



# A Max-Min Conflict Algorithm for the Stable Marriage Problem

Hoang Huu Viet<sup>1</sup>, Nguyen Thi Uyen<sup>1</sup>, Pham Tra My<sup>1</sup>, Son Thanh Cao<sup>1</sup>,  
and Le Hong Trang<sup>2</sup>(✉)

<sup>1</sup> School of Engineering and Technology, Vinh University, Vinh City, Vietnam  
{viethh, uyennt, mypt, sonct}@vinhuni.edu.vn

<sup>2</sup> Faculty of Computer Science and Engineering,  
Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam  
lhtrang@hcmut.edu.vn

**Abstract.** In this paper we present a max-min conflict algorithm to find a stable matching rather than the man- and woman-optimal matchings for the stable marriage problem. We solve the problem in terms of a constraint satisfaction problem, i.e. find a complete assignment for men in which every man is assigned to a woman so that the assignment does not contain any blocking pairs. To do this, we apply a local search method in which a max-conflict heuristic is used to choose the man making the maximum number of blocking pairs in a matching, while a min-conflict heuristic is used to remove all the blocking pairs formed by the chosen man. Experiments showed that our algorithm is efficient for finding a stable matching of large stable marriage problems.

**Keywords:** Constraint satisfaction problem · Local search ·  
Max-min-conflicts · Stable marriage problem

## 1 Introduction

The stable marriage (SM) problem, introduced by Gale and Shapley [6] in 1962, is a well-known problem of matching an equal number men and women to satisfy a certain criterion of stability. Recently, this problem has been attracting much 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 [15], the Student-Project Allocation problem (SPA)[1] and the Stable Roommates problem (SR) [5, 11].

An SM instance of size  $n$  consists of a set of  $n$  men and a set of  $n$  women in which each person ranks all members of the other set in a strict order of preference. A matching  $M$  is a set of  $n$  disjoint pairs of men and women. If a man  $m$  and a woman  $w$  are a pair  $(m, w) \in M$ , we say that  $m$  and  $w$  are partners in  $M$ , denoted by  $m = M(w)$  and  $w = M(m)$ . If  $m$  prefers  $w$  to  $M(m)$  and  $w$  prefers  $m$  to  $M(w)$ , then we say that  $m$  and  $w$  form a *blocking pair* in  $M$ . A

matching  $M$  is said to be *stable* if it has no any blocking pairs, otherwise it is said to be *unstable*.

Gale and Shapley showed that there exists at least one stable matching for every instance of SM and proposed an algorithm, called Gale-Shapley algorithm, to find a stable matching of SM instances of size  $n$  in time  $O(n^2)$  [6]. This algorithm is a sequence of proposals either from the men to the women or from the women to the men. In the former case, it finds the so-called *man-optimal* stable matching, in which each man has the best partner, but each woman has the worst partner compared to their partners in any other stable matchings. In the latter case, it finds the *woman-optimal* stable matching, in which the men's and women's properties are interchanged. For some SM instances, there may be many other stable matchings between the *man-* and *woman-optimal* stable matchings. Moreover, the *man-* (respectively *woman-*) *optimal* stable matching is the “*self-ish*” matching for the men (respectively women), i.e. the proposers always get their best partners but the responders get their worst partners. Therefore, it is appropriate to seek other optimal stable matchings to give a more balanced preference for both the men and women.

In this paper, we present a max-min conflict algorithm to seek a stable matching rather than the man- and woman-optimal matchings of SM instances. The algorithm starts from a random matching and tries to search a better one by mean of reducing the number of blocking pairs. At each search step, the algorithm chooses a man making the maximum number of blocking pairs and a woman in blocking pairs formed by the man such that her rank is minimum in the chosen man's preference list. To avoid getting stuck in a local optimum, the algorithm also chooses a random woman in blocking pairs formed by the chosen man in a small probability. Then, the algorithm removes the blocking pair formed by the man and the woman in the matching and repeats for the matching until the matching is stable. The experiments show that our approach is efficient in terms of computational time for large SM problems.

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

## 2 Related Work

In the last few years, there were several approaches to find a stable matching rather than the man- and woman- optimal matchings. Nakamura et al. proposed a genetic algorithm (GA) for finding the *sex-fair* matching of SM instances [14]. In their approach, the problem is first transferred into a directed graph and the GA is used to find the solution in the graph. Vien et al. applied the ant colony system (ACS) algorithm [3] for finding the *man-optimal*, *woman-optimal*, *egalitarian* and *sex-equal* stable matchings of SM instances [16]. Unfortunately, the ACS algorithm finds the matching under a given criterion only for small SM instances because it has to find  $n^2$  pairs (man, woman) to form a stable matching. Zavidovique et al. [18] presented three zigzag algorithms, named ZZ,

OZ and BZ, to find matchings that meet three criteria of stability, sex equality and egalitarian. In  $O(n^2)$  time, the ZZ algorithm finds the egalitarian, while the OZ algorithm finds both the egalitarian and sex equality, but they are not guaranteed that they will find stable matchings. In  $O(n^3)$  time, the BZ algorithm is designed to meet all three criteria rather than the egalitarian or sex-equal matching. Everaere et al. [4] proposed a Swing algorithm for finding the sex-equal matching of SM instances, in which the men and women alternatively play the role of proposers and responders in iterations. When the Swing stops, it takes  $O(n^3)$  time to obtain a stable matching other than the sex-equal matching. Giannakopoulos et al. [10] provided an ESMA algorithm [10] which the idea is similar to that of Swing. However, in the ESMA the proposers are men when the sign of the function  $\sin(k^2)$  is positive and women when the sign of the function is negative, where  $k$  is the iteration counter of the algorithm.

Recently, a local search approach has been applied to deal with the SM problem. Gelain et al. [7] proposed three local search algorithms, named SML, SML1 and SML2, for finding an arbitrary stable matching of SM instances. Starting at a randomly generated matching,  $M$ , these algorithms produce a set of the neighbor matchings of  $M$ , where a neighbor is determined by removing one of the blocking pairs in  $M$ , and moves  $M$  to the neighbor matching which has the smallest number of blocking pairs. This process iteratively performs until the stability in  $M$  is obtained. Viet et al. [17] developed an empirical local search algorithm for finding an approximation solution in terms of the egalitarian or sex-equal matching. This algorithm uses the breakmarriage operation [12] to find all the stable neighbor matchings of the current stable matching and then move the current matching to the best matching in the stable neighbor matchings. Besides, Gent et al. [8] proposed two algorithms to formulate an SM instance to a constraint satisfaction problem. Codognet and Diaz proposed a generic, domain-independent local search method, called adaptive search (AS), for solving constraint satisfaction problems [2]. Accordingly, Gent et al. [9] and Munera et al. [13] proposed local search algorithms to deal with the stable marriage problem with ties and incomplete lists, a variant of the SM problem, based on constraint satisfaction problems.

### 3 Max-Min Conflict Algorithm

In this section, we propose a max-min conflict algorithm, so called MMC, to find a stable matching for the SM problem. Our approach is based on a local search method for solving a constraint satisfaction problem, in which men are considered as variables, women are considered as the domain of each variable and constraints are that there exists no blocking pair in matchings. According to this approach, a stable matching is a complete assignment in which every variable is assigned to a value so that the matching does not violate constraints. To find a stable matching, our search strategy is that at each search step we choose the most constrained variable, i.e. a man making the maximum number of blocking pairs, and select the least constrained value, i.e. a woman having the

---

**Algorithm 1.** *Max-Min Conflict Algorithm*


---

**Input** : An SM instance of size  $n$ ,  
 $max\_iters$ , the maximum number of iterations.

**Output:**  $M$ , a stable matching.

1.  $M :=$  a randomly generated matching;
  2.  $iter := 1$ ;
  3. **while** ( $iter \leq max\_iters$ ) **do**
  4.     **if** (*there exists no blocking pair in  $M$* ) **then**
  5.         | break;
  6.     **end**
  7.      $X := \{(m, w) \mid (m, w) \text{ is a blocking pair in } M\}$ ;
  8.      $error(m) :=$  counting the number of man  $m$  appearing in  $X$ ;
  9.      $Y := \{(m', w') \in X \mid m' = \operatorname{argmax}(error(i)), i = 1, 2, \dots, n\}$ ;
  10.    **if** (*a small probability of  $p$* ) **then**
  11.         |  $w'' :=$  a random woman  $w' \in Y$ ;
  12.    **else**
  13.         |  $w'' := \operatorname{argmin}(mr(m', w'))$ , where  $(m', w') \in Y$ ;
  14.    **end**
  15.     $M :=$  removing the blocking pair  $(m', w'') \in M$ ;
  16.     $iter = iter + 1$ ;
  17. **end**
  18. **return**  $M$ ;
- 

minimum rank in the chosen man's preference list. This is because that choosing the most constrained variable is able to remove the largest number of blocking pairs, while selecting the least constrained value of the chosen variable removes all the blocking pairs formed by the chosen man and therefore, our algorithm accelerates the search of a stable matching.

We let  $mr(m, w)$  denote the rank of a woman,  $w$ , in a man  $m$ 's preference list. For a man,  $m$ , we define an error function,  $error(m)$ , to be the number of blocking pairs formed by  $m$  in a matching  $M$ . Our MMC algorithm is shown in Algorithm 1. Starting from a randomly generated matching,  $M$ , the algorithm performs as follows. First, the algorithm checks if there exists no blocking pair in  $M$ , meaning that  $M$  is stable, then it returns the matching  $M$ . Otherwise, the algorithm determines all the blocking pairs  $(m, w) \in M$  (line 7) and finds a man,  $m'$ , making the maximum number of blocking pairs in  $M$ , i.e., the man has the maximum number of conflicts in terms of blocking pairs (line 9). If a small probability of  $p$  is accepted, the algorithm chooses a random woman,  $w''$ , who is making a blocking pair with  $m'$ . Otherwise, the algorithm chooses a woman,  $w''$ , to whom man  $m'$  most prefers in the set of the women making blocking pairs with  $m'$ . Then, the algorithm removes the blocking pair  $(m', w'')$  by mean of replacing two pairs of  $(m', M(m'))$  and  $(M(w''), w'')$  by two pairs of  $(m', w'')$  and  $(M(w''), M(m'))$  in  $M$ . The algorithm repeats until either the matching  $M$  has no blocking pairs or a maximum number of iterations is reached.

**Table 1.** Preference lists of eight men and eight women

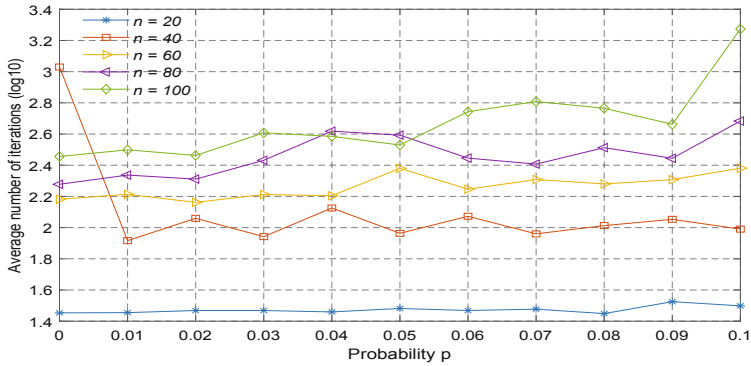
Man	Preference list	Woman	Preference list
$m_1$	4 7 3 8 1 5 2 6	$w_1$	1 3 5 4 2 6 8 7
$m_2$	5 3 4 2 1 8 6 7	$w_2$	8 2 4 5 3 7 1 6
$m_3$	3 8 2 4 6 7 5 1	$w_3$	5 8 1 4 2 3 6 7
$m_4$	5 6 8 3 4 7 1 2	$w_4$	2 4 3 6 5 8 1 7
$m_5$	1 3 5 2 8 6 4 7	$w_5$	6 5 4 8 1 7 2 3
$m_6$	8 6 2 5 1 7 4 3	$w_6$	7 4 2 5 6 8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

Considering an SM instance consists of eight men and eight women with their preference lists given in Table 1. Suppose that given a probability  $p = 0$  and a random matching  $M = \{(1,3), (2,1), (3,2), (4,8), (5,7), (6,4), (7,5), (8,6)\}$ , the algorithm runs as follows. Because the matching  $M$  is unstable, the algorithm continues to find all the blocking pairs in  $M$ :  $X = \{(2,2), (2,4), (4,5), (4,6), (5,1), (5,2), (5,3), (5,5), (5,6), (6,5), (6,6), (6,7), (8,5), (8,7)\}$  (the number of blocking pairs in  $M$  is 14). Then, it counts the number of blocking pairs in  $M$ , i.e.  $error(m)$ , formed by man  $m$ , that is  $error(2) = 2$ ,  $error(4) = 2$ ,  $error(5) = 5$ ,  $error(6) = 3$ ,  $error(8) = 2$ . Next, the algorithm takes the set of blocking pairs  $Y = \{(5,1), (5,2), (5,3), (5,5), (5,6)\}$  corresponding to the man,  $m = 5$ , who is making the maximum number of blocking pairs. Because man  $m = 5$  most prefers woman  $w = 1$  to the other women in his blocking pairs, the algorithm removes the blocking pair  $(5,1)$  in  $M$  by replacing two pairs of  $(5,7)$  and  $(2,1)$  by two pairs of  $(5,1)$  and  $(2,7)$  in  $M$  to obtain the matching  $M = \{(1,3), (2,7), (3,2), (4,8), (5,1), (6,4), (7,5), (8,6)\}$ . Because the number of blocking pairs in  $M$  is 10, i.e. not zero, the algorithm continues repeating until the stability of  $M$  is obtained. Specifically, after four iterations, the algorithm terminates and gives a matching  $M = \{(1,3), (2,4), (3,2), (4,5), (5,1), (6,6), (7,8), (8,7)\}$ , which is a stable matching.

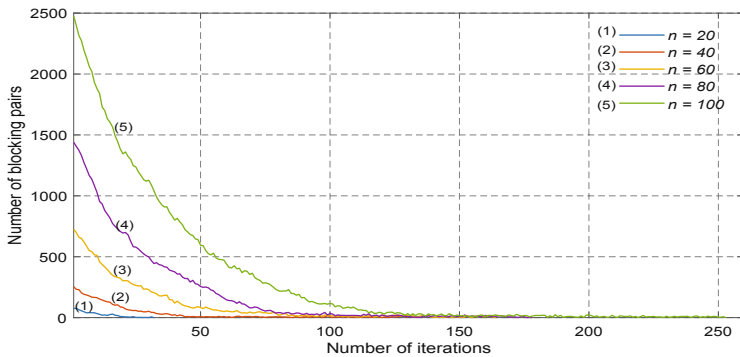
## 4 Experiments

In this section we presents the experiments implemented by Matlab software on a Core i7-8550U CPU 1.8 GHz computer with 16 GB RAM. To evaluate the performance of our algorithm, we randomly produced 100 SM instances of 10 difference sizes from 20 to 200 with step 20 and 10 variants per size.

First, we ran experiments to determine the best value of probability  $p$  for the SM instances of size  $n$ , where  $n = 20, 40, 60, 80$  and 100. Figure 1 shows our experimental results. For the SM instances of size 40, the average number of iterations to find stable matchings is highest when  $p = 0$ . For the SM instances of the other sizes, the average number of iterations to find stable matchings when



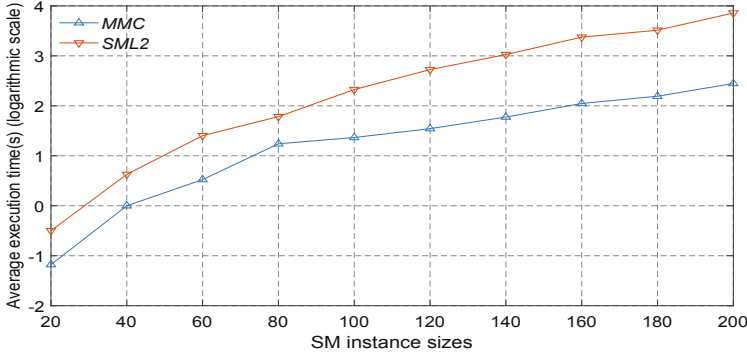
**Fig. 1.** The average number of iterations found by MMC algorithm with probability  $p$



**Fig. 2.** The number of blocking pairs found in iterations of MMC algorithm

$p = 0$  is almost the same as that when  $p = 0.02$ . Therefore, we choose  $p = 0.02$  to be the best value in our experiments.

Second, we performed experiments to consider the behavior of the max-min conflict heuristics. To do this, we considered the variation of the number blocking pairs of matchings in iterations of MMC algorithm. Figure 2 shows experimental results for the sizes of SM instances consisting of  $n = 20, 40, 60, 80$  and  $100$ . Specifically, to find a stable matching, MMC needs 32 iterations for  $n = 20$ , 132 iterations for  $n = 40$ , and 252 iterations for  $n = 100$ . It is easy to see that for an SM instance of size  $n$ , the number of pairs (man, woman) is  $n^2$ , therefore when the sizes of SM instances increases from 20 to 100, the number of pairs (man, woman) increases from 400 to 10000. However, for MMC, the number of iterations increases only from 32 to 252, i.e. MMC is efficient for large SM instances. Figure 2 also shows that the number of blocking pairs found by MMC algorithm does not monotonically decrease, i.e. there exist local minimum values. This is because when removing a blocking pair in a matching to obtain a new

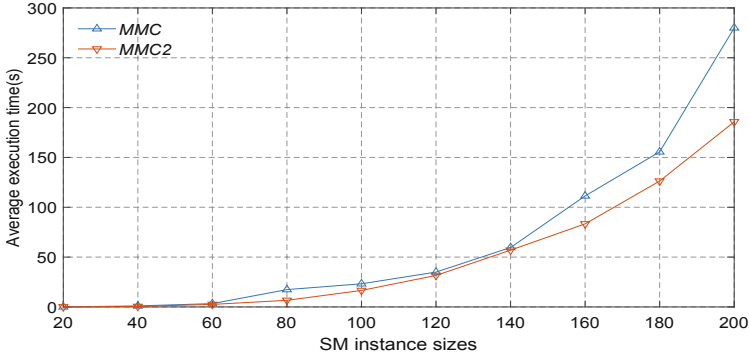


**Fig. 3.** The average execution time (sec.) of MMC algorithm

matching, the new matching will generate the new blocking pairs that it is not guaranteed that the number of blocking pairs of the new matching is smaller than that of the old one. However, even when the probability  $p = 0$  as shown in Fig. 1, our MMC algorithm also overcomes this weak point to find a stable matching after a finite number of iterations.

Next, we performed experiments to compare the execution time of MMC algorithm with that of the SML2 algorithm [7]. Figure 3 shows the experimental results. When the size of SM instances is 20, the execution time found by the MMC and SML2 algorithms is about  $10^{-1.15} \approx 0.07$  and  $10^{-0.5} \approx 0.31$  s, respectively, i.e. our MMC runs about 4.5 times faster than SML2 algorithm. When the size of SM instances is 100, the execution time found by MMC and SML2 algorithm is about  $10^{1.35} \approx 22.38$  and  $10^{2.30} \approx 199.52$  s, respectively, i.e. our MMC runs about 9 times faster than SML2 algorithm. Especially, when the size of SM instances is 200, the execution time found by our MMC is about  $10^{2.42} \approx 263.02$  s, but the execution time found by SML2 algorithm is about  $10^{3.9} \approx 7943.30$  s, i.e. our MMC runs about 30 times faster than SML2 algorithm. This means that the larger SM instances is, the faster MMC runs compared to SML2 algorithm. The experimental results are explained as follows. At each iteration, SML2 keeps at least  $n$  neighbor matchings and for each neighbor matching, it has to take  $O(n^2)$  time to determine the number of blocking pairs. This means that at each iteration, SML2 has to take  $O(n^3)$  time to find the best matching in the neighbor matchings. Meanwhile, our MMC algorithm keeps only one neighbor matching to determine the number of blocking pairs, i.e. MMC needs only  $O(n^2)$  time for each iteration and therefore, our MMC algorithm runs much faster than SML2 algorithm, especially for large SM problems.

It should be noted that in the MMC algorithm, the roles of men and women can be interchanged at iterations as in [4, 10]. This means that the max-min conflict heuristics are applied for the men at the odd iterations, but applied for the women at the even iterations of the algorithm. However, we think that it is better if we improve MMC algorithm as follows: At each iteration, after we



**Fig. 4.** The average execution time (sec.) of MMC and MMC2 algorithms

found all the blocking pairs in  $M$ , we find both a man  $m'$  and a woman  $w'$  who are having the maximum number of conflicts by mean of blocking pairs. Then, we apply the max conflict heuristic for only either  $m'$  or  $w'$  who is making the maximum number of blocking pairs in  $M$ . Next, we apply the min conflict heuristic for the chosen person. We call the improved algorithm the MMC2 algorithm. Figure 4 shows experimental results to compare the runtime of MMC algorithm with that of MMC2 algorithm. Obviously, MMC2 algorithm is an efficient variant of MMC algorithm, especially for SM instances of large sizes.

## 5 Conclusions

This paper proposed a max-min-conflict algorithm to find a stable matching which is different from the man- and woman-optimal matchings of the SM problem. Starting a randomly generated matching, the algorithm finds a man making the *maximum* number of blocking pairs. Once a man has been found, the algorithm chooses a woman in blocking pairs formed by the man such that her rank is *minimum* in the man's preference list. The algorithm then removes the blocking pair formed by the man and the woman in the matching and repeats for the matching until it is stable. Experiments showed that our MMC algorithm outperforms the SML2 algorithm in terms of execution time for finding a stable matching of large stable marriage problems. In the future, we plan to extend the proposed approach to the stable marriage problem with ties and incomplete lists [7, 13].

**Acknowledgment.** This research is funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under **grant number 102.01-2017.09**.



## References

1. Abraham, D.J., Irving, R.W., Manlove, D.F.: The student-project allocation problem. In: Proceedings of the 14th International Symposium, Kyoto, Japan, pp. 474–484, December 2003
2. Codognet, P., Diaz, D.: Yet another local search method for constraint solving. In: Proceedings of the International Symposium on Stochastic Algorithms, Berlin, Germany, pp. 73–90, December 2001
3. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**(1), 53–66 (1997)
4. Everaere, P., Morge, M., Picard, G.: Minimal concession strategy for reaching fair, optimal and stable marriages. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), St. Paul, MN, USA, pp. 1319–1320, May 2013
5. 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)
6. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. *Am. Math. Monthly* **9**(1), 9–15 (1962)
7. 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)
8. Gent, I.P., Irving, R.W., Manlove, D., Prosser, P., Smith, B.M.: A constraint programming approach to the stable marriage problem. In: Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming, Berlin, Heidelberg, vol. 1, pp. 225–239, December 2001
9. Gent, I.P., Prosser, P.: An empirical study of the stable marriage problem with ties and incomplete lists. In: Proceedings of the 15th European Conference on Artificial Intelligence, Lyon, France, pp. 141–145, July 2002
10. Giannakopoulos, I., Karras, P., Tsoumakos, D., Doka, K., Koziris, N.: An equitable solution to the stable marriage problem. In: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), Vietri sul Mare, Italy, pp. 989–996, November 2015
11. Irving, R.W.: An efficient algorithm for the “stable roommates” problem. *J. Algorithms* **6**(1), 577–595 (1985)
12. McVitie, D.G., Wilson, L.B.: The stable marriage problem. *Commun. ACM* **14**(7), 486–490 (1971)
13. Munera, D., Diaz, D., Abreu, S., Rossi, F., Saraswat, V., Codognet, P.: Solving hard stable matching problems via local search and cooperative parallelization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 1212–1218 (2015)
14. 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), Seattle, WA, pp. 509–512, April 1995
15. 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)

16. Vien, N.A., Viet, N.H., Kim, H., Lee, S., Chung, T.: Ant colony based algorithm for stable marriage problem. In: *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pp. 457–461 (2007)
17. Viet, H.H., Trang, L.H., Lee, S.G., Chung, T.C.: An empirical local search for the stable marriage problem. In: Booth, R., Zhang, M.-L. (eds.) *PRICAI 2016. LNCS (LNAI)*, vol. 9810, pp. 556–564. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-42911-3\\_46](https://doi.org/10.1007/978-3-319-42911-3_46)
18. Zavidovique, B., Suvonvorn, N., Seetharaman, G.: A novel representation and algorithms for (quasi) stable marriages. In: *2005 Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Barcelona, Spain, pp. 63–70, September 2005