Journal of Experimental & Theoretical Artificial Intelligence

ISSN: 0952-813X (Print) 1362-3079 (Online) Journal homepage: https://www.tandfonline.com/loi/teta20

A shortlist-based bidirectional local search for the stable marriage problem

Hoang Huu Viet, Le Hong Trang, Le Pham Tuyen & Taechoong Chung

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE
https://doi.org/10.1080/0952813X.2019.1635655

ARTICLE

A shortlist-based bidirectional local search for the stable marriage problem

Hoang Huu Viet[a], Le Hong Trang[b], Le Pham Tuyen[c] and Taechoong Chung[c]

[a]Information Technology Department, Vinh University, NgheAn, Vietnam; [b]Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, VNU-HCM, Vietnam; [c]Computer Engineering Department, Kyung Hee University, Seoul, South Korea

ABSTRACT

This paper proposes a shortlist-based bidirectional local search algorithm to find an approximate solution for either the *egalitarian* or the *sex-equal* matching of the stable marriage problem. Our approach simultaneously searches forward from the *man-optimal* matching and backwards from the *woman-optimal* matching until the search frontiers meet. By using a shortlist-based breakmarriage strategy to rapidly generate all the stable neighbour matchings of all *k*-best stable matchings, the forward local search finds the solutions while moving towards the *woman-optimal* matching and the backward local search finds the solutions while moving towards the *man-optimal* matching. The experiments demonstrate that our proposed algorithm is efficient for finding an approximate solution to the *egalitarian* or *sex-equal* matching of the stable marriage problem.

ARTICLE HISTORY
Received 4 March 2019
Accepted 13 June 2019

KEYWORDS
Breadth first search; breakmarriage; gale-shapley; local search; stable marriage problem

Introduction

The stable marriage problem was first introduced by Gale and Shapley (Gale & Shapley, 1962), 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 (Roth, 1984), the Student-Project Allocation problem (Abraham, Irving, & Manlove, 2003), and the Stable Roommates problem (Fleiner, Irving, & Manlove, 2007; Irving, 1985).

A stable marriage (SM) problem of size n comprises a set of n men and a set of n women in which each person ranks all members of the opposite sex in order of preference in their preference list. The aim of the problem is to match men and women to satisfy a certain stability criterion. A matching, M, is a set of n disjoint pairs of men and women. If a man, m, and a woman, w, are paired up in M, then m and w are said to be partners in M, denoted by $m = M(w)$ and $w = M(m)$. A man, m, and a woman, w, form a blocking pair in a matching, M, if m prefers w to $M(m)$ and w prefers m to $M(w)$. A matching, M, which has no blocking pairs is said to be *stable*, otherwise it is said to be *unstable*. Let $\mathcal{M}$ denote a set of all stable matchings, $p_m(w)$ denote the position of woman w in man m's preference list and $p_w(m)$ denote the position of man m in woman w's preference list.

For a stable matching, $M \in \mathcal{M}$, the *man cost*, $sm(M)$, and the *woman cost*, $sw(M)$, are defined as follows:

$$sm(M) = \sum_{(m,w)\in M} p_m(w), \tag{1}$$

CONTACT Hoang Huu Viet viethh@vinhuni.edu.vn Information Technology Department, Vinh University, NgheAn, Vietnam; Taechoong Chung tcchung@khu.ac.kr Computer Engineering Department, Kyung Hee University, Seoul, South Korea

$$sw(M) = \sum_{(m,w) \in M} p_w(m). \tag{2}$$

Definition 1.1 (man-optimal and woman-optimal). A stable matching, M, is called *man-optimal* (resp., *woman-optimal*) if it has the minimum value of $sm(M)$ (resp., $sw(M)$) for all $M \in \mathcal{M}$.

Gale and Shapley proposed an algorithm known as the Gale-Shapley algorithm to find an optimal solution of SM instances of size n in $O(n^2)$ time (Gale & Shapley, 1962). The Gale-Shapley algorithm is a sequence of proposals from men to women to find the man-optimal matching. If the roles of men and women are interchanged, the matching found by the algorithm is the woman-optimal matching. The man-optimal (resp., woman-optimal) matching is the '*selfish*' matching for men (resp., 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 such as an *egalitarian* or *sex-equal* matching to give more balanced preferences for both men and women. For a stable matching, $M \in \mathcal{M}$, the *egalitarian cost*, $c(M)$, and the *sex-equality cost*, $d(M)$, are defined as follows:

$$c(M) = sm(M) + sw(M), \tag{3}$$

$$d(M) = |sm(M) - sw(M)|. \tag{4}$$

Definition 1.2 (egalitarian and sex-equal). A stable matching, M, is called *egalitarian* (resp., *sex-equal)* if it has the minimum value of $c(M)$ (resp., $d(M)$) for all $M \in \mathcal{M}$.

In this paper, we propose a shortlist-based bidirectional local search algorithm, titled *ShortL-BiLS*, to seek an approximate solution for either the *egalitarian* or the *sex-equal* matching of SM instances. Our algorithm runs two simultaneous local searches: one forward from the *man-optimal* matching and the other backward from the *woman-optimal* matching. At each iteration, each search uses a shortlist-based breakmarriage strategy to generate all the stable neighbour matchings of all k-best stable matchings, selects the best solution from the stable neighbour matchings, keeps the k-best successors from the stable neighbour matchings and repeats for the next iteration. The algorithm terminates when two search frontiers meet each other and it gives the best solution found so far. The experimental results show that our *ShortL-BiLS* is efficient in terms of execution time and solution quality for large SM problems.

The rest of this paper is organized as follows. Section 2 describes the related work, Section 3 provides the background, Sections 4 presents our *ShortL-BiLS* algorithm, Section 5 discusses the experiments and evaluations, and Section 6 concludes our work.

Related work

The basic approach to find the egalitarian or sex-equal matching of SM instances is an exhaustive search. For SM instances of size n, Gusfield (Gusfield, 1987) proposed an algorithm to enumerate all stable matchings in $O(n^2 + n|S|)$ time, where $|S|$ is the number of stable matchings of SM instances. Irving et al. (Irving, Leather, & Gusfield, 1987) exploited a lattice structure of a stable matching set and used graph-theoretic methods to propose an $O(n^4)$ algorithm to find the egalitarian of SM instances.

There are several heuristic approaches to find the egalitarian or sex-equal matching. Nakamura et al. (Nakamura, Onaga, Kyan, & Silva, 1995) proposed a genetic algorithm (GA) for finding the sex-equal matching of SM instances. In their approach, the problem is first transferred into a directed graph and the GA is used to find the solution in the graph. Zavidovique et al. (Zavidovique, Suvonvorn, & Seetharaman, 2005) 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. Vien et al. (Vien, Viet, Kim, Lee, & Chung, 2007) presented an ant colony system (ACS) algorithm for finding the man-optimal, woman-optimal, egalitarian, or sex-equal matching. Unfortunately, the ACS algorithm finds the optimal 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. Iwama et al. (Iwama, Miyazaki, & Yanagisawa, 2010) proposed an algorithm in $O(n^{3+\frac{1}{\epsilon}})$ time which achieves a stable matching, M, such that $|d(M)| \leq \epsilon\Delta$, where ϵ is a given constant, $\Delta = min\{d(M_0), d(M_t)\}$, and M_0 and M_t are man- and woman-optimal matchings, respectively. In addition, they proposed a variant for finding a stable matching, M, which minimizes $c(M)$ under the condition that $|d(M)| \leq \epsilon\Delta$, which runs in time $O(n^{3+2(\frac{1+\epsilon}{\delta})})$ for an arbitrary δ such that $0 < \delta < \epsilon$. Everaere et al. (Everaere, Morge, & Picard, 2013) proposed a $\mathbb{S}$wing algorithm for finding the sex-equal matching of SM instances. At the odd iterations of the algorithm, the men play the role of proposers and the women play the role of responders and the roles are swapped at the even iterations. When the $\mathbb{S}$wing stops, it takes $O(n^3)$ time to obtain a stable matching other than the sex-equal matching. Giannakopoulos et al. (Giannakopoulos, Karras, Tsoumakos, Doka, & Koziris, 2015) provided an ESMA algorithm (Giannakopoulos et al., 2015) which the idea is similar to that of $\mathbb{S}$wing. 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. The ESMA terminates in $O(n^2)$ time and yields a stable matching, which has high $c(M)$ and low $d(M)$ but neither the egalitarian nor sex-equal matchings.

Recently, Gelain et al. (Gelain, Pini, Rossi, Venable, & Walsh, 2013) proposed a local search algorithm, called *SML*, for finding an arbitrary stable matching of SM instances. Starting at a randomly generated matching, M, the *SML* produces a set of the neighbour matchings of M, where a neighbour is determined by removing one of the blocking pairs in M, and moves M to the neighbour matching which has the smallest number of blocking pairs. This process iteratively performs until the stability in M is obtained. Viet et al. (Viet, Trang, Lee, & Chung, 2016b) developed an empirical algorithm, denoted by *SLS*, for finding an approximation solution in terms of the egalitarian or sex-equal matching. The *SLS* is a sequence of local searches, in which each search is a hill-climbing search (Russel & Norvig, 2010) which uses the breakmarriage operation (McVitie & Wilson, 1971) to find all the stable neighbour matchings of the current stable matching. The first hill-climbing algorithm starts from the man-optimal solution and the next one starts from the solution of the previous one. The *SLS* stops when it reaches the woman-optimal matching and the solution is the best one among the solutions of all the hill-climbing searches. Because the *SLS* is a unidirectional search from the man- to the woman-optimal matching, it is inefficient in terms of execution time for large SM problems. To improve this weakness, we proposed a bidirectional local search (Viet, Trang, Lee, & Chung, 2016a), called *BiLS*, which runs two simultaneous local searches: one forward from the man-optimal matching and the other backward from the woman-optimal matching, and it terminates when they both stop and meet. However, when the breakmarriage operation (McVitie & Wilson, 1971) is applied with the men's and women's full preference lists of the SM problem, it has to check every pair of man and woman even if the man and woman form a blocking pair in a matching. Therefore, the *BiLS* is inefficient for large SM problems.

Background

Finding all the stable matchings

McVitie and Wilson proposed a breakmarriage operation (McVitie & Wilson, 1971), denoted by BreakMarriage(M, m), to find a new stable matching from a prior one, M, and a man, m. They showed that every stable matching, M_i $(i = 1, 2, \cdots, t)$, can be obtained by a series of breakmarriage operations starting from the man-optimal matching, M_0, where M_t is the woman-optimal matching. However, if the breakmarriage operation runs for each man, m, in the men set, it can produce some duplicate stable matchings. To enumerate all unique stable matchings, they applied two rules on the breakmarriage operation as follows.

R_1: If BreakMarriage(M, m) returns a stable matching, M', then it is only run for M' on men $i \geq m$.

R_2: In BreakMarriage(M, m), only men $m' \geq m$ propose, i.e., if some man m' is free and $m' < m$ during the execution, then BreakMarriage(M, m) stops and returns no matching.

Let A and B denote the preference lists of men and women, respectively, of an SM instance of size n. The algorithm to find all the stable solutions (McVitie & Wilson, 1971) can be represented by a breadth first search (BFS) algorithm shown in Algorithm 1. Initially, the Gale-Shapley algorithm is used to find the man-optimal matching, M_0. Starting at M_0 as a root node, the algorithm constructs a tree of all stable matchings. For each node of a stable matching, M, a set of its children is produced using BreakMarriage(M, m) for each man, m, from $x[i]$ to n, where $x[i]$ is a man which BreakMarriage(M, m) breaks M and successfully returns a stable matching, M'. A leaf of the tree is a stable matching which cannot be broken to another, i.e., the BreakMarriage(M, m) returns empty. The algorithm ends when it reaches the woman-optimal matching, M_t, since BreakMarriage(M_t, m) does not produce any stable matching. After the algorithm ends, it returns a stable matching set, $\mathcal{M}$, and an optimal stable matching, M^{opt}, with respect to the search criterion, defined by the cost function $f(M)$, which is $c(M)$ or $d(M)$ with respect to finding the egalitarian or sex-equal matching.

Algorithm 1: *BFS algorithm*

Input: The men's preference list, A, and the women's preference list, B.
Output: The best matching, M_{opt}, and the stable matching set, $\mathcal{M}$.

$M_0 := $ Gale-Shapley(A, B);
$M^{opt} := M_0$; $\mathcal{P} := \{M_0\}$; $\mathcal{M} := \{M_0\}$;
$x[1] := 1$; ▷ used for the R_1 rule;
while (*true*) **do**
    $\mathcal{C} := \emptyset$; $k := 1$; $y[k] := 0$;
    for $i := 1$ **to** $card(\mathcal{P})$ **do**
        $M := \mathcal{P}(i)$;
        for $m := x[i]$ **to** n **do**
            $M' := $ BreakMarriage(M, m);
            if $(M' \neq NULL)$ **then**
                $\mathcal{C} := \mathcal{C} \cup \{M'\}$;
                $\mathcal{M} := \mathcal{M} \cup \{M'\}$;
                $y[k] := m$;
                $k := k + 1$;
    if $(\mathcal{C} = \emptyset)$ **then** break;
    $M_{best} := \operatorname{argmin}(f(M))$, where $M \in \mathcal{C}$;
    if $f(M^{opt}) > f(M_{best})$ **then**
        $M^{opt} := M_{best}$;
    $\mathcal{P} := \mathcal{C}$; $x := y$;
return M^{opt} and $\mathcal{M}$;

Consider an SM instance consists of eight men and eight women with the preference lists given in Table 1. Figure 1 shows the tree of all stable matchings produced by the BFS algorithm. The

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

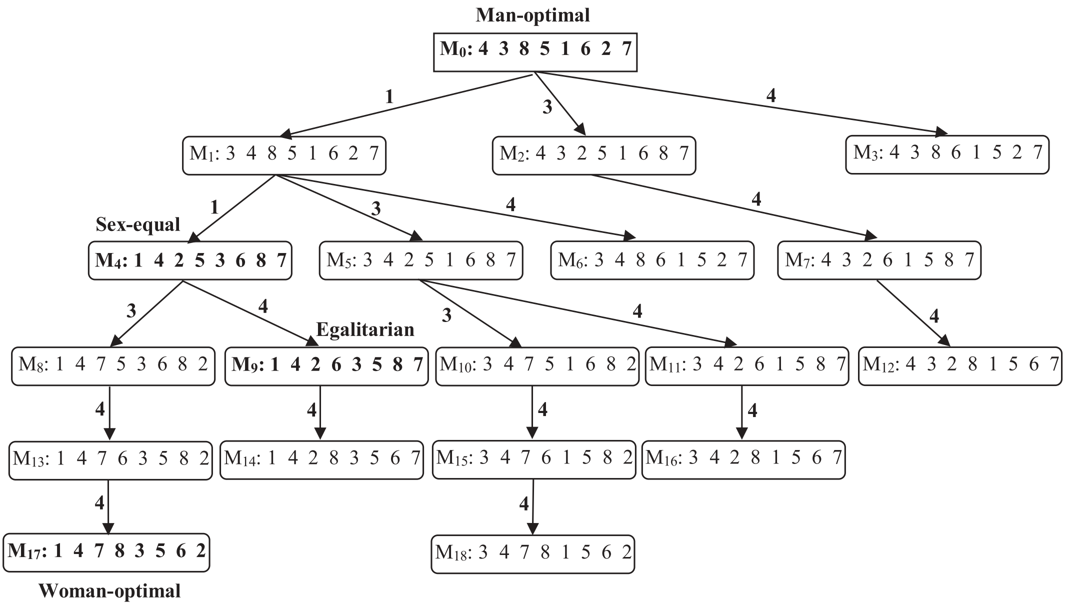

Figure 1. The tree found by the BFS algorithm for Table 1. The number in position i $(i = 1, 2, \ldots, 8)$ of any matching indicates the woman paired to man i in that matching.

man-optimal matching is the root of the tree and each node $M_i (i = 1, \ldots, 18)$ is obtained by the breakmarriage operation on man m, where m is indicated on the branches of the tree.

Shortlists for the SM problem

In the Gale-Shapley algorithm, at each iteration, when a free man, m, proposes to the most preferred woman, w, on his preference list to whom he has not proposed, meaning that (*i*) there exists no stable matching in which m has a better partner than w; and (*ii*) if w accepts a proposal from m, then there exists no stable matching in which w has a worse partner than m (McVitie & Wilson, 1971). By exploiting the properties (*i*) and (*ii*), we can obtain the men's and women's shortlists from the men's and women's preference lists in the iterations of the Gale-Shapley algorithm. Specifically, in the case of the men as proposers, if a woman, w, accepts a proposal from some free man, m', that she prefers m' to her current partner m, then all the men that w less prefers to m' are removed from w's preference list and also w is removed from the men's preference lists. Then, the men's and women's preference lists after such removals are called the man-oriented men's and women's shortlists, respectively. Otherwise, if the roles of men and women are interchanged, we obtain the woman-oriented men's and women's shortlists from the men's and women's preference lists, respectively. Obviously, (*i*) if w does not appear on m's shortlist, then m and w cannot be partners in any stable matching and (*ii*) w appears on m's shortlist if and only if m appears on w's shortlist. Table 2 shows the men-oriented men's and women's shortlists obtained

Table 2. The man-oriented men's and women's shortlists.

Man	Shortlists	Woman	Shortlists
m_1	4 0 3 8 1 0 0 0	w_1	1 3 5 0 0 0 0 0
m_2	0 3 4 2 0 0 6 0	w_2	8 2 4 5 3 7 0 0
m_3	0 8 2 4 0 7 0 1	w_3	5 8 1 4 2 0 0 0
m_4	5 6 8 3 4 0 0 2	w_4	2 4 3 6 5 8 1 0
m_5	1 3 5 2 0 6 4 0	w_5	6 5 4 0 0 0 0 0
m_6	0 6 0 5 0 0 4 0	w_6	7 4 2 5 6 0 0 0
m_7	2 0 8 0 6 0 0 0	w_7	3 8 0 0 0 0 0 0
m_8	0 7 4 0 0 2 0 3	w_8	4 7 1 3 0 0 0 0

by the Gale-Shapley algorithm for the SM instance in Table 1 by removing unacceptable men and women from every woman's and man's preference lists in the iterations of the Gale-Shapley algorithm.

Proposed shortL-BiLS algorithm

Breakmarriage operation with shortlists

We recognized that if the breakmarriage operation (McVitie & Wilson, 1971) is used to construct a stable matching, it has to check every pair (m, w) even if the pair (m, w) is a blocking pair. This is avoided by using shortlists instead. In particular, if a woman, w, does not appear in the m's shortlist, then checking (m, w) to be a blocking pair is ignored and therefore, the breakmarriage operation results in a huge reduction of computational time for checking blocking pairs. Thus, we improve the breakmarriage operation (McVitie & Wilson, 1971) such that it adapts to the shortlists.

The men's and women's shortlists found by the Gale-Shapley algorithm are man-oriented shortlists, as illustrated in Table 2. If the roles of men and women are interchanged, the men's and women's shortlists found by the Gale-Shapley algorithm are woman-oriented shortlists, as illustrated in Table 3. Because of the construction method of the shortlists in the Gale-Shapley algorithm, the man-oriented shortlists are different from the woman-oriented shortlists. Therefore, we combine the man-oriented shortlists and woman-oriented shortlists to reduce time for checking blocking pairs. Let $X^{(m)}$ and $Y^{(m)}$ be the man-oriented men's and women's shortlists, and $X^{(w)}$ and $Y^{(w)}$ be the woman-oriented men's and women's shortlists, respectively. The men's shortlists, X, and women's shortlists, Y, are composed, respectively, from $X^{(m)}$ and $X^{(w)}$, and $Y^{(m)}$ and $Y^{(w)}$, denoted by $X = X^{(m)} \wedge X^{(w)}$ and $Y = Y^{(m)} \wedge Y^{(w)}$, as given by Equations (5) and (6):

$$X(i,j) = \begin{cases} X^{(m)}(i,j) \text{ if } X^{(m)}(i,j) = X^{(w)}(i,j), \\ 0, \quad \text{otherwise}, \end{cases} \tag{5}$$

and

$$Y(i,j) = \begin{cases} Y^{(m)}(i,j) \text{ if } Y^{(m)}(i,j) = Y^{(w)}(i,j), \\ 0, \quad \text{otherwise}, \end{cases} \tag{6}$$

for $i, j = 1, 2, \cdots, n$.

Table 3. The woman-oriented men's and women's shortlists.

Man	Shortlists	Woman	Shortlists
m_1	4 7 3 8 1 0 0 0	w_1	1 0 5 0 0 0 8 0
m_2	5 3 4 0 0 0 0 0	w_2	8 0 0 0 3 7 0 6
m_3	3 8 2 4 6 7 0 0	w_3	5 0 1 0 2 3 0 7
m_4	5 6 8 0 0 0 0 0	w_4	2 0 3 0 0 8 1 0
m_5	1 3 0 0 0 0 0 0	w_5	6 0 4 8 0 7 2 0
m_6	8 6 2 5 0 0 0 0	w_6	7 4 0 0 6 8 0 3
m_7	2 5 8 3 6 0 0 0	w_7	3 8 0 0 0 0 1 0
m_8	5 7 4 1 6 2 0 0	w_8	4 7 1 3 0 0 0 6

Theorem 4.1. Given a man, m, if a woman, w, does not appear on m's shortlist in X, then there is no stable matching in which m and w are partners.

Proof. By hypothesis, w does not appear on m's shortlist in X, meaning that $X(m,w) = 0$. Therefore, $X^{(m)}(m,w)$ or $X^{(w)}(m,w)$ is equal to 0. This means that w is absent from m's shortlist in $X^{(m)}$ or $X^{(w)}$. It follows from Property 1 in (Irving et al., 1987) that m and w cannot be partners in any stable matching. □

We note that the property stated in Theorem 4.1 also holds if we alternatively consider Y given by Equation (6).

Table 4 shows the men's and women's shortlists found in the man-oriented shortlists in Table 2 and woman-oriented shortlists in Table 3. Obviously, the men's shortlists, X, and women's shortlists, Y, shown in Table 4 are sparse matrices, in which most of the elements of X and Y are zero. Therefore, if we use X and Y as inputs of the breakmarriage operation (McVitie & Wilson, 1971), the breakmarriage operation will result in a large reduction of the number of unnecessary iterations compared to the preference lists for checking blocking pairs. Moreover, in the woman-optimal matching, M_t, each man has the worst partner that he has in any stable matching and therefore, to make the breakmarriage operation more efficient, each man, m, is restricted to search a woman, w, whose position is less than that of the woman, $M_t(m)$, in the woman-optimal matching.

The shortlist-based breakmarriage operation is given by the *ShortL-BreakM* function. At each iteration, if there exists no free man, the function returns a stable matching, M'. Otherwise, if a free man, m', meets the R_2 rule, i.e. $m' < m$, then the function stops and returns no matching. Otherwise, the man m' proposes the most preferred woman, w', in his shortlist to whom he has not proposed and w''s position is less than that of the woman, $M_t(m')$. If w' prefers m' to her current partner, then w' rejected her partner to engage m'. If w' is different from w (the initial partner of m), the rejected man becomes free.

Function ShortL-BreakM(X, Y, M, m, M_t)

Input : – The men's shortlist, X, and the women's shortlist, Y;
– A stable matching M, a man m, and the woman-optimal matching M_t.

Output: – A stable matching M' or NULL.

```
w := M(m);
M(w) := 0;
while (true) do
    m' := find(M(w) = 0)
    if (m' is not found) then
        M' := M;
        break;
    if (m' < m) then
        return NULL;
    r := next woman's position on m''s list;
    if (X(m', r) > 0) then
        if (r > p_m'(M_t(m'))) then
            return NULL;
        w' := X(m', r);
        Determine p_w'(m') and p_w'(M(w')) in Y;
        if (p_w'(m') < p_w'(M(w'))) then
            M(m') := w';
            M(w') := m';
            if (w' ≠ w) then
                M(w') := 0;
return M';
```

Table 4. The men's and women's shortlists.

Man	Shortlists	Woman	Shortlists
m_1	4 0 3 8 1 0 0 0	w_1	1 0 5 0 0 0 0 0
m_2	0 3 4 0 0 0 0 0	w_2	8 0 0 0 3 7 0 0
m_3	0 8 2 4 0 7 0 0	w_3	5 0 1 0 2 0 0 0
m_4	5 6 8 0 0 0 0 0	w_4	2 0 3 0 0 8 1 0
m_5	1 3 0 0 0 0 0 0	w_5	6 0 4 0 0 0 0 0
m_6	0 6 0 5 0 0 0 0	w_6	7 4 0 0 6 0 0 0
m_7	2 0 8 0 6 0 0 0	w_7	3 8 0 0 0 0 0 0
m_8	0 7 4 0 0 2 0 0	w_8	4 7 1 3 0 0 0 0

Lemma 4.2. *If ShortL-BreakM* (X, Y, M, m, M_t) *returns a stable matching,* M'*, then* $sm(M) < sm(M')$ *and consequently,* $sw(M) > sw(M')$.

Proof. Since *ShortL-BreakM* returns a stable matching, M', there exists a subsequence of men, denoted by $m_1, m_2, \ldots, m_k$, such that, in M', m_i engages the woman following $M(m_i)$ in his preference list (by line 10), where $m_1 = m$ and $2 \leq k \leq n$. This means that

$$p_{m_i}(M(m_i)) < p_{m_i}(M'(m_i)), \text{ for } i = 1, 2, \ldots, k. \tag{7}$$

Summing up each side of Equation (7) yields the following:

$$\sum_{i=1}^{k} p_{m_i}(M(m_i)) < \sum_{i=1}^{k} p_{m_i}(M'(m_i)).$$

Since *ShortL-BreakM* only changes $k - 1$ pairs in M, $n - k + 1$ remaining pairs are unchanged in M'. Hence, $sm(M) < sm(M')$.

We note also that if M dominates M' from the man's point of view, then M' dominates M from the woman's point of view. This means that $sw(M) > sw(M')$. □

ShortL-BiLS algorithm

In this section, we propose a shortlist-based bidirectional local search algorithm, named *ShortL-BiLS*. Our algorithm is shown in Algorithm 2 to find either the egalitarian or the sex-equal matching of SM instances of size n. *ShortL-BiLS* runs two simultaneous searches: one forward from the man-optimal matching and the other backward from the woman-optimal matching. At the beginning, the Gale-Shapley algorithm is used to find the man- and woman-optimal matchings which are the starting solutions for the bidirectional search. At each iteration, for one of two searching directions, the algorithm finds a stable neighbour set, $\mathcal{N}$, of all k-best stable matchings in the set $\mathcal{N}_{left}$ (resp., $\mathcal{N}_{right}$) by calling the *ShortL-BreakM*(X, Y, M', m, M_t) (resp., *ShortL-BreakM*(Y, X, M', w, M_0)) function for each man (resp., woman), in turn, in the men (resp., women) set. The algorithm evaluates all the stable neighbour matchings in $\mathcal{N}$ using the cost function, $f(M)$, which is $c(M)$ or $d(M)$ with respect to finding the egalitarian or sex-equal matching. The algorithm then selects the next solution to be a neighbour whose cost function is the smallest value. If the next solution of each searching direction is worse than the current one, the search of the direction is paused. Furthermore, if the best solution of the direction found so far is worse than the current solution, then the current solution is the best one. The algorithm then moves the current solution to the next one, keeps the k-best matchings of $\mathcal{N}$ in $\mathcal{N}_{left}$ or $\mathcal{N}_{right}$ and repeats for the next iteration. The algorithm terminates when either one of the searching has no neighbours or two searchings meet each other by means of the man cost. In particular, if both forward and backward searches are paused and the man cost of the current matching of the forward search, $sm(M_{left})$, is equal or greater than that of the backward search, $sm(M_{right})$, then the bidirectional search is completed. Thus, the algorithm stops and gives the best solution so far.

Algorithm 2: *ShortL-BiLS algorithm*

Input : – The men's preference list, A, and the women's preference list, B
– An integer number, k
Output: – A stable matching, M_{best}

$[X^{(m)}, Y^{(m)}, M_0]$: = *Gale-Shapley*(A, B);
$[Y^{(w)}, X^{(w)}, M_t]$: = *Gale-Shapley*(B, A);
if ($M_0 = M_t$) **then** return M_t
$X = X^{(m)} \wedge X^{(w)}$; $Y = Y^{(m)} \wedge Y^{(w)}$;
$M_{best} := \text{argmin}(f(M))$, where $M \in \{M_0, M_t\}$;
$M_{left} := M_0$; $M_{right} := M_t$;
$\mathcal{N}_{left} := \{M_0\}$; $\mathcal{N}_{right} := \{M_t\}$;
forward := *true*; *backward* := *true*;
while (*true*) **do**
  if (*forward*) **then**
    $\mathcal{N} := \emptyset$
    for $t := 1$ **to** $card(\mathcal{N}_{left})$ **do**
      $M' := \mathcal{N}_{left}(t)$;
      for $m := 1$ n **do**
        M_{child} : = *ShortL-BreakM*(X, Y, M', m, M_t);
        if ($M_{child} \neq NULL$) **then** $\mathcal{N} := \mathcal{N} \cup \{M_{child}\}$;
    $M_{next} := \text{argmin}(f(M))$, where $M \in \mathcal{N}$;
    if ($f(M_{next}) > f(M_{left})$) **then**
      forward := *false*;
      if $f(M_{best}) > f(M_{left})$ **then** $M_{best} := M_{left}$;
    $M_{left} := M_{next}$;
    if ($k \geq card(\mathcal{N})$) **then**
      $\mathcal{N}_{left} := \mathcal{N}$;
    else
      $\mathcal{N}_{left} := \{M_i \in \mathcal{N} | f(M_i) \text{ is the } i^{th} \text{ smallest value}, i = 1, \cdots, k.\}$;
  if (*backward*) **then**
    $\mathcal{N} := \emptyset$
    for $t := 1$ **to** $card(\mathcal{N}_{right})$ **do**
      $M' := \mathcal{N}_{right}(t)$;
      for $w := 1$ **to** n **do**
        M_{child} : = *ShortL-BreakM*(Y, X, M', w, M_0);
        if ($M_{child} \neq NULL$) **then** $\mathcal{N} := \mathcal{N} \cup \{M_{child}\}$;
    $M_{next} := \text{argmin}(f(M))$, where $M \in \mathcal{N}$;
    if ($f(M_{next}) > f(M_{right})$) **then**
      backward := *false*;
      if $f(M_{best}) > f(M_{right})$ **then** $M_{best} := M_{right}$;
    $M_{right} := M_{next}$;
    if ($k \geq card(\mathcal{N})$) **then**
      $\mathcal{N}_{right} := \mathcal{N}$;
    else
      $\mathcal{N}_{right} := \{M_i \in \mathcal{N} | f(M_i) \text{ is the } i^{th} \text{ smallest value}, i = 1, \cdots, k.\}$;
  if *((not forward) and (not backward))* **then**
    if ($sm(M_{left}) \leq sm(M_{right})$) **then**

```
            forward := true; backward := true;
        else
            break
return M_best
```

An illustration of *ShortL-BiLS* to find a sex-equal matching for the SM instance in Table 1 is depicted in Figure 2, where $k = 1$. Initially, the algorithm assigns M_{left} to M_0 and M_{right} to M_{17}. At the first iteration, the algorithm finds a better solution in the neighbours of M_0 and moves M_{left} to M_1. The algorithm also finds a better solution in the neighbours of M_{17} and then moves M_{right} to M_8. The algorithm repeats for M_{left} and M_{right} until $M_{left} = M_4$ and $M_{right} = M_4$. At this point, no better solutions in the neighbours of M_{left} and M_{right} are found. Therefore, both searching directions are paused and $M_{best} = M_{left} = M_{right}$. Then, the algorithm moves M_{left} to M_9 and M_{right} to M_5. Since $sm(M_{left}) > sm(M_{right})$, the algorithm terminates and returns M_4, which is the sex-equal matching in Figure 1.

Theorem 4.3. ShortL-BiLS will stop after a finite number of iterations.

Proof. Gale and Shapley (Gale & Shapley, 1962) showed that there always exists at least one stable matching for any SM instance. Therefore, if an SM instance has only one stable matching, i.e., $M_0 = M_t$, then *ShortL-BiLS* returns $M = M_t$. Otherwise, *ShortL-BiLS* runs two searches simultaneously, one from man-optimal matching and the other from woman-optimal matching. For each step of moving of a search, *ShortL-BreakM* is called to move the next stable matching. According to Lemma 4.2, in forward searching from the man-optimal matching which is the minimal value of the man cost, the man costs of the stable matchings found by *ShortL-BreakM* increase. Meanwhile, the man costs decrease in

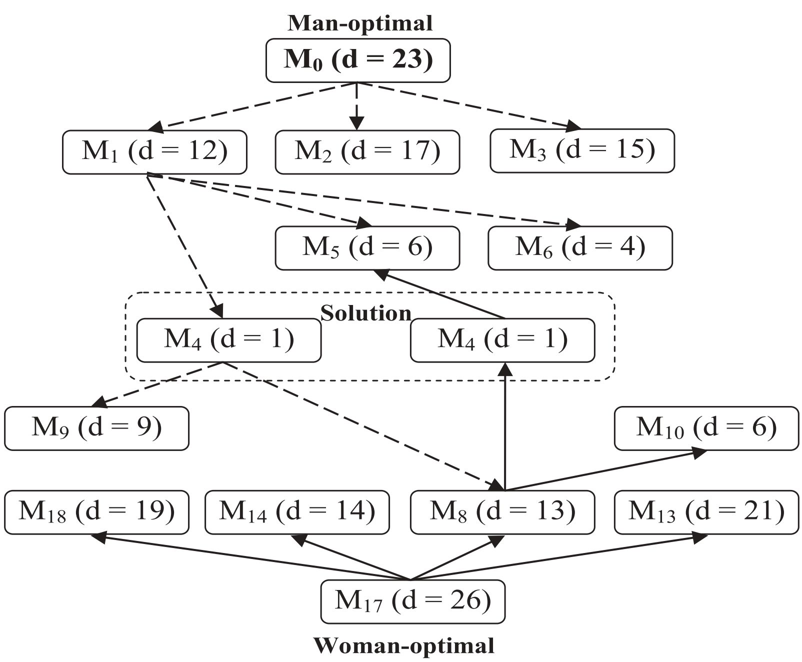

Figure 2. The search trace of finding the sex-equal matching of Table 1, where *d* indicated in any matching is the sex-equality cost of that matching.

backward searching where the algorithm starts at the woman-optimal matching which is the maximal value of the man cost. Therefore, at some step, two searches have to pass each other by means of the man cost of matchings, i.e., they respectively produce two stable matchings with $sm(M_{left}) > sm(M_{right})$. At this time, the algorithm terminates and gives the best matching obtained so far as a solution. □

Theorem 4.4. ShortL-BiLS takes $kO(dn^3)$ time to find a solution of an SM instance of size *n*, where *k* is the beam width and *d* is the maximum depth of search space.

Proof. *ShortL-BiLS* takes $O(n^2)$ time to determine a stable neighbour matching, M_{child}, of a stable matching, M', on a man, m, by using *ShortL-BreakM*(X, Y, M', m, M_t) (line 15) or *ShortL-BreakM*(Y, X, M', w, M_0) (line 31). Therefore, it takes $O(n^3)$ time to determine all the stable neighbour matchings of a stable matching, M', by using *ShortL-BreakM*(X, Y, M', m, M_t) for each man $m = 1, 2, \cdots, n$ (or by using *ShortL-BreakM*(Y, X, M', w, M_0) for each woman $w = 1, 2, \cdots, n$). At each iteration, *ShortL-BiLS* takes $kO(n^3)$ time to determine the stable neighbour matching set, $\mathcal{N}$, of $\mathcal{N}_{left}$ (or $\mathcal{N}_{right}$) and therefore, *ShortL-BiLS* takes totally $kdO(n^3) = kO(dn^3)$ time to find a solution of an SM instance of size n, where k is the beam width and d is the maximum depth of search space, i.e. the iteration number of outer while loop. □

Although the time complexity of *ShortL-BiLS* is $O(kdn^3)$, but at the beginning, all possible blocking pairs are removed by the Gale-Shapley algorithm to obtain the men's and women's shortlists, which are very sparse matrices (e.g., see Table 4). Therefore, the use of the shortlists in the breakmarriage operation to produce a stable neighbour matching will significantly speed up the execution time of *ShortL-BiLS*.

Performance evaluation

This section 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 *ShortL-BiLS* algorithm, we randomly produced 240 SM instances of 12 difference sizes from 50 to 600 with step 50 and 20 variants per size.[1]

Solution quality evaluation

ShortL-BiLS is an approximation algorithm for finding the egalitarian or sex-equal matching of the SM problem. Therefore, it is necessary to evaluate the solution quality obtained by *ShortL-BiLS*. We evaluated the solution quality based on the percentage of exact solutions and the relative accuracy of algorithms.

First, we performed experiments to compare the percentage of exact solutions found by *ShortL-BiLS* with that found by *BiLS* (Viet et al., 2016a) and *SLS* (Viet et al., 2016b) algorithms in order to identify which algorithm gives the best result among all them. To do this, we compared the solutions found by the algorithms with the exact solutions found by the *BFS* algorithm (i.e., Algorithm 1) for the SM instances. Tables 5 and 6 show the experimental results. Specifically, the percentage of the egalitarian matchings as well as that of the sex-equal matchings increases, respectively, with *SLS, BiLS* and *ShortL-BiLS* algorithms. However, for *ShortL-BiLS*, the percentage is almost the same when $k = 3$ or $k = 4$ for both finding the egalitarian and sex-equal matchings. Besides, even when $k = 1$, the percentage of exact solutions found by *ShortL-BiLS* is much higher than that found by *BiLS* and *SLS* algorithms. In briefly, *ShortL-BiLS* significantly outperforms *BiLS* and *SLS* algorithms in terms of finding exact solutions.

Next, we performed experiments to compare the *relative accuracy* of solutions found by *ShortL-BiLS, BiLS* and *SLS* algorithms. The *relative accuracy* of an algorithm A is defined by $max\{(max(I) - opt(I))/(max(I) - A(I))\}$ overall instances I, where $opt(I)$, $max(I)$, and $A(I)$ are the costs of the optimal solution, the worst solution, and the algorithm's solution, respectively (Charikar & Wirth, 2004). For each SM instance, let M_0 and M_t be the man- and woman-optimal matchings, respectively, found by the Gale-

Table 5. The percentage of the egalitarian matchings found by *ShortL-BiLS, BiLS* and *SLS* algorithms, where k is the number of the best successors in the stable neighbour matching set.

		ShortL-BiLS (%)					
Data Set	Size	$k = 1$	$k = 2$	$k = 3$	$k = 4$	*BiIS (%)*	*SLS (%)*
1	50	100	100	100	100	95	80
2	100	85	95	100	100	75	70
3	150	70	90	90	90	70	50
4	200	70	75	75	75	70	50
5	250	60	85	85	85	60	50
6	300	70	85	85	85	65	60
7	350	50	80	95	95	45	40
8	400	80	95	95	95	65	35
9	450	30	65	70	70	30	25
10	500	80	90	90	90	75	65
11	550	65	75	80	80	65	50
12	600	45	65	70	70	40	25
Average		**67.08**	**83.33**	**86.25**	**86.25**	**62.91**	**50.00**

Shapley algorithm; let M_e^{opt} and M_s^{opt} be the egalitarian and sex-equal matchings, respectively, found by the *BFS* algorithm. The *relative accuracy* of an algorithm A for finding the egalitarian and sex-equal matchings, respectively, of each SM instance is

$$\max\left\{\frac{\max(c(M_0), c(M_t)) - c(M_e^{opt})}{\max(c(M_0), c(M_t)) - c(M_e)}\right\} \text{and} \max\left\{\frac{\max(d(M_0), d(M_t)) - d(M_s^{opt})}{\max(d(M_0), d(M_t)) - d(M_s)}\right\}, \tag{8}$$

where M_e and M_s are solutions found by the algorithm A for finding the egalitarian and sex-equal matchings, respectively. Obviously, if the *relative accuracy* of an algorithm is close to one, the solution obtained by the algorithm is near to the optimal solution. For 240 SM instances in our experiments, the *relative accuracy* of *ShortL-BiLS, BiLS* and *SLS* algorithms for finding the egalitarian matching is determined to be 1.0019, 1.0071 and 1.0264, respectively. While the *relative accuracy* of *ShortL-BiLS, BiLS* and *SLS* algorithms for finding the sex-equal matching is 1.0081, 1.0182, and 1.0461, respectively. This means that *ShortL-BiLS, BiLS* and *SLS* algorithms can obtained approximate solutions close to exact solutions of the SM instances, but *ShortL-BiLS* gives the best solution among all them.

The main observations regarding the experimental results can be explained as follows. *SLS* algorithm only performs one unidirectional local search from the man- to woman-optimal matching, i.e., the searching range is smallest and therefore, it gives the lowest percentage of exact solutions. By performing two simultaneous searches, the searching range of *BiLS* is larger than that of *SLS* and therefore, it gives the higher percentage of exact solutions compared with *SLS*. Also, performing two simultaneous searches, but the searching range of *ShortL-BiLS* is more expanded when k increases. This provides an opportunity to achieve better solutions compared with *BiLS* and *SLS* algorithms.

Execution time evaluation

In this section, we compared the execution time of the *ShortL-BiLS* with that of the *BiLS, SLS*, and *BFS* algorithms. Figure 3 shows the average execution time of the algorithms for finding the egalitarian and sex-equal matchings of the SM instances of the same size. Observations of the experimental results can be summarized as follows:

(1) For all of the algorithms, the average execution time for finding the egalitarian matching is almost the same as that for finding the sex-equal matching. This is because the search mechanism for the egalitarian matching is the same as that for the sex-equal matching, except for the evaluation functions.
(2) The execution time of *ShortL-BiLS* with $k = 1$ is smaller than that of *BiLS*. For all of the SM instances, *ShortL-BiLS* runs about 1.2 times faster than *BiLS*. For example, when the size of

Table 6. The percentage of the sex-equal matchings found by *ShortL-BiLS*, *BiLS*, and *SLS* algorithms, where k is the number of the best successors in the stable neighbour matching set.

Data Set	Size	*ShortL-BiLS (%)*				*BilS (%)*	*SLS (%)*
		$k = 1$	$k = 2$	$k = 3$	$k = 4$		
1	50	95	95	95	95	95	75
2	100	85	90	100	100	85	45
3	150	90	100	100	100	85	65
4	200	50	70	75	80	50	45
5	250	70	85	85	85	70	50
6	300	65	75	80	80	65	50
7	350	70	95	90	90	70	30
8	400	65	90	95	95	65	35
9	450	45	65	65	65	40	30
10	500	60	85	85	85	60	50
11	550	65	70	75	75	60	45
12	600	75	80	85	85	75	35
Average		**69.58**	**83.33**	**85.83**	**86.25**	**68.33**	**46.25**

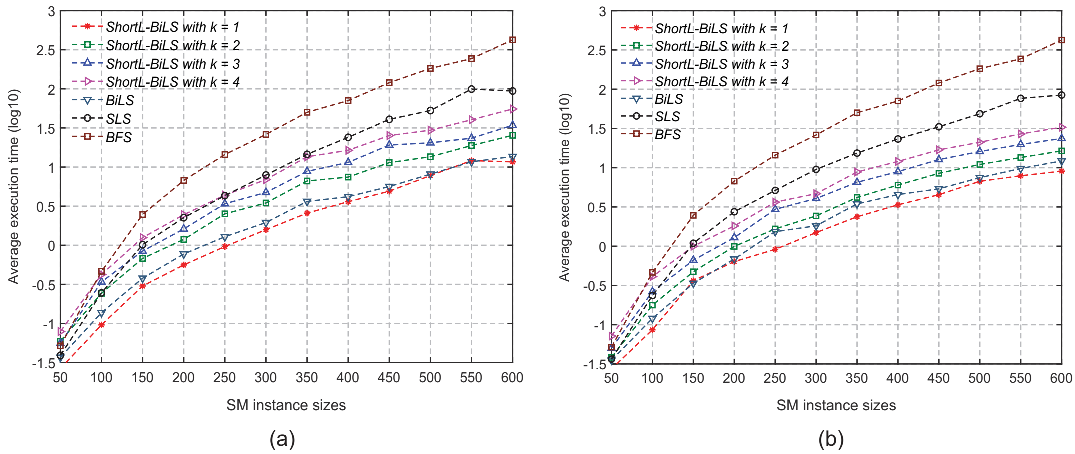

Figure 3. The average execution time of the *ShortL-BiLS, BiLS, SLS* and *BFS* algorithms for finding (a) egalitarian and (b) sex-equal matching of the SM instances.

the SM instances is 350, the average execution time of *BiLS* and *ShortL-BiLS* for finding the egalitarian matching is about $10^{0.56} = 3.63$ and $10^{0.41} = 2.57$ seconds, i.e., *ShortL-BiLS* runs about 1.41 times faster than *BiLS*. This is because the *ShortL-BreakM* function used in *ShortL-BiLS* ignores checking the women who do not appear in the men's shortlists and vice versa when it constructs a stable matching.

(3) The execution time of *ShortL-BiLS* increases when k increases. This is because the more k increases, the more neighbours to be searched increases, however, *ShortL-BiLS* has many opportunities to achieve better solutions.

(4) The execution time of *ShortL-BiLS* with $k = 4$ is much smaller than that of *SLS*. For all of the SM instances, *ShortL-BiLS* runs about 1.75 times faster than *SLS*. Figure 3 also shows that *ShortL-BiLS* is efficient compared to *SLS* for large SM problems. For example, when the size of the SM instances is 550, the average execution time of *SLS* and *ShortL-BiLS* for finding the egalitarian matching is about $10^{2.00} = 100$ and $10^{1.60} = 39.81$ seconds, i.e., *ShortL-BiLS* runs about 2.51 times faster than *SLS*. This is because *ShortL-BiLS* simultaneously performs two searches 'moving' to the other by means of the man cost. In particular, both *ShortL-BiLS* and *SLS* try to seek a best solution in the range between the man- and woman-optimal matchings. While *ShortL-BiLS* performs searching in two directions (from the man- to woman-optimal matching and vice versa) and stops when its searches meet or pass each other in the middle, *SLS* only search from the man- to woman-optimal matching.

(5) For large SM problems, the execution time of *ShortL-BiLS* with $k = 4$ is much smaller than that of *BFS*. For example, when the size of the SM instances is 600, the average execution time of *BFS* and *ShortL-BiLS* for finding the egalitarian matching is about $10^{2.62} = 416.86$ and $10^{1.74} = 54.95$ seconds, i.e., *ShortL-BiLS* runs about 8 times faster than *BFS*. This is because *BFS* algorithm is an exhaustive search, it has to explore all the search space to find solutions of the SM instances.

We would like to note that we compared our *ShortL-BiLS* with the *SML* algorithm (Gelain et al., 2013) and ACS algorithm (Vien et al., 2007) in terms of both the execution time and solution quality. Our experiments showed that our *ShortL-BiLS* runs much faster than the *SML* and the ACS algorithms even when the size of SM instances is 50.

Table 7. Experimental results for large SM instances.

Instance size	k	d	Runtime (sec.)	$d(M_{best})$	$d(M_0)$	$d(M_t)$
1000	1	7	60.76	134	121,785	113,674
	2	7	101.94	134		
	3	7	145.91	134		
	4	7	215.02	133		
1500	1	7	262.00	355	238,799	300,777
	2	7	470.69	355		
	3	7	681.31	355		
	4	8	993.81	355		
2000	1	7	664.63	423	496,068	582,589
	2	7	1228.90	423		
	3	7	1733.37	423		
	4	7	2465.47	423		

Experiment for large instances

In this section, we ran experiments to find the sex-equal matching of large SM instances. Specifically, we randomly produced 3 SM instances of size 1000, 1500 and 2000. For each SM instance, we ran *ShortL-BiLS* algorithm with k = 1, 2, 3 and 4 to determine the maximum depth, d, of search space, the runtime, and the sex-equal cost $d(M_{best})$, where M_{best} is the solution found by the algorithm. Table 7 shows the experimental results, where M_0 and $d(M_0)$ are the man-optimal matching and its sex-equal cost, respectively, and M_t and $d(M_t)$ are the woman-optimal matching and its sex-equal cost, respectively, of the SM instances.

The experimental results show that for all of the SM instances, the maximum depth of searching space is $d = 8$. The runtime for all instance sizes is reasonable. When k is increased, the runtime increases. This is because of the larger searching range of *ShortL-BiLS*. However, $d(M_{best})$ of each SM instance probably keeps the same, excepting for $k = 4$ with SM instance size of 1000. This indicates the consistency of *ShortL-BiLS*. Moreover, $d(M_{best})$ is too small compared to $d(M_0)$ and $d(M_t)$, meaning that the solution M_{best} found by the algorithm is the optimal or near optimal solution.

Conclusions

In this paper, we proposed a *ShortL-BiLS* algorithm to find an approximate solution in terms of the egalitarian or sex-equal matching of the SM problem. *ShortL-BiLS* is a bidirectional local search, in which the forward local search finds a solution while moving towards the woman-optimal matching and the backward local search finds a solution while moving towards the man-optimal matching. *ShortL-BiLS* interleaves iterations of the forward and backward searches until their search frontiers meet each other. When the algorithm ends, the solution is the best one of the solutions found by the forward and backward searches. The experiments show that *ShortL-BiLS* is efficient in terms of the execution time and solution quality for the SM problem. In the future, we plan to extend the proposed approach to a wide range of matching problems such as the stable marriage with ties and incomplete lists (Manlove, 1999, July; Trang, Viet, & Chung, 2016) or the roommate problem (Fleiner et al., 2007; Irving, 1985).

Endnote

1. Source codes are available at the URL: https://github.com/vietjho/ShortL-BiLS.

Acknowledgments

This research is funded by the Basic Science Research Program through the National Research Foundation of Korea under grant number NRF-2017R1D1A1B04036354 and the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2017.09.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea [NRF-2017R1D1A1B04036354]; Vietnam National Foundation for Science and Technology Development (NAFOSTED) [102.01-2017.09].

References

Abraham, D. J., Irving, R. W., & Manlove, D. F. (2003, December). The student-project allocation problem. In *Proceedings of the 14th international symposium on Algorithms and Computation* (pp. 474–484). Kyoto, Japan.

Charikar, M., & Wirth, A. (2004, October). Maximizing quadratic programs: Extending grothendieck's inequality. In *Proceedings of 45th annual ieee symposium on foundations of computer science* (pp. 54–60). Rome, Italy.

Everaere, P., Morge, M., & Picard, G. (2013, May). 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)* (pp. 1319–1320). St. Paul, MN, USA.

Fleiner, T., Irving, R. W., & Manlove, D. F. (2007). Efficient algorithms for generalized stable marriage and roommates problems. *Theoretical Computer Science*, *381*(1–3), 162–176.

Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, *9*(1), 9–15.

Gelain, M., Pini, M. S., Rossi, F., Venable, K. B., & Walsh, T. (2013). Local search approaches in stable matching problems. *Algorithms*, *6*(1), 591–617.

Giannakopoulos, I., Karras, P., Tsoumakos, D., Doka, K., & Koziris, N. (2015, November). An equitable solution to the stable marriage problem. In *2015 ieee 27th international conference on tools with artificial intelligence (ictai)* (pp. 989–996). Vietri sul Mare, Italy.

Gusfield, D. (1987). Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, *16*(1), 111–128.

Irving, R. W. (1985). An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms*, *6*(1), 577–595.

Irving, R. W., Leather, P., & Gusfield, D. (1987). An efficient algorithm for the "optimal" stable marriage. *Journal of the Association for Computing Machinery*, *34*(3), 532–543.

Iwama, K., Miyazaki, S., Morita, Y., & Manlove, D. (1999), July. Stable marriage with incomplete lists and ties. In Proceedings of the 26th International Colloquium on Automata, Languages, and Programming (p. 443-452). Prague: Czech Republic.

Iwama, K., Miyazaki, S., & Yanagisawa, H. (2010). Approximation algorithms for the sex-equal stable marriage problem. *ACM Transactions on Algorithms*, *7*(1), 2: 1–2:17.

McVitie, D. G., & Wilson, L. B. (1971). The stable marriage problem. *Communication of ACM*, *14*(7), 486–490.

Nakamura, M., Onaga, K., Kyan, S., & Silva, M. (1995, Apr.). Genetic algorithm for sex-fair stable marriage problem. In Proceedings of the International Symposium on Circuits and Systems (ISCAS'95) (p. 509-512), WA, USA: Seattle.

Roth, A. E. (1984). The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Pilitical Economy*, *92*(6), 991–1016.

Russel, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3nd ed.). UpperSaddle River, NJ, USA: Prentice Hall Press.

Trang, L. H., Viet, H. H., & Chung, T. (2016, October). Finding "optimal" stable marriages with ties via local search. In *Proceedings of the 2016 eighth international conference on knowledge and systems engineering (kse2016)* (pp. 61–66). Hanoi, Vietnam.

Vien, N. A., Viet, N. H., Kim, H., Lee, S., & Chung, T. (2007). Ant colony based algorithm for stable marriage problem. In K. Elleithy (Ed.), *Advances and innovations in systems, computing sciences and software engineering* (pp. 457–461). Dordrecht: Springer Netherlands.

Viet, H. H., Trang, L. H., Lee, S. G., & Chung, T. C. (2016a, November). A bidirectional local search for the stable marriage problem. In *Proceedings of the 2016 international conference on advanced computing and applications (acomp)* (pp. 18–24). Can Tho City, Vietnam.

Viet, H. H., Trang, L. H., Lee, S. G., & Chung, T. C. (2016b, August). An empirical local search for the stable marriage problem. In *Proceedings of the 14th pacific rim international conference on artificial intelligence - pricai 2016: Trends in artificial intelligence* (pp. 556–564). Phuket, Thailand.

Zavidovique, B., Suvonvorn, N., & Seetharaman, G. (2005, Sep.). 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)* (pp. 63–70). Barcelona, Spain.