Richard Booth · Min-Ling Zhang (Eds.)

LNAI 9810

PRICAI 2016: Trends in Artificial Intelligence

14th Pacific Rim
International Conference on Artificial Intelligence
Phuket, Thailand, August 22–26, 2016, Proceedings

Springer

Lecture Notes in Artificial Intelligence 9810

Subseries of Lecture Notes in Computer Science

Richard Booth · Min-Ling Zhang (Eds.)

PRICAI 2016: Trends in Artificial Intelligence

14th Pacific Rim
International Conference on Artificial Intelligence
Phuket, Thailand, August 22–26, 2016
Proceedings

Springer

Editors
Richard Booth
Cardiff University
Cardiff
UK

Min-Ling Zhang
Southeast University
Nanjing
China

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-319-42910-6 ISBN 978-3-319-42911-3 (eBook)
DOI 10.1007/978-3-319-42911-3

Library of Congress Control Number: 2016945115

LNCS Sublibrary: SL7 – Artificial Intelligence

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Contents

An Empirical Local Search for the Stable Marriage Problem

Hoang Huu Viet[1(✉)], Le Hong Trang[1], SeungGwan Lee[2], and TaeChoong Chung[3]

[1] Department of Information Technology, Vinh University, 182-Le Duan, Vinh City, Nghe An, Vietnam
{viethh,lhtrang}@vinhuni.edu.vn

[2] Humanitas College, Kyung Hee University, Seoul, South Korea
leesg@khu.ac.kr

[3] Department of Computer Engineering, Kyung Hee University, 1732, Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 446-701, Korea
tcchung@khu.ac.kr

Abstract. This paper proposes a local search algorithm to find the *egalitarian* and the *sex-equal* stable matchings in the stable marriage problem. Based on the dominance relation of stable matchings from the men's point of view, our approach discovers the *egalitarian* and the *sex-equal* stable matchings from the *man-optimal* stable matching. By employing a breakmarriage strategy to find stable neighbor matchings of the current stable matching and moving to the best neighbor matching, our local search finds the solutions while moving towards the *woman-optimal* stable matching. Simulations show that our proposed algorithm is efficient for the stable marriage problem.

1 Introduction

The stable marriage problem (SM) is a well-known problem of matching an equal number men and women to satisfy a certain criterion of stability. This problem was first introduced by Gale and Shapley [3], and has recently received a great deal of attention from the research community due to its important role in a wide range of applications such as the Evolution of the Labor Market for Medical Interns and Residents [12], the Student-Project Allocation problem (SPA)[1] and the Stable Roommates problem (SR) [2,7].

An instance of SM of size n comprises a set of n men and a set of n women and each person has a preference list (PL) in which they rank all members of the opposite sex in strict order. A matching M is a set of n disjoint pairs of men and women. If a man m and a woman w is a pair in M, then m and w are partners in M, denoted by $m = M(w)$ and $w = M(m)$. Matching M is stable if there is no man m and woman w such that m prefers w to $M(m)$ and w prefers m to $M(w)$, otherwise M is unstable. For a stable matching M, we define the *man cost* $sm(M)$ and the *woman cost* $sw(M)$ as follows:

R. Booth and M.-L. Zhang (Eds.): PRICAI 2016, LNAI 9810, pp. 556–564, 2016.
DOI: 10.1007/978-3-319-42911-3_46

$$sm(M) = \sum_{(m,w) \in M} mr(m, w), \;\; sw(M) = \sum_{(m,w) \in M} wr(w, m), \tag{1}$$

where $mr(m, w)$ is the rank of woman w in man m's PL and $wr(w, m)$ is the rank of man m in woman w's PL.

Definition 1 (Man-Optimal and Woman-Optimal [11]). *A stable matching M is called man-optimal (respectively woman-optimal) if it has the minimum value of $sm(M)$ (respectively $sw(M)$) over all stable matchings.*

Gale and Shapley proposed an algorithm known as the Gale-Shapley algorithm to find an optimal solution of SM instances of size n in time $O(n^2)$[3]. The Gale-Shapley algorithm is basically a sequence of proposals from men to women to find the *man-optimal* stable matching. If the roles of men and women are interchanged, the matching found by the algorithm is the *woman-optimal* stable matching. It is proved that in the *man-optimal* stable matching, each woman has the worst partner that she can have in any stable matching and that in the *woman-optimal* stable matching, each man has the worst partner that he can have in any stable matching [6]. Therefore, it is appropriate to seek other optimal stable matchings to give more balanced preference for both men and women. For a stable matching M, we define the *egalitarian cost* $c(M)$ and the *sex-equality cost* $d(M)$ as follows:

$$c(M) = sm(M) + sw(M), \;\; d(M) = |sm(M) - sw(M)|. \tag{2}$$

Definition 2 (Egalitarian and Sex-Equal [11]). *A stable matching M is called egalitarian (respectively sex-equal) if it has the minimum value of $c(M)$ (respectively $d(M)$) over all stable matchings.*

There are several methods to find the *egalitarian* or *sex-equal* stable matching based on local search approach [4,9,11,14]. Because the number of stable matchings of SM instances grows exponentially in general [8], the above methods either are inefficient for finding solutions or finds only one stable matching of SM instances of large sizes. In this paper, we propose a new local search algorithm to seek an *egalitarian* or *sex-equal* stable matching of a given SM. Start from the *man-optimal* stable matching, the proposed algorithm finds a better solution in the neighbors of the current solution. If a better solution is found, the current solution is moved to the better solution and the local search is repeated for the current solution until no neighbor matchings of the current solution are found. The simulation results show that our approach guarantees to find a stable matching, which is an optimal solution or a near optimal solution, and discovers the solution faster than the ACS approach [14] does.

The rest of this paper is organized as follows: Sect. 2 describes the background, Sect. 3 presents the proposed approach, Sect. 4 discusses the simulations and evaluations, and Sect. 5 concludes our work.

Table 1. Preference lists of eight men and women.

Man	Preference list	Woman	Preference list
m_1	4 3 1 5 2 6 8 7	w_1	4 7 3 8 1 5 2 6
m_2	2 8 4 5 3 7 1 6	w_2	5 3 4 2 1 8 6 7
m_3	5 8 1 4 2 3 6 7	w_3	2 8 6 4 3 7 5 1
m_4	6 4 3 2 5 8 1 7	w_4	5 6 8 3 4 7 1 2
m_5	6 5 4 8 1 7 2 3	w_5	1 8 5 2 3 6 4 7
m_6	7 4 2 5 6 8 1 3	w_6	8 6 2 5 1 7 4 3
m_7	8 5 6 3 7 2 1 4	w_7	5 2 8 3 6 4 7 1
m_8	4 7 1 3 5 8 2 6	w_8	4 5 7 1 6 2 8 3

2 Background

Consider an example of SM consisting of eight men and eight women with their preference lists shown in Table 1.

Definition 3 (Dominance [5]). *Let $M \in \mathcal{M}$ and $M' \in \mathcal{M}$ be two stable matchings. M is said to dominate M' under the men's point of view if and only if every man prefers his partner in M at least as well as to his partner in M'.*

Brute-Force Algorithm. In order to generate all stable matchings and show which one is the *man-optimal*, *woman-optimal*, *egalitarian* or *sex-equal* stable matching of the example in Table 1, we give a brute-force search algorithm consisting of three steps: (i) generate all of the permutations of the women set; (ii) pair each man to each woman in the permutations to form 40320 (i.e. 8!) matchings; and (iii) search on the whole matchings and check stable matchings with the given criterion. All stable matchings sorted by the dominance relation from the men's point of view and their cost of the example are shown in Table 2. Obviously, the proposed algorithm is simple to implement and it finds exactly solutions of SM of size n, but its cost is $n!$. Therefore, if the number of men or women is large then this algorithm is inefficient in terms of time complexity.

Gale-Shapley Algorithm [3]. The Gale-Shapley algorithm is basically a sequence of proposals from men to women to find the *man-optimal* stable matching. At the beginning, the algorithm assigns each person to be free. At each iteration step, the algorithm chooses a free man m and finds the most preferred woman w in m's list to whom m has not proposed. If w is free then w and m become engaged. If w is engaged to m' then she rejects the man that she least prefers to engage to the other man. The rejected man becomes free. The algorithm terminates when all men are engaged.

Breakmarriage Operation [10]. Let M be a stable matching and (m, w) be an engaged pair in M. The breakmarriage operation of (M, m) is to find a stable matching. The idea of the breakmarriage is similar to the Gale-Shapley algorithm.

Table 2. Evaluations of stable matchings.

Stable matchings	sm	sw	c	d
M_0 = {(1,1),(2,2),(3,5),(4,3),(5,6),(6,7),(7,8),(8,4)}	12	33	45	21
M_1 = {(1,1),(2,2),(3,5),(4,3),(5,6),(6,4),(7,8),(8,7)}	14	30	44	16
M_2 = {(1,5),(2,2),(3,1),(4,3),(5,6),(6,7),(7,8),(8,4)}	15	27	42	12
M_3 = {(1,5),(2,2),(3,1),(4,3),(5,6),(6,4),(7,8),(8,7)}	17	24	41	7
M_4 = {(1,5),(2,3),(3,1),(4,2),(5,6),(6,7),(7,8),(8,4)}	20	23	43	3
M_5 = {(1,5),(2,3),(3,1),(4,2),(5,6),(6,4),(7,8),(8,7)}	22	20	42	2
M_6 = {(1,5),(2,2),(3,1),(4,3),(5,4),(6,6),(7,8),(8,7)}	22	21	43	1
M_7 = {(1,5),(2,3),(3,1),(4,2),(5,4),(6,6),(7,8),(8,7)}	27	17	44	10
M_8 = {(1,5),(2,3),(3,2),(4,8),(5,6),(6,7),(7,1),(8,4)}	30	19	49	11
M_9 = {(1,5),(2,3),(3,2),(4,8),(5,6),(6,4),(7,1),(8,7)}	32	16	48	16
M_{10} ={(1,5),(2,3),(3,2),(4,8),(5,4),(6,6),(7,1),(8,7}	37	13	50	24
•M_0: *man-optimal*, M_{10}: *woman-optimal*.				
•M_3: *egalitarian*, M_6: *sex-equal*.				

At the beginning, the algorithm assigns the woman w to the partner of the man m and sets m to be free. At each iteration step, the algorithm performs a sequence of proposals, rejects and acceptances as those of the Gale-Shapley algorithm. The algorithm terminates either when some man has been rejected by all women or when the woman w accepts a man m' to whom she prefers to her partner and in this case the algorithm returns a stable matching of n engaged pairs [10]. The following theorem and corollary are the basis of our approach to SM.

Theorem 1 [10]**.** *Every stable matching* $M_i (i = 1, 2, \cdots, t)$ *can be obtained by a series of breakmarriage operations starting from the man-optimal stable matching* M_0*, where* M_t *is the woman-optimal stable matching.*

Corollary 1 [5]**.** *If breakmarriage*(M, m) *results in a stable matching* M'*, then* M' *dominates all stable matchings which are dominated by* M *and in which* m *is not married to his mate in* M*.*

3 Proposed Algorithm

In this section, we propose a local search algorithm to find an *egalitarian* or *sex-equal* stable matching of SM of size n. Local search algorithms are among the popular methods for solving optimization problems because of two key advantages: (i) they take very little memory; and (ii) they find quickly reasonable solutions in large or infinite state spaces. However, the local search algorithms often fail to find a global optimal solution when one exists because they can get stuck on a local optimum solution.

Our algorithm is shown in Algorithm 1. The basic idea is derived from the Theorem 1, that is, the algorithm finds the best solution while moving the current solution from the *man-optimal* towards the *woman-optimal* one. Procedure GALE-SHAPLEY (line 1) finds the *man-optimal* stable matching using Gale-Shapley algorithm. For each search step, procedure BREAK-MARRIAGE (line 6) discovers a neighbor set of the current solution for every man. The next solution is selected to be the best one among of the neighbor set by means of a cost function $f(M)$, which is *egalitarian cost* (respectively *sex-equal cost*) for finding the *egalitarian* (respectively *sex-equal*) stable matching. The next solution can also be selected randomly in the neighbor set to overcome the stuck on a local optimum. It differs from other local search algorithms such as hill climbing and simulated annealing search [13], our algorithm ends if no neighbors exits, meaning that the algorithm reaches to the *woman-optimal* stable matching. This makes the algorithm increase the run time but it can potentially achieve a global optimum solution.

Algorithm 1. Local Search Algorithm

Input : an instance of SM
Output: a stable matching

1: $M_{current} \leftarrow$ GALE-SHAPLEY(an instance of SM);
2: $M_{best} \leftarrow M_{current}$;
3: **while** *(true)* **do**
4:     *neighbors* $\leftarrow \emptyset$;
5:     **for** *(each man m in men)* **do**
6:         *matching* $\leftarrow$ BREAK-MARRIAGE($M_{current}$,m);
7:         add *matching* to *neighbors*;
8:     **end**
9:     **if** *(no neighbors are found)* **then**
10:         break;
11:     **end**
12:     **if** *(small random probability p)* **then**
13:         $M_{next} \leftarrow$ a randomly selected matching in *neighbors*;
14:     **else**
15:         $M_{next} \leftarrow \underset{M \in neighbors}{\arg\min}\ (f(M))$;
16:     **end**
17:     **if** *($f(M_{best}) > f(M_{next})$)* **then**
18:         $M_{best} \leftarrow M_{next}$;
19:     **end**
20:     $M_{current} \leftarrow M_{next}$;
21: **end**
22: **return** M_{best};

An illustration of Algorithm 1 to find a *sex-equal* stable matching for the SM in Table 1 is depicted in Fig. 1. The probability to move the solution to a random neighbor is set to be zero. Initially, the algorithm assigns the current solution and the best solution to the *man-optimal* stable matching M_0. The algorithm then finds the stable neighbor matchings of M_0, which are M_1 and M_2. M_2 is selected to be

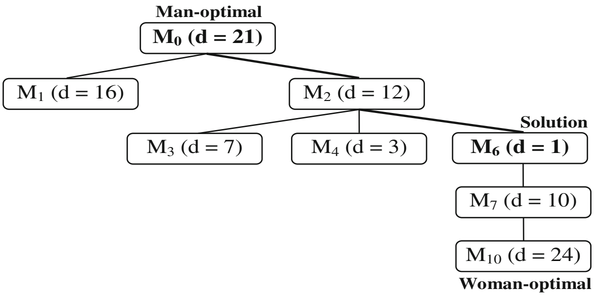

Fig. 1. The tree search of finding the *sex-equal* stable matching of Example 1.

the next solution, since $d(M_1) > d(M_2)$. Because $d(M_0) > d(M_2)$, the best solution is M_2. The algorithm repeats for M_2 to obtain M_6 as the current solution. M_7 and then M_{10} are obtained by the same operation. Since no stable neighbor matchings of M_{10} are found, the algorithm ends and outputs the best solution to be M_6 because M_6 is the best solution found so far. As shown in Table 2, the algorithm finds exactly the *sex-equal* stable matching M_6. Apparently, if we assume that $d(M_5) < d(M_6)$, then the algorithm gives a local optimum M_6 because it does not reach to M_5, M_8 and M_9. However, according to Corollary 1, M_5 must be generated from M_1, M_2, M_3 or M_4 and therefore, if the algorithm selects the next solution to be a random neighbor with a certain probability, then the matching M_1, M_2, M_3 or M_4 can be selected to generate M_5.

4 Simulations

This section presents simulations implemented with the Matlab software on a Core i5-2430M CPU 2.4 GHz with 4 GB RAM computer. The simulations are designed to evaluate the performance of our algorithm for SM instances, in which the preference lists of men and women are generated randomly. The probability of choosing a random stable matching in stable neighbor matchings is set $p = 0.1$.

First, we make simulations in order to compare the solutions found by the proposed algorithm with those found by the brute-force algorithm. The brute-force algorithm is typically used when the SM size is small and therefore, the size of SM instances is set $n = 10$. Table 3 shows the results of 30 simulations on 30 SM instances. The simulation results show that the *egalitarian cost* and the *sex-equal cost* by our algorithm (LS algorithm) is the same as that found by the brute-force algorithm (BF algorithm).

Second, we make simulations on SM instances of large sizes to evaluate the correctness and the run time of our algorithm. We generate randomly 12 SM instances of size n. For each SM instance, we run 10 times and calculate the mean and the standard deviation of the *egalitarian cost* (respectively *sex-equal cost*) for finding an *egalitarian* (respectively *sex-equal*) stable matching. Table 4 shows the simulation results. For each SM instance, the standard deviation value is zero

Table 3. The simulation results of the brute-force algorithm and our algorithm.

Data set	BF algorithm		LS algorithm		Data Set	BF algorithm		LS algorithm	
	$min(c)$	$min(d)$	$min(c)$	$min(d)$		$min(c)$	$min(d)$	$min(c)$	$min(d)$
1	56	6	56	6	16	51	3	51	3
2	61	7	61	7	17	53	5	53	5
3	63	4	63	4	18	51	13	51	13
4	62	0	62	0	19	58	0	58	0
5	80	12	80	12	20	61	1	61	1
6	71	21	71	21	21	58	0	58	0
7	59	9	59	9	22	59	7	59	7
8	60	6	60	6	23	50	4	50	4
9	53	5	53	5	24	60	8	60	8
10	58	6	58	6	25	47	3	47	3
11	56	6	56	6	26	52	2	52	2
12	55	2	55	2	27	60	2	60	2
13	58	20	58	20	28	64	1	64	1
14	53	9	53	9	29	62	0	62	0
15	60	10	60	10	30	59	14	59	14

Table 4. The simulation results of our algorithm for SM instances of large sizes.

Data set	Size	*egalitarian cost*			*sex-equal cost*		
		Mean	Deviation	Time (sec.)	Mean	Deviation	Time (sec.)
1	n = 50	678	0	0.661	45	0	0.593
2	n = 75	1214	0	3.128	31	0	3.564
3	n = 100	2050	0	6.764	107	0	6.964
4	n = 125	2748	0	10.095	3	0	15.351
5	n = 150	3630	0	16.589	63	0	18.154
6	n = 175	4514	0	54.497	19	0	48.670
7	n = 200	5348	0	67.438	17	0	77.152
8	n = 225	6413	0	77.319	31	0	88.202
9	n = 250	7847	0	188.323	173	0	238.256
10	n = 300	10094	0	232.674	106	0	220.490
11	n = 350	12565	0	572.411	363	0	578.730
12	n = 400	15607	0	676.545	23	0	680.780

and the mean value is the *egalitarian cost* (respectively *sex-equal cost*) of the *egalitarian* (respectively *sex-equal*) stable matching. Although the probability of choosing a random stable matching in stable neighbor matchings is set $p = 0.1$, the standard deviation of *egalitarian cost* (respectively *sex-equal cost*) in simulations on each SM instance is zero, meaning that our algorithm finds exactly the *egalitarian* (respectively the *sex-equal*) stable matching. Furthermore, the average time shows that our algorithm is efficient in terms of computational time even the size of SM instances is large.

Finally, we make simulations on SM instances to compare the run time of our algorithm with that of ACS algorithm [14]. The simulations show that the ACS algorithm finds an *egalitarian* or *sex-equal* stable matching only for SM instances of small sizes ($n \leq 30$) since the ACS algorithm has to find a large amount of pairs (man,woman) to form a stable matching. For example, given an instance of SM of size $n = 100$, the ACS algorithm has to find $n^2 = 10000$ pairs (man,woman) to form a stable matching of 100 engaged pairs. Moreover, simulations show that our algorithm finds the solution of SM instances more faster than the ACS algorithm does.

5 Conclusions

In this paper, we proposed a local search algorithm to find an *egalitarian* or *sex-equal* stable matching of SM instances by using the Gale-Shapley algorithm, the breakmarriage operation, and a random walk. The simulation results show that the proposed algorithm is efficient for solving SM instances. Moreover, the proposed algorithm is a general approach since with minor modifications of the cost function it can find other optimal stable matchings.

Acknowledgements. The authors are grateful to the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2014R1A1A2057735), IITP (2015-(R0134-15-1033)) and the Kyung Hee University in 2013 [KHU-20130439].

References

1. Abraham, D.J., Irving, R.W., Manlove, D.F.: The student-project allocation problem. In: Proceedings of the 14th International Symposium, pp. 474–484. Kyoto, Japan, December 2003
2. Fleiner, T., Irving, R.W., Manlove, D.F.: Efficient algorithms for generalized stable marriage and roommates problems. Theor. Comput. Sci. **381**(1–3), 162–176 (2007)
3. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am. Math. Mon. **9**(1), 9–15 (1962)
4. Gelain, M., Pini, M.S., Rossi, F., Venable, K.B., Walsh, T.: Local search approaches in stable matching problems. Algorithms **6**(1), 591–617 (2013)
5. Gusfield, D.: Three fast algorithms for four problems in stable marriage. SIAM J. Comput. **16**(1), 111–128 (1987)
6. Gusfield, D., Irving, R.W.: The Stable Marriage Problem: Structure and Algorithms. MIT Press, Cambridge (1989)
7. Irving, R.W.: An efficient algorithm for the “stable roommates” problem. J. Algorithms **6**(1), 577–595 (1985)
8. Irving, R.W., Leather, P.: The complexity of counting stable marriages. SIAM J. Comput. **15**(3), 655–667 (1986)

9. Iwama, K., Miyazaki, S., Yanagisawa, H.: Approximation algorithms for the sex-equal stable marriage problem. ACM Trans. Algorithms **7**(1), 2:1–2:17 (2010)
10. McVitie, D.G., Wilson, L.B.: The stable marriage problem. Commun. ACM **14**(7), 486–490 (1971)
11. Nakamura, M., Onaga, K., Kyan, S., Silva, M.: Genetic algorithm for sex-fairstable marriage problem. In: 1995 IEEE International Symposium on Circuits and Systems, ISCAS 1995, vol. 1, pp. 509–512. Seattle, WA, April 1995
12. Roth, A.E.: The evolution of the labor market for medical interns and residents: a case study in game theory. J. Polit. Econ. **92**(6), 991–1016 (1984)
13. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Pearson Education, Upper Saddle River (2010)
14. Vien, N.A., Viet, N.H., Kim, H., Lee, S., Chung, T.: Ant colony based algorithm for stable marriage problem. Adv. Innov. Syst. Comput. Sci. Softw. Eng. **1**, 457–461 (2007)