every optimality criterion to the fitness value, three constants S , H , and E are multiplied to $\text{stable pairs}(x)$, $\text{hap}(x)$ and $\text{ehap}(x)$ respectively.

Therefore the fitness function becomes:

$$F(x) = S * \text{stable pairs}(x) + H * \text{hap}(x) + E * \text{ehap}(x) \quad 7$$

The values S , H and E represent how much every criterion affects the final fitness score. The more we adjust the value of one of these three, the more significant is any increase or decrease of that criterion, meaning the fitness function focuses more on minimizing that specific one. By setting $S=H=E=1$, we acknowledge that all three criteria play the same role and we search for a solution which compromises for a stable matching that has good happiness and egalitarian happiness values. By setting $H=E=0$, we can search only for stable matchings. By setting $E=0$, we would obtain better happiness solution, and when $H=0$, the algorithm returns a better egalitarian solution.

3.4 Selection

Selection is a method that randomly picks chromosomes out of the population according to their fitness function. The higher the fitness function, the more chance an individual has to be selected. There are various selection methods, Roulette wheel selection is used in this research.

In Roulette wheel selection method, each individual is assigned a slice of a circular "roulette wheel", the size of the slice being proportional to the individual fitness. The wheel is spun N time, where N is the number of individuals in the population. On each spin, the individual under the wheel marker is selected to be in the pool of parents for the next generation.

3.5 Crossover

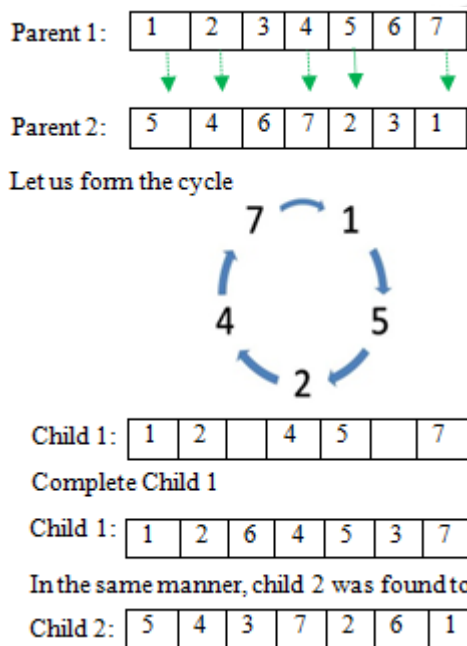
Crossover is the process of taking two parent solutions and producing from them a child. The traditional one point or points crossover are not suitable for the Stable Marriage Problem. If we choose to implement any of the two methods above, we would get disastrous results. For example, let

Parent 1: 1 2 4 3 6 5 and Parent 2: 6 3 1 4 2 5 be parents with chromosome length of 6. If we apply one-point

crossover after point 3, we would come with the following off springs ;child 1: 1 2 4 3 2 5 and child 2: 6 3 1 4 6 5,which contain the same number twice and thus destroys the chromosomes rendering them not valid representation for the SMP.In order to address the problem encounter above, Andershof and Carducci proposed and use a reliable crossover technique called “Cyclic crossover” to solve the Stable Marriage Problem (SMP).

The cyclic crossover operates as follows:-

- Step 1: find the cycle defined by the parents.
 - Step 2: copy from parent 1 elements within the cycle for the first child.
 - Step 3: fill the rest elements from parent 2 to complete the first child.
 - Step 4: proceed in similar way but filling first with parent 2 to produce the other child.
- For better understanding of how the cyclic crossover works, let us consider the example below:

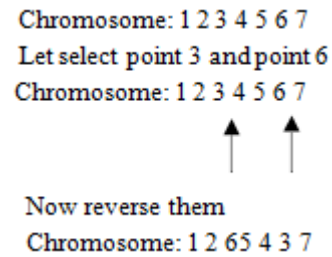


One disadvantage of this method is that, although it guarantees the reproduction of acceptable off springs, the children due to the strict positioning rule, clones of the same chromosome are created. This happen mostly when the generating function creates chromosomes randomly. The use of a slightly increased rate of mutation helps to avoid that danger and also ensure diversity.

3.6 Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. There are various mutation methods; inversion mutation is used in this thesis. Inversion mutation works as follows:

- Step1: select two random points
 - Step 2: reverse them
- For better understanding of Inverse mutation, let us consider this example.



4. Experiment

• Implemented algorithm for comparison.

Gale-Shapley algorithm was implemented as a means of comparison with the Genetic Algorithm. It always yields the man-optimal solution, so I know beforehand that unless the man-optimal solution is the same with the happiness solution, it always give results that are not happiness optimal. So I am quite optimistic that the GA implementation would perform better compare to the GSS algorithm.

• Algorithms settings

Both the GSS and GA algorithms were run with different sets of values for each and every setting. The solutions of both GSS and GA were studied and then compared accordingly. Because of the nature of GA, it is not guarantee that it always succeeds in finding all the possible stable matchings, but in most cases it always finds at least one solution that is of good quality as we are going to see.

Table 2: GSS results for n= 10 for SMP

Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
10	3.0	2.5	5.5	0.6
10	1.6	4.1	5.7	2.5
10	2.7	3.0	5.7	0.4
10	1.9	4.0	5.9	2.1
10	2.1	3.7	5.8	1.9
Average values				
10	2.26	3.46	5.72	1.5

Table 3: GSS results for n= 30 for SMP

Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
30	4.0	7.56	11.56	3.10
30	3.43	8.4	11.83	5.6
30	2.83	10.9	13.73	9.0
30	5.0	6.44	11.44	2.24
30	3.34	8.80	12.24	6.3
Average values				
30	3.74	8.42	12.16	5.25

Table 4: GSS results for n= 50 for SMP

Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
50	4.28	13.4	17.68	9.96
50	5.1	12.98	18.08	8.88
50	4.62	17.0	21.62	12.76
50	4.94	13.1	18.04	9.0
50	5.44	12.14	17.58	7.48
Average values				
50	4.88	13.72	18.60	9.62

Table 5: Genetic Algorithm's settings

N	10	30	50
Population size	100	1000	2000
No. of iterations	100	1000	3000
Selection method	Roulette wheel selection		
Crossover rate	0.5		
Mutation rate	0.8		

Table 6: Genetic Algorithm results for n=10 for SMP

S=1 H=1 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
10	2.6	3.1	5.7	0.5
10	2.0	2.4	4.4	0.4
10	3.1	2.7	5.8	0.3
9	3.2	3.0	6.2	0.2
10	2.4	2.9	5.3	0.5
Average values				
9.8	2.66	2.82	5.58	0.38
S=1 H=0 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
10	2.6	3.1	5.7	0.5
10	2.0	2.4	4.4	0.4
10	2.9	3.0	5.9	0.2
9	3.2	3.2	6.4	0
10	2.4	2.9	5.3	0.5
Average values				
9.8	2.62	2.92	5.54	0.32
S=1 H=1 E=0				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
10	2.6	3.1	5.7	0.5
10	2.0	2.4	4.4	0.4
10	3.1	2.8	5.9	0.3
10	3.3	2.7	6.0	0.5
10	1.4	2.6	4.0	2.2
Average values				
10	2.48	2.72	5.2	0.78
S=0 H=1 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
9	2.7	2.6	5.3	0.1
8	1.9	2.2	4.1	0.2
7	2.8	2.9	5.7	0.1
6	2.9	3.0	5.9	0.1
9	2.5	2.1	4.6	0.2
Average values				
7.8	2.56	2.56	5.12	0.14

Table 7: Genetic Algorithm results for n=30 for SMP

S=1 H=1 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
29	4.66	5.42	10.08	0.62
29	4.42	4.98	9.40	0.62
28	5.44	5.54	10.98	0.08
28	4.68	4.88	9.56	0.26
28	5.26	4.78	10.04	0.55
Average values				
28.4	4.89	5.12	10.01	0.43
S=1 H=0 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
27	5.82	5.86	11.68	0.04
30	4.43	4.8	9.23	0.26
28	5.42	5.62	11.04	0.26
30	4.64	4.94	9.58	0.4
30	4.78	4.86	9.64	0.08
Average values				

29	5.02	5.22	10.24	0.21
S=1 H=1 E=0				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
30	3.42	6.84	10.28	3.5
28	4.80	4.82	9.62	0.04
29	3.78	7.0	10.78	3.54
27	3.56	4.96	8.52	1.46
30	4.87	4.72	9.59	0.08
Average values				
28.8	4.09	5.67	9.76	1.72
S=0 H=1 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
24	4.78	4.78	9.56	0
22	4.80	4.78	9.58	0.04
22	5.27	5.28	10.55	0.04
25	4.10	4.39	8.49	0.24
24	4.72	4.86	9.58	0.02
Average values				
23.4	4.73	4.82	5.12	0.07

Table 8: Genetic Algorithm results for n=50 for SMP

S=1 H=1 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
41	8.32	8.31	16.63	0.04
37	8.53	8.42	16.95	0.12
40	8.78	9.06	17.84	0.21
43	7.22	7.50	14.72	0.28
41	8.96	9.12	18.08	0.13
Average values				
40.4	8.36	8.48	16.84	0.16
S=1 H=0 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
41	11.12	11.10	22.22	0.02
38	12.34	12.40	24.74	0.05
41	11.36	11.43	22.79	0.08
38	11.34	11.32	22.66	0.05
39	12.28	12.24	24.52	0
Average values				
39.4	11.69	11.70	23.39	0.04
S=1 H=1 E=0				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
43	5.38	9.86	15.24	4.58
43	6.40	7.34	13.74	1.08
43	5.62	11.28	16.9	5.70
42	6.38	9.86	16.24	3.60
43	6.34	8.72	15.06	2.22
Average values				
42.8	6.01	9.36	15.37	3.44
S=0 H=1 E=1				
Stable pairs	Hap(m)	Hap(w)	Hap(m+w)	Ehap
30	6.92	7.04	13.96	0.10
33	7.32	7.41	14.73	0.06
28	7.02	7.06	14.08	0.02
32	6.86	6.66	13.52	0.00
30	7.42	7.54	14.96	0.11
Average values				
30.6	7.11	7.14	14.25	0.06

5. Result Discussion and Conclusion

5.1 Result Discussion

Looking at the results we can see that for all criteria GA results for happiness and egalitarian happiness are better compared to the GSS algorithm's results. The results are very close to zero which is the optimal as seen most especially for the egalitarian happiness. There is however a small drop in the number of stable pairs. This is in accordance to the theory since every person tries to get as best matching as possible and that is at the expense of stability (Caldarelli and Capocci, 2000). This is seen where $S=0, H=1$ and $E=1$ for all n . For these settings, the results are much better as far as happiness and egalitarian happiness are concerned. This is one of the reasons of using this combination of settings even though it doesn't take into account the number of stable pairs in the fitness function. On all other combination of weight values, we can observe that there is a high improvement of happiness and egalitarian happiness at the cost of a few stable pairs. This proves that Genetic Algorithm can get better results with different properties if we use a fitness function containing more than one criteria. Moreover, if we compare the solutions to each other, we can see that when either H or E is absent from the fitness function, we get a greater improvement in the value of the remaining criterion. This is quite evident from the results. That way we have either a generally good solution for all criteria ($S=H=E=1$), or have a better happiness ($E=0$) or a better egalitarian happiness solution ($H=0$). It can also be noticed in the results for $n=50$ that the number of stable pairs drops more than in the previous experiments, this is due to the fact that GA cannot find all stable pairs in a reasonable running time for higher problem sizes. However, in this research stability was not considered more important than the other two criteria (Happiness and egalitarian happiness).

5.2 Conclusion

In this paper we discussed a genetic algorithm for the Stable Marriage Problem. Genetic Algorithm appears to find good solutions for the SMP, however it depends very much on the way the problem is encoded and which crossover and mutation methods are used. It also depends on the fitness function which allows us to define it to suit our need. By experimental evaluation, we show the effectiveness of the proposed GA over the GSS algorithm.

References

- [1] Aldershof B. and Carducci O. M., (1999), Stable marriage and genetic algorithms: a fertile union. *J. Heuristics (Netherlands)*, 5(1):29-46.
- [2] Biro, P. and Manlove, D.F. and Mittal, S., (2010), Size versus stability in the marriage problem. *Theoretical Computer Science*, 411 (16-18). pp. 1828-1841. ISSN 0304-3975
- [3] Goldberg D.E., (1989), *Genetic Algorithms in Search, Optimization and Machine Learning* (1st Ed.). Addison- Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [4] Holland, J.H., (1992), *Genetic algorithms*, *Scientific American* Vol. 267, p. 66-72.
- [5] Caldarelli G. and Capocci A., (2000), Beauty and distance in the stable marriage problem. *Physica A: Statistical Mechanics and its Applications*, 300:325-331.
- [6] Gale D., Shapley L., (1962), College admissions and the stability of marriage, *Amer. Math. Monthly* 69 9-15
- [7] C.H. Papadimitriou and K. Steglitz. "Combinatorial Optimization: Algorithms and Complexity". Prentice Hall of India Private Limited, India, 1997
- [8] Gale D. and Sotomayor M., (1985), "Some remarks on the stable matching problem," *Discrete Applied Mathematics*, Vol.11, pp.223-232.
- [9] Yanagisawa H., (2007), *Approximation Algorithms for Stable Marriage Problems* PhD Thesis