A Searching for Strongly Egalitarian and Sex-Equal Stable Matchings

Le Hong Trang[1(✉)], Hoang Huu Viet[2], Tran Van Hoai[1], and Tran Xuan Hao[2]

[1] Ho Chi Minh City University of Technology, VNU-HCM, 268 Ly Thuong Kiet, Ho Chi Minh City, Vietnam
lhtrang@hcmut.edu.vn

[2] Vinh University, 182 Le Duan, Vinh City, Nghe An Province, Vietnam

Abstract. The stable marriage problem with ties (SMT) is a variant of the stable marriage problem in which people are permitted to express ties in their preference lists. In this paper, an algorithm based on bidirectional searching is presented for trying to find *strongly egalitarian* and *sex-equal* stable matchings. We indicate that the use of two simultaneous searches in the algorithm not only accelerate the finding of solutions but also is appropriate for the strong stability criterion of SMT. The algorithm is implemented and tested for large datasets. Experimental results show that our algorithm is significant.

Keywords: Strongly stable · Egalitarian · Sex-equal · Ties · Bidirectional search

1 Introduction

An instance I of the classical stable marriage problem (SMP) of size n involves n men and n women. Each man ranks n women to give himself a preference list, and similarly each woman ranks n man to also give herself a preference list. A matching M in I is an one-one correspondence between the men and women of I. For a pair of man and woman $(m, w) \in M$, we denote by $M(m)$ and $M(w)$ the partner of m and w in M, respectively, i.e., $w = M(m)$ and $m = M(w)$. A pair (m, w) is said to be blocking pair for M, if m and w are not partners and m (w, respectively) prefers w (m, respectively) to $M(m)$ ($M(w)$, respectively). A matching that admits no blocking pair is said to be stable, otherwise it is unstable. Let us denote by $p_m(w)$ ($p_w(m)$, respectively) the position of w (m, respectively) in m's (w's, respectively) preference list. It was shown by Gale and Shapley that every instance of SMP admits at least a stable matching [1], and the matching can be found in $O(n^2)$. A stable matching found by Gale and Shapley's algorithm is man- or woman-optimal.

© Springer Nature Switzerland AG 2019
S. Lee et al. (Eds.): IMCOM 2019, AISC 935, pp. 1100–1111, 2019.
https://doi.org/10.1007/978-3-030-19063-7_87

For a stable matching M, we define the *man cost*, denoted by $sm(M)$, and the *woman cost*, denoted by $sw(M)$, as follows:

$$sm(M) = \sum_{(m,w)\in M} p_m(w),$$
$$sw(M) = \sum_{(m,w)\in M} p_w(m).$$

We also define the egalitarian and sex-equal costs by:

$$c(M) = sm(M) + sw(M), \tag{1}$$
$$d(M) = |sm(M) - sw(M)|. \tag{2}$$

Let $\mathcal{M}$ be the set of all stable matchings of an instance I of the SMP. For $M \in \mathcal{M}$, M is called to be *egalitarian* (*sex-equal*, respectively) if $c(M)$ ($d(M)$, respectively) is minimum among all stable matchings in $\mathcal{M}$ [14]. The fairness is emphasized for the egalitarian and sex-equal matchings in which we attempt to obtain the balance of preferences between men and women in a stable matching. Several approaches were proposed for finding egalitarian and sex-equal matching such as genetic algorithm [14], ant colony system [15], and approximation algorithm [9].

A generalization of SMP called the stable marriage problem with ties (SMT) arises when people are permitted to express ties in their preference lists. Particularly, each person does not need to rank members of the opposite sex in *strict order*. Some of those involved might be indifference among members of the opposite sex. When once the ties is allowed in the preference lists, stability of a matching can be defined in three possible forms [6]. In particular, a matching M is *weakly stable* if there is no couple (x, y), each of whom strictly prefers the other to his/her partner in M. Also, a matching M is *strongly stable* if there is no couple (x, y) such that x strictly prefers y to his/her partner in M, and y either strictly prefers x to his/her partner in M or is indifferent between them. Finally, a matching M is *super-stable* if there is no couple (x, y), each of whom either strictly prefers the other to his/her partner in M or is indifferent between them. We note that a person p *strictly prefers* a person q in a preference list, p precedes q in the list.

The SMT problem has been received many attentions of researchers (examples can be found in [3,5,6,8]). In order to deal with the problem of large size, a useful approach based on local searching was studied. In 2013, Gelain et al. proposed a local search method to speed up the process of finding solutions [2]. Munera et al. have also addressed the problem using local search approach, based on Adaptive Search [13]. However, these methods only find a stable matching of a given instance.

In this paper, we present a local search algorithm to address SMT for strongly egalitarian or sex-equal stable matchings. Based on the distributive lattice structure of strongly stable matchings in an SMT instance, a search scheme consisting of two simultaneous local searches is performed in both man and woman points of view. In each locally searching step, an adapted breakmarriage operation is used to determine the neighbor set of a strongly stable matching. By the dominance relation which is maintained by the breakmarriage operation, a stop condition is established to terminate the algorithm. The algorithm is implemented and tested for large datasets. Experimental results are given to show the time performance of our algorithm.

The rest of the paper is organized as follows. Section 2 recalls the distributive lattice structure formed by the set of strongly stable matchings, which is the essential structure for establishing the searching scheme in our algorithm. The proposed algorithm is given in Sect. 3. Section 4 is devoted to implementation and testing for large datasets. Finally, some concluding remarks are given in Sect. 5.

2 Preliminaries

The lattice structure of stable matchings plays an important role for solving some problems associated with SMP, such as finding all stable pairs, all stable matchings [4], and also an egalitarian stable matching [7] of a given instance of SMP. Utilizing the dominance properties in the lattice structure, a number of local search algorithms have been proposed for finding optimally stable matchings, i.e. sex-equal and egalitarian, of SMPs [16,17].

Since the preference condition is not symmetric in the definition of strongly stability, the distributive lattice of strongly stable matchings is thus not obviously derived. Fortunately, Manlove in [11] indicated that under a equivalence relation defined on the set of strongly stable matchings for a given SMT instance, the set of equivalence classes forms a distributive lattice under a dominance relation.

Let M and M' be matchings of a given SMT instance I, and q be a person in I. We say that q strictly prefers M to M' (is indifferent between M and M', respectively) if q strictly prefers $p_M(q)$ to $p_{M'}(q)$ (is indifferent between $p_M(q)$ to $p_{M'}(q)$, respectively).

Definition 1 (Equivalence relation [11]). *Given a SMT instance I, let $\mathcal{M}$ be the set of strongly stable matchings in I. An equivalence relation, denoted by $\sim$, is defined on $\mathcal{M}$ as follows: for any $M, M' \in \mathcal{M}$, $M \sim M'$ if and only if each man is indifferent between M and M'.*

A dominance relation is defined on the set of stable matchings in an SMP instance (see [4]). Such a relation on strongly stable matchings can be defined as follows. Let I be an SMT instance and $\mathcal{M}$ be the set of strongly stable matchings in I. Let $M, M' \in \mathcal{M}$. M *dominates* M', written $M \preceq M'$,if each man strictly prefers M to M', or is different between them [11]. We denote by $\mathcal{C}$ the set of equivalence classes of $\mathcal{M}$ under $\sim$, and by $[M]$ the equivalence class containing M, for $M \in \mathcal{M}$. A partial order, denoted by $\preceq^*$, is defined on equivalence classes as follows.

Definition 2 ([11]). *For two equivalence classess $[M], [M'] \in \mathcal{C}$, $[M] \preceq^* [M']$ if and only if $M \preceq M'$.*

The distributive lattice structure of strongly stable matchings for an SMT instance is given in the following, which helps us to develop a searching for optimal matchings (described in Sect. 3).

Theorem 1 (Distributive lattice [11]). *Let I be an SMT instance, $\mathcal{M}$ be the set of strongly stable matchings in I. Let $\mathcal{C}$ be the set of equivalence classes of $\mathcal{M}$ under $\sim$, and $\preceq^*$ be the dominance partial order on $\mathcal{C}$. Then $(\mathcal{C}, \preceq^*)$ forms a finite distributive lattice.*

3 Searching Algorithm

Our scheme is to locally search, among neighborhoods of a strongly stable matchings, a better matching with respect to an optimal criterion, i.e. egalitarian or sex-equal. The selection of a better matching is performed due to the dominance relation. The distributive lattice of strongly stable matchings for an SMT instance (as indicated in Theorem 1), is the essential structure for such a search.

Given an SMT instance, the existence of a strongly stable matching can be determined and such a matching can be found in $O(n^4)$ [6,10]. If a strongly matching exists, the search starts at the matching and then iteratively performs the followings: (i) determining a set of neighbors of the matching, (ii) then selecting a better matching among them due to an optimal criterion. The search terminates when meeting a top condition (mentioned in the end of this section). Task (i) can be performed by using a modification of the breakmarriage operation as below.

Breakmarriage for Strongly Stable Matchings

The concept of breakmarriage operation was introduced by McVitie and Wilson [12]. Let M be a matching in a given SMP instance and let (m, w) be a pair in M. The operation breaks the marriage of m and w. Starting with m proposing the woman following w in his preference list, it then performs a sequence of proposals, acceptances, and rejections as given by the Gale and Shapley's Algorithm [1]. The operation terminates if w is engaged to m' she prefers to m or some man is rejected by all women. If the operation terminates with the former case, i.e. all men are engaged, we obtain a new matching, denoted by M'. It was shown in [12] that M' is stable too. Furthermore, M' dominates M.

Unlike in SMP, the acceptance and rejection of a woman w in the sequence of proposals, acceptances, and rejections are different for strongly stability in SMT instances. Namely, by the definition of strongly stability, w can accept a man she prefers to her current partner or the man and her current partner are indifferent. Let (m, w) be a pair in strongly stable matching in an SMT instance. The breakmarriage operation should be modified as follows. Let us consider the situation when w is received a proposal from some man:

(a) If the man is indifferent between his partner and w, w accepts the man only if she strictly prefers the man to m.
(b) Otherwise, w accepts the man if she strictly prefers the man to m or is indifferent between them.

Proposition 1. *Let M be a strongly stable matching in a given SMT instance. If a matching M' is obtained by performing the breakmarriage operation due to (a) and (b), M' is also strongly stable.*

Proof. Regarding the stability, we use the similar argument in the proof of Theorem 3 in [12]. In particular, all pairs in M' that keep the same in M, are still stable. The remaining pairs in M' are generated in which a man gets a better partner and a woman gets a worse partner, or he/she is indifferent between the current partner and new one. In the meanwhile, by the rules (a) and (b), the strongly stability is maintained.

The Algorithm

We now describe the algorithm. Let M be a strongly stable matching in a given SMT instance. Procedure FindNext(M) determines a better matching for the search. It computes the set of neighbors using the modified breakmarriage operation, and then chooses a matching in the set that has minimum egalitarian or sex-equal cost, given by Eqs. (1) and (2). Because our algorithm is local search based, the search can get stuck at a local minimum. We can overcome this issue by randomly choosing the next matching due to a small value of p.

```
procedure FindNext(M)
    neighborSet := ∅
    for m := 1 to n do
        M_child := Breaking(M, m)
        if (M_child ≠ NULL) then
            neighborSet := neighborSet ∪ M_child
        end if
    end for
    if (small random probability p) then
        M_next := a random matching in neighborSet
    else
        M_next := arg min_{M ∈ neighborSet}(f(M))
    end if
    return M_next
end procedure
```

The notation m in Procedure FindNext(M) means that we perform breakmarriage with respect to man. If the role of man and woman is swapped, a breakmarriage operation with respect to woman is established in the same manner.

Algorithm 1. Searching algorithm for strongly stable matchings

```
Input: an SMT instance with preference lists A and B.
Output: a strongly stable matching.

M_left := ManOptimal(A, B)
M_right := WomanOptimal(B, A)
if (A strongly stable matching exists) then
    M_best := arg min_{M ∈ {M_left, M_right}} (f(M))
    forward := true
    backward := true
    loop
        if (forward) then searching w.r.t man point of view
            M_next := FindNext(M_left)
            if (f(M_next) > f(M_left)) then
                forward := false
                if f(M_best) > f(M_left) then
                    M_best := M_left
                end if
            end if
            M_left := M_next
        end if
        if (backward) then searching w.r.t woman point of view
            M_next := FindNext(M_right)
            if (f(M_next) > f(M_right)) then
                forward := false
                if f(M_best) > f(M_right) then
                    M_best := M_right
                end if
            end if
            M_right := M_next
        end if
        if ((not forward) and (not backward)) then
            if (sm(M_left) ≤ sm(M_right)) then
                forward := true
                backward := true
            else
                break
            end if
        end if
    end loop
    return M_best
else
    There is no strongly stable matching.
end if
```

From that observation, we can develop an algorithm consisting of two simultaneously local searches. One carries out with respect to man point of view, and the other is with respect to woman point of view. These two searches not only aims to speed up the algorithm but also needs to improve the accuracy solutions obtained by the algorithm. To this end, it is important to determine appropriately an initial matching as well as the searching direction for each search. The distributive lattice structure as described in Sect. 2 allows again us to solve this.

Our search is given in Algorithm 1. On the lattice, we performs two searches in which one starts at a man-optimal strongly stable matching and the other at a woman-optimal one. The former called *forward*, by the breakmarriage operation with respect to man, computes the set of neighbors and then chooses a better matching due to the optimal cost $f(M)$ (that is the egalitarian or sex-equal cost). In the meantime, the later called *backward* does in the same manner, with respect to woman point of view. Two searches are performed iteratively until a so-call meeting condition is satisfied. The algorithm, thereby, travels on the lattice due to the dominance relation. Because of the dominance, man costs of matchings found during forward searching should be increased, while those during backward searching should be decreased. The meeting condition is defined to be the moment when the man cost of a matching by forward search is greater than that by backward one, i.e., $sm(M_{left}) > sm(M_{right})$. The idea behind of Algorithm 1 was actually introduced in [16], which is called the bi-directional local search. It, however, defers from which in [16] that the breakmarriage operation here is modified to adapt to the strongly stability. Furthermore, some computational aspects of the set of stable marriages in an SMT instance, are also utilized to improve the performance of the algorithm as described in the next section.

4 Implementation

4.1 Simulation Results

We used the method given in [3] to generate SMT instances. Due to the method, we take two parameters: the problem's size n and a probability, say p_t, of ties. Given a 2-tuple $\langle n, p_t \rangle$, an instance of SMT is generated iteratively as follows:

1. A random preference list of size n for each man and woman is produced.
2. We iterate over each person's (men and women's) preference list: for a man m_i and for his choices c_i from his second to his last, a random value p such that $0 \leq p < 1$, is generated; if $p \leq p_t$ then the preference for his c_i^{th} choice is the same as c_{i-1}^{th} choice.

An instance generated as $\langle n, 0.0 \rangle$ will be a classical SMP. Table 1 shows an SMT instance which is randomly generated with $\langle 8, 0.5 \rangle$ (ties are denoted by braces).

The algorithm is implemented in Matlab 2016a and run on the platform OS X, Core i5 2.5 GHz with 8 GB RAM. We run the algorithm for SMT instances of different sizes. For each instance, the algorithm is tested for 10 times obtain the average costs and the probability of ties p_t is varied from 0.0 to 1.0 in steps of 0.01.

We first investigate how the parameter p_t influences the egalitarian and sex-equal costs of an SMT instance. Figure 1 shows the average cost of egalitarian

Table 1. An SMT instance generated with $\langle 8, 0.5 \rangle$.

Men's list	Women's list
1: (5 7) (1 2) 6 (8 4) 3	1: 5 3 7 (6 1 2 8) 4
2: 2 (3 7) 5 4 1 8 6	2: 8 6 (3 5 7 2 1 4)
3: (8 5 1) 4 6 2 3 7	3: (1 5 6) (2 4) 8 7 3
4: (3 7 2) 4 1 (6 8 5)	4: 8 (7 3 2) (4 1 5) 6
5: (7 2) 5 1 3 (6 8 4)	5: (6 4 7 3 8) (1 2) 5
6: (1 6) (7 5) 8 (4 2 3)	6: 2 8 (5 4) (6 3) 7 1
7: (2 5) 7 6 3 4 (8 1)	7: (7 5 2) 1 8 (6 4) 3
8: (3 8 4 5) (7 2 6 1)	8: (7 4 1) 5 2 3 6 8

and sex-equal matchings found by the algorithm for $\langle 100, p_t \rangle$, varying p_t. The trend of egalitarian costs is decreased as p_t increases. This is because of the definition of egalitarian cost, i.e. $sm(M) + sw(M)$ (see Eq. (1)), for a given matching M. When p_t increases, there are more people which are indifferent in preference lists. Therefore, the values of $sm(M)$ and $sw(M)$ should be decreased. For the sex-equal cost, by (2), it is defined to be $|sm(M) - sw(M)|$. The figure shows that the probability of the cost that is closer to zero, is high as higher value of p_t. These observations on the behaviors of egalitarian and sex-equal costs, when varying p_t, thus also indicate the correctness of the algorithm.

In Figs. 2 and 3, we show the results of testing for datasets of large size. We report here the average execution time of the algorithm which runs for five instance sizes $n = 50, 100, 200, 300, 500$, varying p_t in steps of 0.01. The running times for finding egalitarian and sex-equal matchings are quite similar for each instance size. It also seems that the variability of running times is higher as p_t increases. This can be caused by increasing the number of ties in preference lists. Then, the number of strongly stable matchings can also be increased. The searching space should be larger and thus the necessary time for searching might sometimes be increased. On the other hand, a high people number of ties can also help to quickly obtain solutions, since the stop condition of the algorithm can be quickly reached. Finally, we would also like to show here the performance of the algorithm for large datasets. For $n \leq 300$, the running time mostly is not exceeded $5\,(s)$ for both egalitarian and sex-equal costs. For largest size of 500, the times in worse cases mostly are about $30\,(s)$ and $35\,(s)$ for egalitarian and sex-equal costs, respectively. Such execution times indicate the significance of our algorithm.

4.2 Algorithm Acceleration

Procedures ManOptimal(A, B) and WomanOptimal(B, A) in Algorithm 1 can be given by STRONG given in [6] which is an extension of Gale and Shapley's algorithm.

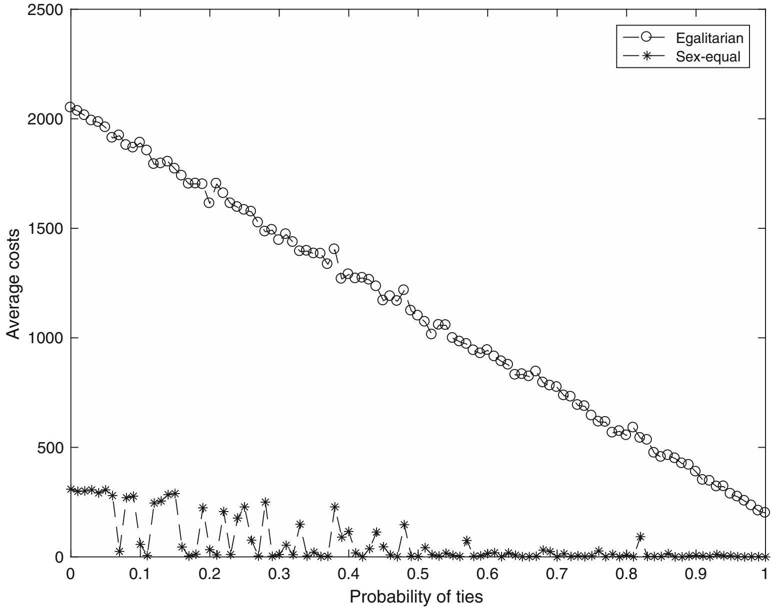

Fig. 1. The average cost of egalitarian and sex-equal matchings for $\langle 100, p_t \rangle$.

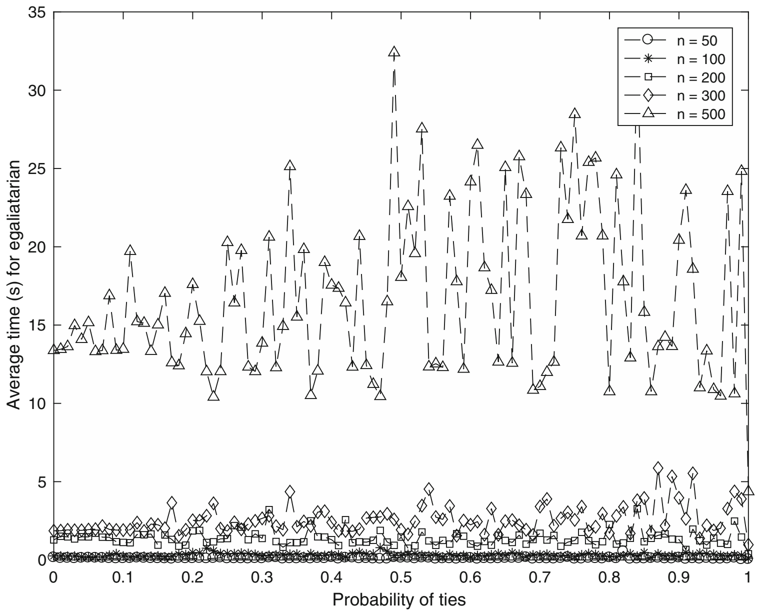

Fig. 2. The average execution time of finding egalitarian matchings for $\langle n, p_t \rangle$.

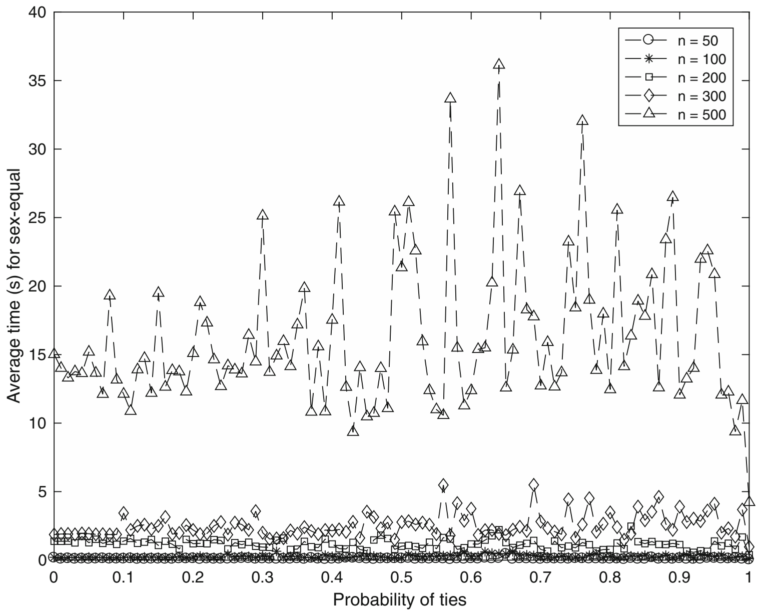

Fig. 3. The average execution time of finding sex-equal matchings for $\langle n, p_t \rangle$.

In Algorithm 1, For each iteration of finding a next matching (Procedure FINDNEXT(M)), the algorithm performs a number of breakmarriage operations (Procedure BREAKING(M, m)). This number should be very large as the size of instances increases. Consequently, the computational cost of the breakmarriage operation is high. It is possible to reduce the cost by using a so-called shortlist. The concept of shortlists for a given SMP instance was given in [7]. In particular, if a woman w accepts a proposal from a man m, then the woman never accepts a proposal from a man following m in her preference list. Then, all men following m should be removed from the list. We also remove w from preference lists of the men. It was known that if w (m, respectively) is absent in the shortlist of a man m (w, respectively), (m, w) is not a stable pair. Thereby, in the sequence of proposals of a breakmarriage operation, before proposing to a person of opposite sex, he/she checks if the person is in his/her shortlist. If the person is absent in the list, the pair of them is not stable. Therefore, we can ignore the stability checking for the pair.

In the case of ties, for a given SMT instance, shortlists with respect to man point of view can be obtained during performing Procedure MANOPTIMAL(A, B) as follows. When a woman w accepts the proposal from a man m:

1. remove all men whose strictly worse preference than m from the preference list of w,
2. remove also w from the preference lists of those men.

Let us denote by X^m and Y^m such shortlists (i.e. obtained with respect to man point of view). From woman point of view, similar shortlists can be obtained in the same manner, denoted by X^w and Y^w. We now define shortlists for the instance, denoted by X and Y,

$$X = X^m \wedge X^w \quad \text{and} \quad Y = Y^m \wedge Y^w,$$

where the operator $\wedge$ is defined by $X(m_i, w_j) = X^m(m_i, w_j)$ if $X^m(m_i, w_j) = X^w(m_i, w_j)$, otherwise $X(m_i, w_j) = 0$ (meaning that w_j is absent from m_i's preference list). It is similar to Y. Using shortlists, we now improve the Procedure FindNext(M) to FindNext(X, Y, M), where the breakmarriage takes X and Y as parameters to reduce the number of stability checking operations for pairs of a man and woman.

```
procedure FindNext(X, Y, M)
    neighborSet := ∅
    for m := 1 to n do
        M_child := Breaking(X, Y, M, m)
        if (M_child ≠ NULL) then
            neighborSet := neighborSet ∪ M_child
        end if
    end for
    if (small random probability p) then
        M_next := a random matching in neighborSet
    else
        M_next := arg min_{M∈neighborSet}(f(M))
    end if
    return M_next
end procedure
```

5 Conclusion

The paper presented a searching for strongly egalitarian and sex-equal matchings of SMT instances. The algorithm performs two simultaneous local searches on the distributive lattice structure of strongly stable matchings in SMT instances to find optimal solutions. By dominance relation in the lattice, the algorithm terminates when the man costs of two searches "meet" each other. The simulation results obtained on large datasets show that the algorithm can find solutions in a reasonable time. In order to speed up the searching, a use of shortlist concept also was presented, which helps to reduce the number of checking pairs in the breakmarriage operation. It however has not implemented yet in this paper. This is performed in a general locally searching scheme in which we aims to develop for several variants of SMP.

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

References

1. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am. Math. Monthly **69**(1), 9–15 (1962)
2. Gelain, M., Pini, M., Rossi, F., Venable, K., Walsh, T.: Local search approaches in stable matching problems. Algorithms **6**(4), 591–617 (2013)
3. Gent, I.P., Prosser, P.: An empirical study of the stable marriage problem with ties and incomplete lists. In: Proceedings of European Conference on Artificial Intelligence, Lyon, France, pp. 141–145 (2002)
4. Gusfield, D.: Three fast algorithms for four problems in stable marriage. SIAM J. Comput. **16**(1), 111–128 (1987)
5. Irving, R.W.: An efficient algorithm for the "stable roommates" problem. J. Algorithms **6**, 577–595 (1985)
6. Irving, R.W.: Stable marriage and indifference. Discrete Appl. Math. **48**, 261–272 (1994)
7. Irving, R.W., Leather, P., Gusfield, D.: An efficient algorithm for the "optimal" stable marriage. J. ACM **34**(3), 532–543 (1987)
8. Iwama, K., Manlove, D., Miyazaki, S., Morita, Y.: Stable marriage with incomplete lists and ties. In: Proceedings of the 26th International Colloquium on Automata, Languages, and Programming, LNCS, vol. 1644, pp. 443–452. Springer (1999)
9. Iwama, K., Miyazaki, S., Yanagisawa, H.: Approximation algorithms for the sex-equal stable marriage problem. ACM Trans. Algorithms **7**(2) (2010). Article no. 2
10. Manlove, D.: Stable marriage with ties and unacceptable partners. Technical report TR-1999-29, University of Glasgow (1999)
11. Manlove, D.F.: The structure of stable marriage with indifference. Discrete Appl. Math. **122**, 167–181 (2002)
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: Proceeding of 29th AAAI Conference on Artificial Intelligence, Austin, Texas, United States, pp. 1212–1218 (2015)
14. Nakamura, M., Onaga, K., Kyan, S., Silva, M.: Genetic algorithm for sex-fair stable marriage problem. In: Proceedings of IEEE International Symposium on Circuits and Systems, Seattle, WA, ISCAS 95, vol. 1, pp. 509–512 (1995)
15. Vien, N.A., Viet, N., Kim, H., Lee, S., Chung, T.: Ant colony based algorithm for stable marriage problem, vol. 1, pp. 457–461. Springer, Dordrecht (2007)
16. Viet, H.H., Trang, L.H., Lee, S., Chung, T.: A bidirectional local search for the stable marriage problem. In: Proceedings of the 2016 International Conference on Advanced Computing and Applications, Can Tho City, Vietnam, pp. 18–24 (2016)
17. Viet, H.H., Trang, L.H., Lee, S., Chung, T.: An empirical local search for the stable marriage problem. In: Proceedings of the 14th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence, Phuket, Thailand, pp. 556–564 (2016)