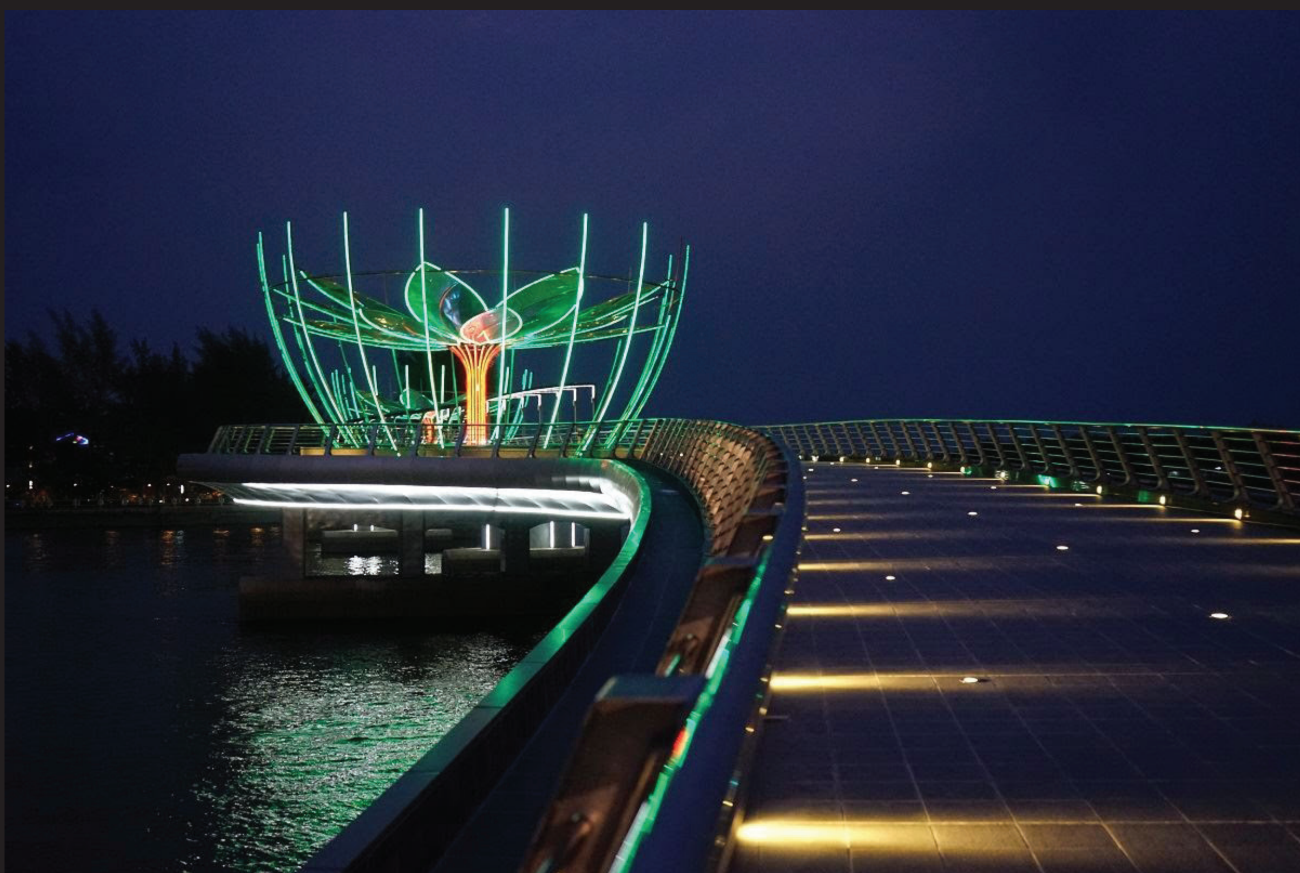
ACOMP 2016

2016 International Conference on Advanced Computing and Applications

23-25 November 2016, Can Tho City, Vietnam

Edited by Lam-Son Lê, Tran Khanh Dang, Josef Küng, Nam Thoai, and Roland Wagner



10/14/2016 12:04:39 PM

Proceedings

2016 International Conference on Advanced Computing and Applications

ACOMP 2016

23–25 November 2016
Can Tho City, Vietnam

Proceedings

2016 International Conference on Advanced Computing and Applications

ACOMP 2016

23–25 November 2016
Can Tho City, Vietnam

Editors
Lam-Son Lê
Tran Khanh Dang
Josef Küng
Nam Thoai
Roland Wagner

CPS Conference Publishing Services

Copyright © 2016 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved.

Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries may photocopy beyond the limits of US copyright law, for private use of patrons, those articles in this volume that carry a code at the bottom of the first page, provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Other copying, reprint, or republication requests should be addressed to: IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 133, Piscataway, NJ 08855-1331.

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society, or the Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Order Number P5978
ISBN-13: 978-1-5090-6143-3
BMS Part # CFP16E01-PRT

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: <reprints@computer.org>

Editorial production by Lisa O'Conner
Cover art production by Mark Bartosik

CPS
Conference Publishing Services

IEEE Computer Society
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

2016 International Conference on Advanced Computing and Applications

ACOMP 2016

Table of Contents

2016 International Conference on Advanced Computing and Applications

A bidirectional local search for the stable marriage problem

Hoang Huu Viet
Information Technology Dept.,Vinh University
182 LeDuan Str., Vinh City, NgheAn, VietNam
Email: viethh@vinhuni.edu.vn (✉)

Le Hong Trang
Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology
268 Ly Thuong Kiet, Ho Chi Minh City, Vietnam
Email: lhtrang@vinhuni.edu.vn

SeungGwan Lee
Humanitas College, Kyung Hee University
1732, Deogyeong-daero, Giheung-gu, Yongin-si,
Gyeonggi-do, 446-701, South Korea
Email: leesg@khu.ac.kr

TaeChoong Chung
Computer Engineering Dept., Kyung Hee University
1732, Deogyeong-daero, Giheung-gu, Yongin-si,
Gyeonggi-do, 446-701, South Korea
Email: tcchung@khu.ac.kr

***Abstract*—This paper proposes a bidirectional local search algorithm to find the *egalitarian* and the *sex-equal* stable matchings in the stable marriage problem. Our approach simultaneously searches forward from the *man-optimal* stable matching and backwards from the *woman-optimal* stable matching until the search frontiers meet. By employing a breakmarriage strategy to find stable neighbor matchings of the current stable matching and moving to the best neighbor matching, the forward local search finds the solutions while moving towards the *woman-optimal* stable matching and the backward local search finds the solutions while moving towards the *man-optimal* stable matching. Simulations show that our proposed algorithm is efficient for the stable marriage problem.**

***Keywords*-break marriage; gale-shapley algorithm; local search; stable marriage problem;**

I. Introduction

The stable marriage (SM) problem is a well-known problem of matching an equal number men and women to satisfy a certain criterion of stability. This problem was first introduced by D. Gale et al. [1], 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 [2], the Student-Project Allocation problem (SPA)[3] and the Stable Roommates problem (SR) [4], [5].

An instance of SM of size n, denoted by I, 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 (PL). A matching M is a set of n disjoint pairs of men and women. If a man m and a woman w form a pair in M, then m and w are 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 that has no any blocking pairs is said to be stable, otherwise it is said to be unstable. Let $\mathcal{M}$ denote a set of all stable matchings, $mr(m, w)$ denote the rank of woman w in man m's PL, and $wr(w, m)$ denote the rank of man m in woman w's PL. For a stable matching $M \in \mathcal{M}$, we define the *man cost* $sm(M)$ and the *woman cost* $sw(M)$ as follows:

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

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

Definition 1 (man-optimal and woman-optimal [6])**.** A stable matching M is called *man-optimal* (respectively *woman-optimal*) if it has the minimum value of $sm(M)$ (respectively $sw(M)$) 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 time $O(n^2)$[1]. The Gale-Shapley algorithm is basically a sequence of proposals from men to women to find the *man-optimal* stable matching. If the roles of men and women are interchanged, the matching found by the Gale-Shapley is the *woman-optimal* stable matching. It is proved that in the *man-optimal* stable matching, each woman has the worst partner that she can have in any stable matching and that in the *woman-optimal* stable matching, each man has the worst partner that he can have in any stable matching [7]. For an instance of SM of size n, there may be many other stable matchings between the *man-optimal* and *woman-optimal* stable matchings in general. Moreover, the *man-optimal* (respectively *woman-optimal*) stable matching is the "*selfish*" matching for men (respectively women) and therefore, it is appropriate to seek other optimal stable matchings such as an *egalitarian* or *sex-equal* stable matching to give more balanced preference for both men and women. For a stable matching $M \in \mathcal{M}$, we define the *egalitarian cost* $c(M)$ and the *sex-equality cost* $d(M)$ as follows:

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

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

978-1-5090-6144-0/16 $31.00 © 2016 IEEE
DOI 10.1109/ACOMP.2016.17

Definition 2 (egalitarian and sex-equal [6]). A stable matching M is called *egalitarian* (respectively *sex-equal*) if it has the minimum value of $c(M)$ (respectively $d(M)$) for all $M \in \mathcal{M}$.

There are several approaches to find the *egalitarian* or *sex-equal* stable matching. Nakamura et al. [6] proposed a genetic algorithm (GA) for *sex-fair* (i.e., *sex-equal*) SM problem. In their approach, the problem is transferred into a directed graph problem and GA is used to find the solution in the graph. Vien et al. [8] presented an ant colony system (ACS) algorithm for SM instances. Based on the heuristic functions defined for optimal stable matching criteria, a set of cooperating agents, called artificial ants, cooperates to find *man-optimal*, *woman-optimal*, *egalitarian* and *sex-equal* stable matchings. Iwama et al. [9] proposed an approximation algorithm for finding a near optimal solution in terms of *sex-equal* or *egalitarian* stable matching. Because the number of stable matchings of SM instances grows exponentially in general [10], the above approaches are inefficient for finding solutions of SM instances of large sizes. To find rapidly a solution of SM instance of a large size, an approach that may be useful is local search. In fact, Gelain et al. [11] recently have proposed local search approaches in stable matching problems that aim to accelerate the finding process of solutions. However, their approach finds only one stable matching of a given SM.

In this paper, we present a bidirectional local search (BiLS) algorithm to seek an *egalitarian* or *sex-equal* stable matching of SM instances. We aim to give a general approach, which is able to be applied to SM instances for some optimal criterion. BiLS algorithm runs two simultaneous local searches: one forward from the man-optimal and the other backward from the woman-optimal, stoping when the both stop and meet. For each local search, the Gale-Shapley algorithm [1] is used to find the initial state, the breakmarriage operation [12] is used to generate stable neighbor matchings of the current stable matching and the random walk is used to avoid getting stuck in a local optimum. The simulation results show that our approach is efficient in terms of computational time and solution quality for SM problem.

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

II. Background

For illustrative purposes, we consider an instance of SM consisting of eight men and eight women with their preference lists shown in Table I.

TABLE I
PREFERENCE LISTS OF EIGHT MEN AND 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

Gale-Shapley algorithm. The Gale-Shapley algorithm [1] finds the *man-optimal* stable matching of SM instances. At the beginning, the algorithm assigns each person to be free. At each iteration step, the algorithm chooses a free man m and finds the most preferred woman w in m's preference list to whom m has not proposed. If w is free then w and m become engaged. If w is engaged to m' then she rejects the man that she least prefers to engage to the other man. The rejected man becomes free. The algorithm terminates when all men are engaged. If the roles of men and women are interchanged, the matching found by the Gale-Shapley is the *woman-optimal* stable matching. For example, the *man-* and *woman-optimal* stable matchings of Table I found by the Gale-Shapley algorithm are $M_0 = \{(1,4),(2,3),(3,8),(4,5),(5,1),(6,6),(7,2),(8,7)\}$ and $M_t = \{(1,1),(2,4),(3,7),(4,8),(5,3),(6,5),(7,6),(8,2)\}$, respectively.

Breakmarriage operation. Let M be a stable matching and (m, w) be an engaged pair in M. The breakmarriage operation [12], denoted by BreakMarriage(M, m), finds a stable matching from M and m. The idea of the breakmarriage operation is similar to the Gale-Shapley algorithm. At the beginning, the algorithm assigns the woman w to the partner of the man m and sets m to be free. At each iteration step, the algorithm performs a sequence of proposals, rejects and acceptances as those of the Gale-Shapley algorithm. The algorithm terminates when either some man has been rejected by all women or the woman w accepts a man m' to whom she prefers to her partner. Because there is exactly one free man at any time of algorithm execution, if the woman w accepts a man m' then there are no free men and BreakMarriage(M, m) returns a stable matching M' of n engaged pairs. For example, BreakMarriage($M_0, 1$) returns the stable matching $M' = \{(1,4),(2,3),(3,2),(4,5),(5,1),(6,6),(7,8),(8,7)\}$ on man 3, while BreakMarriage($M_0, 2$) returns no stable matching.

Finding all the stable matchings. If the breakmarriage operation is applied for each man m in the men set, it would generate in general the same stable stable matching being obtained many times. In order to find all the stable matchings that every stable matching is produced once and only once, McVitie and Wilson proposed an algorithm [12] by imposing two restriction rules on the breakmarriage operation as follows:

- $R1$: If BreakMarriage(M, m) returns M' on man m' then BreakMarriage(M', i) only be performed on men $i \geq m'$.
- $R2$: In BreakMarriage(M, m) only men $m' \geq m$ propose, i.e., if some man m' is free and $m' < m$ during the execution time then BreakMarriage(M, m) is stopped and said to be unsuccessful.

The breakmarriage operation with the above two restriction rules leads to the following important theorem:

Theorem 1 ([12]). *Every stable matching $M_i (i = 1, 2, \cdots, t)$ can be obtained by a series of breakmarriage operations starting from the man-optimal stable matching M_0, where M_t is the woman-optimal stable matching.*

For a given SM instance, a dominance relation on the set of stable matchings is defined as follows:

Definition 3 (Dominance [13]). Let $M \in \mathcal{M}$ and $M' \in \mathcal{M}$ be two stable matchings. M is said to dominate M' under the men's point of view if and only if every man prefers his partner in M at least as well as to his partner in M'.

Corollary 1 ([13]). *If* BREAKMARRIAGE(M, m) *results in a stable matching M', then M' dominates all stable matchings which are dominated by M and in which m is not married to his mate in M.*

Breadth-first search algorithm. The algorithm for finding all the stable matchings [12] is an exhaustive search method. Therefore, to find the *egalitarian* and the *sex-equal* stable matchings in the stable marriage problem, a breadth-first search (BFS) algorithm based on the algorithm [12] can be described as Algorithm 1. At the beginning, the algorithm assigns the best solution M_{best} to the man-optimal M_0 found by the Gale-Shapley algorithm, denoted by GALESHAPLEY(*I,Men*) (*i.e.*, a sequence of proposals from men to women), and assigns the parent set to M_0. At each iteration step, the algorithm finds a child set of matchings M in the parent set by performing a sequence of BREAKMARRIAGE(M, m) for each man m from i to n, where $i = brokenMan(M)$ is the man corresponding to M obtained by some breakmarriage operation (*i.e.*, apply the rule 1). If there exist no children, the algorithm ends. Otherwise, the algorithm evaluates all of the matchings in the child set using a cost function $f(M)$, which is the *egalitarian cost* in (3) (respectively *sex-equal cost* in (4)) for finding the *egalitarian* (respectively *sex-equal*) stable matching, and then selects the best child M_{best_child} to be a matching with the smallest value of the cost function. If the best solution found so far is worse than the best child, the best solution is assigned to the best child solution. Finally, the algorithm assigns the parent set to the child set and repeats until it meets the ending condition.

By using BFS algorithm, all stable matchings of Table I are shown in Table II. Specifically, M_0 is the *man-optimal* stable matching, M_{17} is the *woman-optimal* stable matching, M_9 is the *egalitarian* stable matching and M_4 is the *sex-equal* stable matching. Figure 1 shows a tree-like structure found by BFS algorithm. The man-optimal stable matching is the top of the tree, each branch $M - M'$ shows that M' is obtained by BREAKMARRIAGE(M, m) on man m', where m' is labeled on the branch. Because BFS algorithm is an exhaustive search method, it always finds exactly solutions of SM instances. However, the number of stable matchings of SM instances grows exponentially in general [10] and therefore, BFS algorithm is only efficient when the size of SM instances is small.

Algorithm 1: BFS Algorithm

Input : an instance I of SM
Output: the best matching and all stable matchings

M_0 := GALESHAPLEY(*I,Men*); ▷ *men* propose *women*;
$M_{best} := M_0$;
$parentSet := M_0$;
$stableSet := M_0$;
$brokenMen(M_0) := 1$; ▷ used for the rule $R1$;
while *(true)* **do**
  $childSet := \emptyset$;
  for *(each matching M in parentSet)* **do**
    for $m := brokenMen(M)$ **to** n **do**
      $[M_{child}, m']$:= BREAKMARRIAGE(M, m);
      if *($M_{child} \neq \emptyset$)* **then**
        $childSet := childSet \cup M_{child}$;
        $brokenMen(M_{child}) := m'$;
        $stableSet := stableSet \cup M_{child}$;
      end
    end
  end
  if *($childSet = \emptyset$)* **then**
    break;
  end
  $M_{best\_child} := \arg\min_{M \in childSet}(f(M))$;
  if $f(M_{best}) > f(M_{best\_child})$ **then**
    $M_{best} := M_{best\_child}$;
  end
  $parentSet := childSet$;
end
print M_{best} and $stableSet$;

TABLE II
EVALUATIONS OF STABLE MATCHINGS

Stable matchings	sm	sw	c	d
M_0 = {(1,4),(2,3),(3,8),(4,5),(5,1),(6,6),(7,2),(8,7)}	12	35	47	23
M_1 = {(1,3),(2,4),(3,8),(4,5),(5,1),(6,6),(7,2),(8,7)}	15	27	42	12
M_2 = {(1,4),(2,3),(3,2),(4,5),(5,1),(6,6),(7,8),(8,7)}	15	32	47	17
M_3 = {(1,4),(2,3),(3,8),(4,6),(5,1),(6,5),(7,2),(8,7)}	15	30	45	15
M_4 = {(1,1),(2,4),(3,2),(4,5),(5,3),(6,6),(7,8),(8,7)}	21	20	41	1
M_5 = {(1,3),(2,4),(3,2),(4,5),(5,1),(6,6),(7,8),(8,7)}	18	24	42	6
M_6 = {(1,3),(2,4),(3,8),(4,6),(5,1),(6,5),(7,2),(8,7)}	18	22	40	4
M_7 = {(1,4),(2,3),(3,2),(4,6),(5,1),(6,5),(7,8),(8,7)}	18	27	45	9
M_8 = {(1,1),(2,4),(3,7),(4,5),(5,3),(6,6),(7,8),(8,2)}	28	15	43	13
M_9 = {(1,1),(2,4),(3,2),(4,6),(5,3),(6,5),(7,8),(8,7)}	24	15	39	9
M_{10} = {(1,3),(2,4),(3,7),(4,5),(5,1),(6,6),(7,8),(8,2)}	25	19	44	6
M_{11} = {(1,3),(2,4),(3,2),(4,6),(5,1),(6,5),(7,8),(8,7)}	21	19	40	2
M_{12} = {(1,4),(2,3),(3,2),(4,8),(5,1),(6,5),(7,6),(8,7)}	21	25	46	4
M_{13} = {(1,1),(2,4),(3,7),(4,6),(5,3),(6,5),(7,8),(8,2)}	31	10	41	21
M_{14} = {(1,1),(2,4),(3,2),(4,8),(5,3),(6,5),(7,6),(8,7)}	27	13	40	14
M_{15} = {(1,3),(2,4),(3,7),(4,6),(5,1),(6,5),(7,8),(8,2)}	28	14	42	14
M_{16} = {(1,3),(2,4),(3,2),(4,8),(5,1),(6,5),(7,6),(8,7)}	24	17	41	7
M_{17} = {(1,1),(2,4),(3,7),(4,8),(5,3),(6,5),(7,6),(8,2)}	34	8	42	26
M_{18} = {(1,3),(2,4),(3,7),(4,8),(5,1),(6,5),(7,6),(8,2)}	31	12	43	19

•M_0: *man-optimal*, M_{17}: *woman-optimal*.
•M_9: *egalitarian*, M_4: *sex-equal*.

III. PROPOSED ALGORITHM

Local search algorithms are among the popular methods for solving optimization problems because of two key advantages:

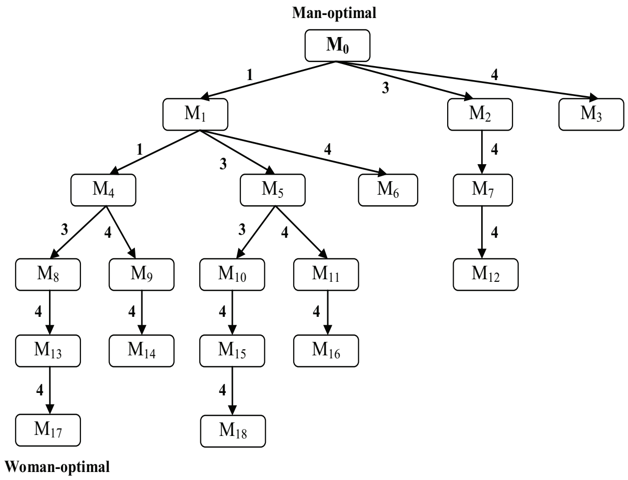


Fig. 1. The tree-like structure generated by the BFS algorithm for Table I

(*i*) they take very little memory; and (*ii*) they find quickly reasonable solutions in large or infinite state spaces. However, the local search algorithms often fail to find a global optimal solution when one exists because they can get stuck on a local optimum solution. To avoid this disadvantage, many of local search algorithms such as random-restart hill climbing, simulated annealing or local beam search have been proposed [14]. Basically, a classical local search algorithm starts from a given solution and tries to find a better one in the neighbors of the solution. If a better solution is found, the current solution is moved to the better one and the local search is repeated for the current solution. Such a local search can be considered as a unidirectional search. However, a unidirectional search can be inefficient for a problem of large size.

Since Theorem 1 and obviously if the roles of men and women are interchanged, then every stable matching can be obtained by a series of breakmarriage operations starting from the woman-optimal stable matching. Therefore, we propose a bidirectional local search (BiLS) algorithm to find an *egalitarian* or *sex-equal* stable matching of SM of size n. BiLS runs two simultaneous searches: one forward from the man-optimal and the other backward from the woman-optimal one. The framework of BiLS is shown in Algorithm 2. At the beginning, Procedure GALESHAPLEY is called to find, respectively, the man- and woman-optimal matchings which are starting solutions for the bidirectional search. At each iteration, for one of two searching directions, the algorithm finds a neighbor set of the current solution, which are M_{left} or M_{right}, by calling Procedure BREAKMARRIAGE(M_{left}, m) (respectively BREAKMARRIAGE(M_{right}, w)) for each man (respectively woman) in turn in the men (respectively women) set. The algorithm evaluates all stable neighbor matchings in the neighbor set using the cost function $f(M)$ as defined in Algorithm 1. The algorithm then selects the next solution to be a neighbor whose smallest value of the cost function. The algorithm can also selects the next solution to be a random neighbor with a small probability in order to avoid getting stuck in a local optimum. If the next solution of

Algorithm 2: BiLS Algorithm

Input : an instance I of SM
Output: a stable matching

M_{left} := GALESHAPLEY(*I,Men*); ▷ *men* propose *women*;
M_{right} := GALESHAPLEY(*I,Women*); ▷ *women* propose *men*;
if $(f(M_{left}) < f(M_{right}))$ **then**
  $M_{best} := M_{left}$;
else
  $M_{best} := M_{right}$;
end
$forward := true$;
$backward := true$;
while $(true)$ **do**
  if $(forward)$ **then**
    $neighborSet := \emptyset$;
    for *(each man m in the men set)* **do**
      $stableMatching$:= BREAKMARRIAGE(M_{left}, m);
      $neighborSet := neighborSet \cup stableMatching$;
    end
    if *(small random probability p)* **then**
      M_{next} := a random matching in $neighborSet$;
    else
      $M_{next} := \arg\min_{M \in neighborSet}(f(M))$;
    end
    if $(f(M_{next}) > f(M_{left}))$ **then**
      $forward := false$;
      if $f(M_{best}) > f(M_{left})$ **then**
        $M_{best} := M_{left}$;
      end
    end
    $M_{left} := M_{next}$;
  end
  if $(backward)$ **then**
    $neighborSet := \emptyset$;
    for *(each woman w in the women set)* **do**
      $stableMatching$:= BREAKMARRIAGE(M_{right}, w);
      $neighborSet := neighborSet \cup stableMatching$;
    end
    if *(small random probability p)* **then**
      M_{next} := a random matching in $neighborSet$;
    else
      $M_{next} := \arg\min_{M \in neighborSet}(f(M))$;
    end
    if $(f(M_{next}) > f(M_{right}))$ **then**
      $backward := false$;
      if $f(M_{best}) > f(M_{right})$ **then**
        $M_{best} := M_{right}$;
      end
    end
    $M_{right} := M_{next}$;
  end
  if *((not forward) and (not backward))* **then**
    if $(sm(M_{left}) \leq sm(M_{right}))$ **then**
      $forward := true$;
      $backward := true$;
    else
      break;
    end
  end
end
return M_{best};

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, the best one is assigned to the current one. The algorithm then moves the current solution to the next one and repeats the iteration. The algorithm terminates when either one of searching directions has no neighbors or two searching directions meet each other by means of the man cost. In particular, if both forward and backwards searches are pausing and the man cost of the current matching of the forward search $sm(M_{left})$ is equal or greater than that of the backwards one $sm(M_{right})$, then the bidirectional search is completed. The algorithm thus stops and gives the best solution so far. It should be noted that in the local search approach, for each stable matching, we must find all stable neighbor matchings instead of only one stable matching as in Algorithm 1. To do so, only the restriction rule $R2$ is applied to the breakmarriage operation.

An illustration of Algorithm 2 to find a *sex-equal* stable matching for the SM in Table I is depicted in Figure 2, in which the probability to move the solution to a random neighbor is set to be zero. Initially, the algorithm assigns M_{left} to the man-optimal M_0 and assigns M_{right} to the woman-optimal M_{17}. At the first iteration, the algorithm finds a better solution in the neighbors of M_{left} and moves M_{left} to M_1. The algorithm also finds a better solution in the neighbors of M_{right} 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 neighbors of M_{left} and M_{right} are found. So, 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. Because of Corollary 1, M_{left} and M_{right} are generated in the way that $sm(M_{left})$ increases while $sm(M_{right})$ decreases. Since $sm(M_{left}) > sm(M_{right})$, the algorithm terminates and returns the solution M_4. As shown in Table II, the algorithm finds exactly the *sex-equal* stable matching M_4.

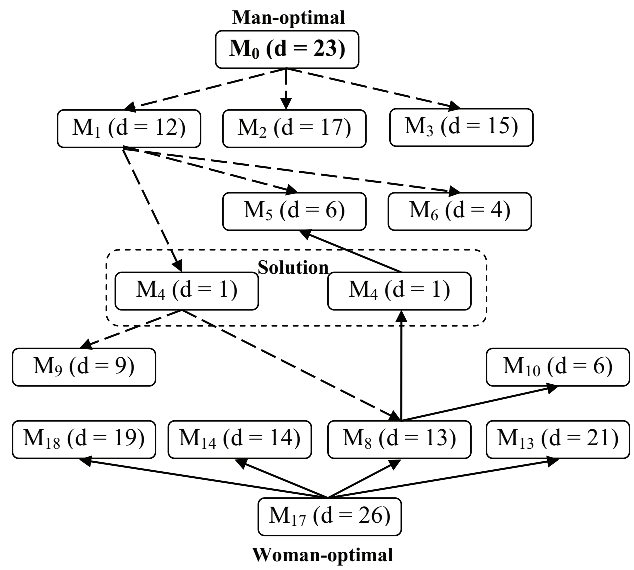


Fig. 2. The trace search for finding the *sex-equal* stable matching of Table I

TABLE III
THE SIMULATION RESULTS OF THE BFS AND BILS ALGORITHMS

Data Set	Size	*BFS algorithm*			*BiLS algorithm*			
		(1)	(2)	(3)	(1)	(3)	(2)	(3)
1	50	645	84	0.337	645	0.117	84	0.107
2	50	647	11	4.465	647	0.165	11	0.123
3	50	678	10	2.366	678	0.112	10	0.166
4	50	669	12	8.110	669	0.135	12	0.143
5	50	788	31	7.038	788	0.174	31	0.202
6	100	1930	63	41.881	1930	0.535	63	0.348
7	100	2021	5	58.872	2021	1.155	5	0.583
8	100	2098	17	17.172	2098	1.638	17	1.402
9	100	2030	13	15.265	2030	0.809	13	0.707
10	100	2031	13	30.293	2031	0.576	13	0.784
11	200	5674	111	888.997	5766†	3.513	324†	1.853
12	200	5615	13	1052.289	5615	2.010	13	1.647
13	200	5594	9	2011.252	5613†	1.549	9	1.041
14	200	5638	156	2412.611	5638	2.570	156	2.117
15	200	5376	14	748.382	5376	2.622	14	2.140

(1): *egalitarian cost*
(2): *sex-equal cost*
(3): *time (second)*
†: *local optimum*

IV. SIMULATIONS

This section presents simulations implemented with the Matlab software on a Core i5-2430M CPU 2.4 GHz with 4 GB RAM computer. The simulations are designed to evaluate the performance of BiLS algorithm for SM instances, in which the preference lists of men and women are generated randomly. The probability of choosing a random stable matching in stable neighbor matchings is set $p = 0.05$.

First, we make simulations to evaluate the solution quality found by BiLS. To do so, we compare the solutions found by BiLS with those found by BFS algorithm. BFS algorithm is an exhaustive search method and therefore, it does not only find all stable matchings but also finds exactly the *egalitarian* and *sex-equal* stable matchings of SM instances. Obviously, using an exhaustive search algorithm like BFS to compare with BiLS is reasonable for evaluation of BiLS. Table III shows the simulation results of 15 SM instances. From the simulation results, three main observations are summarized as follows:

1) The running time of BFS or BiLS algorithms depends not only on the size of SM instances but also on the number of the stable matchings of SM instances.
2) The running time of BiLS is much smaller than that of BFS. In particular, BiLS is efficient when the size of SM instances is large, while BFS is not so since it is an exhaustive search algorithm.
3) The solutions found by BiLS are the same as those found by BFS when the size of SM instances is small. However, BiLS is a local search method and therefore, it can get stuck on a local optimum solution (e.g., the data set at the rows 11 or 13).

Second, we compare BiLS with two local search algorithms to evaluate the computational time and solution quality. The local search algorithms are as follows.

1) The first algorithm is the hill-climbing algorithm [14]. Specifically, the hill-climbing algorithm for SM problem is the same as the Algorithm 2, excepting the backward search phase and the ending condition (i.e, the algorithm ends when $forward = true$).
2) The second algorithm comprises of a sequence of local searches (called SLS algorithm) [15], in which each search is a hill-climbing algorithm. The first hill-climbing algorithm starts from the man-optimal and the next one starts from the solution of the previous one. Because the woman-optimal solution does not generate any stable matching under the dominance relation of stable matchings from the mens point of view (Theorem 1), SLS algorithm ends when a hill-climbing algorithm reaches to the woman-optimal solution. The solution of SLS algorithm is the best solution among solutions of all hill-climbing searches.

We generate randomly 10 SM instances of size n. For each SM instance, we run 50 times and take the cost, the frequency and the average time of finding the solutions. Tables IV, V and VI show the simulation results of BiLS, the hill-climbing and SLS algorithms, respectively. Observations on simulation results can be summarized as follows:

1) The solutions found by BiLS are better than that found by the hill-climbing algorithm and SLS, while the solutions found by the hill-climbing algorithm are almost the same as those found by SLS. For examples in the data set of 7 (respectively 9), BiLS gives the egalitarian matching with cost of 10398 (respectively 21719), but the hill-climbing algorithm and SLS give the egalitarian matching with cost of 12776 (respectively 22014).
2) The frequency of finding the solutions of BiLS is higher than that of SLS. The frequency of finding the solutions of SLS is higher than that of the hill-climbing algorithm.
3) The running time of BiLS is approximately twice bigger than that of the hill-climbing algorithm. Meanwhile, the running time of BiLS is much smaller than that of SLS. Moreover, SLS is inefficient in terms of computational time when the size of SM instances is large.

Finally, we implement the ACS algorithm [8] to compare with BiLS. The simulations show that the ACS algorithm finds an *egalitarian* or *sex-equal* stable matching only for SM instances of small sizes ($n \leq 30$). This is because the ACS algorithm has to find a large amount of pairs (man,woman) to form a stable matching. For example, given an instance of SM of size $n = 100$, the ACS algorithm has to find $n^2 = 10000$ pairs (man,woman) to form a stable matching of 100 engaged pairs. Even the size of SM instances is small, the simulations show that BiLS outperforms the ACS in terms of computational time and solution quality.

TABLE IV
THE SIMULATION RESULTS OF BILS

Data Set	Size	*egalitarian*			*sex-equal*		
		cost	%	time(sec.)	cost	%	time(sec.)
1	50	689	94	0.144	3	98	0.103
2	75	1188	100	0.196	52	96	0.167
3	100	2074	90	0.786	142	100	0.674
4	150	3655	100	1.737	143	100	1.282
5	200	5262	96	3.299	15	100	3.770
6	250	7534	92	3.061	210	94	3.042
7	300	10398	90	6.702	3077	98	3.948
8	400	15915	98	21.648	12	96	23.405
9	500	21719	98	19.054	52	98	20.345
10	700	36260	90	142.687	32	100	126.444

TABLE V
THE SIMULATION RESULTS OF THE HILL-CLIMBING ALGORITHM

Data Set	Size	*egalitarian*			*sex-equal*		
		cost	%	time(sec.)	cost	%	time(sec.)
1	50	689	88	0.057	3	96	0.033
2	75	1188	98	0.113	52	96	0.106
3	100	2074	88	0.423	142	100	0.447
4	150	3655	96	1.020	143	96	0.908
5	200	5262	96	0.992	15	88	1.583
6	250	7629	92	1.121	1234	98	0.799
7	300	12776	90	2.303	7778	98	1.394
8	400	15915	80	15.077	12	96	11.870
9	500	22014	100	11.923	3962	100	8.909
10	700	36338	82	60.258	32	88	84.624

TABLE VI
THE SIMULATION RESULTS OF SLS

Data Set	Size	*egalitarian*			*sex-equal*		
		cost	%	time(sec.)	cost	%	time(sec.)
1	50	689	96	0.155	3	96	0.259
2	75	1188	100	0.411	52	98	0.753
3	100	2074	94	1.588	142	100	1.680
4	150	3655	94	3.139	143	94	3.148
5	200	5262	96	7.596	15	94	8.670
6	250	7629	94	14.668	1234	100	14.692
7	300	12776	96	9.258	7778	100	11.914
8	400	15915	82	63.501	12	92	87.752
9	500	22014	96	86.519	3962	100	89.681
10	700	36260	84	732.249	32	88	827.714

V. CONCLUSIONS

In this paper, we proposed a bidirectional local search algorithm to find an *egalitarian* or *sex-equal* stable matching of SM instances by using the Gale-Shapley algorithm [1], the breakmarriage operation [12], and a random walk. By using the breakmarriage operation to find stable neighbor matchings of the current stable matching, the forward local search finds the solutions while moving towards the woman-optimal stable matching and the backward local search finds the solutions while moving towards the man-optimal stable matching. The proposed algorithm interleaves iterations of the forward search and backwards search until their search frontiers meet. When the algorithm ends, the solution is the best solution of solutions found by the forward search and backwards search. The simulations show that the proposed algorithm is efficient in terms of computational time and solution quality for 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 or the Roommate problem.

ACKNOWLEDGMENT

The authors are grateful to the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2014R1A1A2057735), IITP (2015-(R0134-15-1033)).

REFERENCES

[1] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 9, no. 1, pp. 9–15, 1962.

[2] A. E. Roth, "The evolution of the labor market for medical interns and residents: A case study in game theory," *Journal of Pilitical Economy*, vol. 92, no. 6, pp. 991–1016, 1984.

[3] D. J. Abraham, R. W. Irving, and D. F. Manlove, "The student-project allocation problem," in *Proceedings of the 14th International Symposium*, Kyoto, Japan, Dec. 2003, pp. 474–484.

[4] T. Fleiner, R. W. Irving, and D. F. Manlove, "Efficient algorithms for generalized stable marriage and roommates problems," *Theoretical Computer Science*, vol. 381, no. 1-3, pp. 162–176, 2007.

[5] R. W. Irving, "An efficient algorithm for the "stable roommates" problem," *Journal of Algorithms*, vol. 6, no. 1, pp. 577–595, 1985.

[6] M. Nakamura, K. Onaga, S. Kyan, and M. Silva, "Genetic algorithm for sex-fair stable marriage problem," in *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on*, vol. 1, Seattle, WA, Apr. 1995, pp. 509–512.

[7] D. Gusfield and R. W. Irving, *The stable marriage problem: structure and algorithms*. MIT Press Cambridge, 1989.

[8] N. A. Vien, N. H. Viet, H. Kim, S. Lee, and T. Chung, "Ant colony based algorithm for stable marriage problem," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, vol. 1, 2007, pp. 457–461.

[9] K. Iwama, S. Miyazaki, and H. Yanagisawa, "Approximation algorithms for the sex-equal stable marriage problem," *ACM Transactions on Algorithms*, vol. 7, no. 1, pp. 2:1–2:17, 2010.

[10] R. W. Irving and P. Leather, "The complexity of counting stable marriages," *SIAM Journal on Computing*, vol. 15, no. 3, pp. 655–667, 1986.

[11] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Local search approaches in stable matching problems," *Algorithms*, vol. 6, no. 1, pp. 591–617, 2013.

[12] D. G. McVitie and L. B. Wilson, "The stable marriage problem," *Communication of ACM*, vol. 14, no. 7, pp. 486–490, 1971.

[13] D. Gusfield, "Three fast algorithms for four problems in stable marriage," *SIAM Journal on Computing*, vol. 16, no. 1, pp. 111–128, 1987.

[14] S. Russel and P. Norvig, *Artificial Intelligence A Modern Approach*. Pearson Education, 3nd Ed., 2010.

[15] H. H. Viet, L. H. Trang, S. G. Lee, and T. C. Chung, "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*, Phuket, Thailand, Aug. 2016, pp. 556–564.