

A $(2 - c \frac{\log N}{N})$ -Approximation Algorithm for the Stable Marriage Problem*

Kazuo IWAMA^{†a)}, Fellow, Shuichi MIYAZAKI^{††b)}, Member, and Kazuya OKAMOTO^{†c)}, Nonmember

SUMMARY An instance of the classical stable marriage problem requires all participants to submit a strictly ordered preference list containing all members of the opposite sex. However, considering applications in real-world, we can think of two natural relaxations, namely, incomplete preference lists and ties in the lists. Either variation leaves the problem polynomially solvable, but it is known that finding a maximum cardinality stable matching is NP-hard when both variations are allowed. It is easy to see that the size of any two stable matchings differ by at most a factor of two, and so, an approximation algorithm with a factor two is trivial. A few approximation algorithms have been proposed with approximation ratio better than two, but they are only for restricted instances, such as restricting occurrence of ties and/or lengths of ties. Up to the present, there is no known approximation algorithm with ratio better than two for general inputs. In this paper, we give the first nontrivial result for approximation of factor less than two for general instances. Our algorithm achieves the ratio $2 - c \frac{\log N}{N}$ for an arbitrarily positive constant c , where N denotes the number of men in an input.

key words: stable marriage problem, incomplete lists, ties, approximation algorithms, local search

1. Introduction

The stable marriage problem is a matching problem first introduced by Gale and Shapley [9]. An instance of this problem consists of N men, N women, and each person's preference list. A preference list is a totally ordered list including all members of the opposite sex according to his/her preference. For a matching M between men and women, a pair of a man m and a woman w is called a *blocking pair* if (i) m prefers w to his partner in M , and (ii) w prefers m to her partner in M . A matching with no blocking pair is called *stable*. The stable marriage problem is to find a stable matching for a given instance. Gale and Shapley showed that every instance admits at least one stable matching, and they also proposed so-called the Gale-Shapley algorithm to find one, which runs in $O(N^2)$ time [9].

However, considering an application to large-scale assignment systems, it is unreasonable to force agents to write all members of the other party in a *strict* order. Hence it is natural to think of the following two relaxations: One

is to allow for indifference [11], [16], in which each person is allowed to include *ties* in his/her preference. When ties are allowed, the definition of stability needs to be extended. A man and a woman form a blocking pair if each *strictly* prefers the other to his/her current partner. A matching without such a blocking pair is called *weakly stable*. (There are two other stability definitions, *strong stability* and *super-stability*. However, in this paper, we consider only weak stability, and hence we say simply "stable" instead of "weakly stable".) It is known that the Gale-Shapley algorithm can be modified to always find a stable matching [11]. The other one is to allow participants to declare one or more *unacceptable partners*. Thus each person's preference list may be incomplete. Again, the definition of a blocking pair is extended as follows: A pair of a man m and a woman w is called a blocking pair if (i) m prefers w to his current partner, or m is currently single but includes w in the list, and (ii) w prefers m to her current partner, or w is currently single but includes m in the list. In this case, a stable matching may not be a perfect matching, but all stable matchings for a fixed instance are of the same size [10], and again, it is easy to modify the Gale-Shapley algorithm for this relaxed case. Hence, finding a maximum cardinality stable matching is trivial.

However, if both ties and incomplete lists are allowed, one instance can admit stable matchings of different sizes, and it is known that the problem of finding a maximum stable matching, which we call *MAX SMTI* (MAXimum Stable Marriage with Ties and Incomplete lists), is NP-hard [21], [26]. For approximability, it is easy to see that two stable matchings for the same instance differ in size by at most a factor of two (see Theorem 5 of [26], for example). Since a stable matching can be found in polynomial time by a modified Gale-Shapley algorithm, the existence of an approximation algorithm with a factor of two is trivial. Recently, Halldórsson, et. al. [14] presented several approximability upper bounds which are significantly better than two for restricted inputs, such as a $\frac{2}{(1+1/L^2)}$ -approximation algorithm for instances where length of ties is at most L and ties appear in the lists of only one sex, and a $13/7$ -approximation algorithm for instances where length of ties is two but ties may appear in both sides.

Our Contribution. In this paper, we give the first nontrivial approximability result for *general* MAX SMTI. Namely, our new algorithm, based on local search, achieves an approximation factor of $2 - c \frac{\log N}{N}$, where c is an arbitrarily

Manuscript received February 22, 2006.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.

^{††}The author is with the Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606-8501 Japan.

*A preliminary version of this paper was presented at The 9th Scandinavian Workshop on Algorithm Theory, SWAT 2004

a) E-mail: iwama@kuis.kyoto-u.ac.jp

b) E-mail: shuichi@media.kyoto-u.ac.jp

c) E-mail: okia@kuis.kyoto-u.ac.jp

DOI: 10.1093/ietisy/e89-d.8.2380

positive constant. From an initial stable matching, our algorithm successively improves the size of the solution. While the size of the current solution is at most $\frac{OPT}{2} + c \log N$ where OPT is the size of an optimal solution, we can increase the size by at least one. Hence, we finally obtain a stable matching of size greater than $\frac{OPT}{2} + c \log N$, which gives an upper bound on the approximation ratio.

Related Results. There are several examples of using the stable marriage problem in assignment systems. Among others, one of the most famous applications is to assign medical students to hospitals based on the preference lists of both sides. For example, more than 30,000 applicants are enrolled in the hospitals/residents matching system in the U.S., which is known as NRMP [11], [25]. In Japan, this kind of matching system is used since 2003, where more than 95% of 8,000 applicants obtained their positions in the first year. Other examples for hospitals/residents matching include CaRMS in Canada, and SPA in Scotland [17], [18]. Another famous application is to assign students to schools in Norway [7] and Singapore [29].

As for the problem of approximating MAX SMTI, there have been a lot of positive and negative results. For inapproximability, MAX SMTI was shown to be APX-hard [12], and subsequently, a lower bound 21/19 on the approximation ratio (under the assumption that $P \neq NP$) was presented [14]. This lower bound holds for restricted instances where ties appear in only one sex, the length of ties is two, and each person has at most one tie. For approximability, there are some approximation algorithms with factor better than two for restricted inputs, in which, mainly restrictions are done in terms of occurrence of ties and/or lengths of ties [13], [14], [26], as mentioned previously.

Other than SMTI, research on stable matchings has been quite intensive recently, which includes studies on *strong* stability [20], [24], rank-maximal matchings [19], Pareto optimal matchings [1], popular matchings [2], and others [3], [6], [8].

There are several optimization problems that resemble MAX SMTI, where designing a 2-approximation algorithm is trivial but obtaining a $(2 - \epsilon)$ -approximation algorithm for a positive constant ϵ appears to be extremely hard, such as Minimum Vertex Cover (MIN VC for short) and Minimum Maximal Matching (MIN MM for short). As is the case with MAX SMTI, there are a lot of approximability results for these problems by restricting instances. For example, MIN VC is approximable within 7/6 if the maximum degree of an input graph is bounded by 3 [5], or within $2/(1 + \epsilon)$ if every vertex has degree at least $\epsilon|V|$ [23]. For MIN MM, there is a $(2 - 1/d)$ -approximation algorithm for regular graphs with degree d [30], and PTAS for planar graphs [28]. For general inputs, $(2 - o(1))$ -approximation algorithms are known for MIN VC, namely, $2 - \frac{\log \log |V|}{2 \log |V|}$ and $2 - (1 - o(1)) \frac{2 \ln \ln |V|}{\ln |V|}$ [4], [15], [27]. Recently, this has been improved to $2 - \Theta\left(\frac{1}{\sqrt{\log |V|}}\right)$ [22].

2. Preliminaries

In this section, we formally define MAX SMTI and approximation ratio.

An instance I of MAX SMTI consists of N men, N women, and each person's preference list that may be incomplete and may include ties. If a person q is included in a person p 's list, we say that q is *acceptable* to p . Let m be a man and w_i and w_j be women. If m strictly prefers w_i to w_j in I , we write $w_i \succ_m w_j$. If w_i and w_j are tied in m 's list, we write $w_i =_m w_j$. The statement $w_i \succeq_m w_j$ is true if and only if $w_i \succ_m w_j$ or $w_i =_m w_j$. We use similar notations for women's preference lists.

A matching M is a set of pairs (m, w) such that m is acceptable to w and vice versa, and each person appears at most once in M . The size of a matching M , denoted $|M|$, is the number of pairs in M . We sometimes call a pair $(m, w) \in M$ an *edge* of M . If a man m is matched with a woman w in M , we write $M(m) = w$ and $M(w) = m$. If a person p is not matched in M , we say that p is *single* in M . We say that m and w form a *blocking pair* for M (or simply, (m, w) *blocks* M) if the following three conditions are met: (i) $M(m) \neq w$ but m and w are acceptable to each other. (ii) $w \succ_m M(m)$ or m is single in M . (iii) $m \succ_w M(w)$ or w is single in M . For a matching M , $BP(M)$ denotes the set of blocking pairs for M . A matching M is called *stable* if and only if $BP(M) = \emptyset$. MAX SMTI is a maximization problem defined as follows:

MAX SMTI

Instance: Men, women and preference lists.

Feasible solution: A stable matching.

Measure: The size of matching.

A goodness measure of an approximation algorithm T of a maximization problem is defined as usual: the *approximation ratio* of T is $\max\{opt(x)/T(x)\}$ over all instances x of size N , where $opt(x)$ and $T(x)$ are the size of the optimal and the algorithm's solution, respectively.

3. Overview of Algorithm LOCALSEARCH(I)

Here we give an overview of our algorithm LOCALSEARCH. We need two parameters k and c , which are fixed constants such that $c < \frac{k}{16}$. LOCALSEARCH takes an input I of MAX SMTI and uses two subroutines, INCREASE and STABILIZE.

INCREASE takes a stable matching M for I and a subset S of M such that $|S| = k \log N$. It outputs a (not necessarily stable) matching M_0 such that $|M_0| > |M|$, satisfying the following property II:

II: For any blocking pair $(m, w) \in BP(M_0)$, either m or w (or both) is single in M_0 .

INCREASE may fail to find such a matching. In such a case, it returns an error.

STABILIZE takes a matching M_0 with property II, and outputs a stable matching of size at least $|M_0|$.

Algorithm LOCALSEARCH(I)

```

1:  $M$ : = arbitrary stable matching for  $I$ ;
   /* This can be found in polynomial time by arbitrary tie-
   breaking and applying the Gale-Shapley algorithm. */
2: while (true)
3:   {Select  $(k + 4c) \log N$  edges from  $M$  in an arbitrary way,
   and let  $P$  be the set of selected edges;
4:   Let  $P_1, P_2, \dots, P_n$  be all subsets of  $P$  of size  $k \log N$ ;
5:   for  $i := 1$  to  $n$ 
6:      $M_i :=$  INCREASE ( $M, P_i$ );
     /* If INCREASE returns an error,
     let  $M_i$  be empty. */
7:   if (there is a non-empty  $M_i$ )
8:      $M_0 := M_i$ ;
9:   else
10:    Terminate and output  $M$ ;
11:   $M :=$  STABILIZE ( $M_0$ );
12: }
```

Fig. 1 Algorithm LOCALSEARCH.

The full description of LOCALSEARCH is given in Fig. 1. One can see that application of the while-loop increases the size of stable matching by at least one. This process can continue as long as the condition at line 7 is true. Later, we show in Lemma 4 that this is the case if (1) an input S for INCREASE has some “nice” property, and (2) $|M|$, the size of the input stable matching for INCREASE, is at most $\frac{OPT}{2} + c \log N$, where OPT denotes the size of a maximum stable matching, and c is a constant defined above. Furthermore, we show in Lemma 3 that, among P_1, P_2, \dots, P_n obtained at line 4, there is at least one “nice” P_i if $|M| \leq \frac{OPT}{2} + c \log N$. So, the condition at line 7 is true if $|M| \leq \frac{OPT}{2} + c \log N$, and hence we have the following theorem:

Theorem 1: Given an SMTI instance I of size N , LOCALSEARCH outputs a stable matching of size more than $\frac{OPT}{2} + c \log N$ in time polynomial in N .

Corollary 1: The approximation ratio of LOCALSEARCH is at most $2 - 2c \frac{\log N}{N}$.

Proof. Let M be the output of LOCALSEARCH. Then, $|M| > \frac{OPT}{2} + c \log N$ by Theorem 1. By multiplying both sides by $\frac{2}{|M|}$, we have that

$$\begin{aligned} \frac{OPT}{|M|} &< 2 - 2c \frac{\log N}{|M|} \\ &\leq 2 - 2c \frac{\log N}{N}. \end{aligned}$$

The last inequality comes from the fact that $|M| \leq N$. \square

Since the above constant c can be set arbitrarily large, we have the following corollary.

Corollary 2: For any positive constant c , there is a polynomial-time approximation algorithm for MAX SMTI with approximation ratio at most $2 - c \frac{\log N}{N}$.

Before showing INCREASE and STABILIZE, we define a “nice” property stated above.

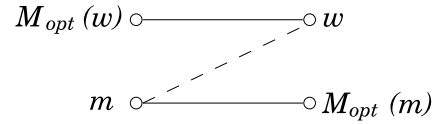


Fig. 2 A good edge (m, w) .

Let us fix an optimal solution M_{opt} , a largest stable matching for I (which we do not know of course). Given a stable matching M for I , let us define the following bipartite graph $G_{M_{opt}, M}$: Each vertex of $G_{M_{opt}, M}$ corresponds to a person in I . There is an edge between vertices m and w if and only if $M_{opt}(m) = w$ or $M(m) = w$. If both $M_{opt}(m) = w$ and $M(m) = w$ hold, we include two edges between m and w ; hence $G_{M_{opt}, M}$ is a multigraph. An edge (m, w) associated with $M_{opt}(m) = w$ is called an *OPT-edge*. Similarly, an edge associated with $M(m) = w$ is called an *M-edge*. Observe that the degree of each vertex is at most two, and hence each connected component of $G_{M_{opt}, M}$ is a simple path, a cycle, or an isolated vertex.

Let us partition M -edges of $G_{M_{opt}, M}$ into good edges and bad ones. If an edge is in the path of length three starting from and ending with OPT-edges, then it is called *good*. Otherwise, it is *bad*. For convenience, we also call an edge in M good (bad, respectively) if the corresponding M -edge in $G_{M_{opt}, M}$ is good (bad, respectively). Figure 2 shows an example of a good edge (m, w) . Solid lines represent OPT-edges and the dotted line represents an M -edge (good edge).

Lemma 1: Let (m, w) be a good edge of M . Then, $w \succeq_m M_{opt}(m)$ and $m \succeq_w M_{opt}(w)$.

Proof. If $M_{opt}(m) \succ_m w$, then $(m, M_{opt}(m))$ is a blocking pair for M , which contradicts the stability of M . So, $w \succeq_m M_{opt}(m)$. For the same reason, $m \succeq_w M_{opt}(w)$. \square

Lemma 2: Let t be an arbitrary positive integer. If $|M| \leq \frac{|M_{opt}|}{2} + t$, then the number of bad edges in $G_{M_{opt}, M}$ is at most $4t$.

Proof. First of all, we show that there is no path of length one in $G_{M_{opt}, M}$. This can be seen as follows: Suppose that there is a path of length one, say (m, w) , and suppose that this is an OPT-edge. Then m and w are acceptable to each other since they are matched in M_{opt} . However, both of them are single in M . This means that (m, w) is a blocking pair for M , which contradicts the stability of M . When (m, w) is an M -edge, we can do a similar argument for a contradiction.

Consider then each connected component C of $G_{M_{opt}, M}$. Let $R(C)$ be the ratio of the number of OPT-edges to the number of M -edges in C . If C is a cycle, then it contains the same number of OPT-edges and M -edges, and hence, $R(C) = 1$. This is the same if C is a path of even length. If C is a path of odd length starting from and ending with M -edges, $R(C) < 1$ since the number of M -edges in C is more than that of OPT-edges. If C is a path of length three starting from and ending with OPT-edges, then the M -edge it contains is good and $R(C) = 2$. If C is a path of length more than three starting from and ending with OPT-edges, then $R(C) \leq 3/2$.

Now, suppose that there are ℓ_1 good edges and ℓ_2 bad edges. Then, the number of OPT-edges, namely $|M_{opt}|$, is at most $2\ell_1 + \frac{3}{2}\ell_2$ by the above argument. Since $\ell_1 + \ell_2 = |M|$ and $|M| \leq \frac{|M_{opt}|}{2} + t$, we have that $\ell_2 \leq 4t$. \square

Lemma 3: If $|M| \leq \frac{|M_{opt}|}{2} + c \log N$, then, at line 4 of LOCALSEARCH, there is at least one i such that P_i contains only good edges.

Proof. Since there are at most $4c \log N$ bad edges in M as proved in Lemma 2, P contains at most $4c \log N$ bad edges. Recall that $|P| = (k + 4c) \log N$. Hence, P contains at least $k \log N$ good edges. Since we consider all subsets of P of size $k \log N$, there must be one with only good edges. \square

4. Procedure INCREASE(M, S)

Recall that INCREASE takes a stable matching M and its subset S of size $k \log N$ as an input, and outputs a matching M_0 such that $|M_0| > |M|$. M_0 may not be stable for I but it satisfies the property II. Before getting into the detail, we roughly explain the execution of INCREASE.

In the following, we assume that S consists of only good edges. (We are interested in computation of INCREASE only in this case. As proved in Lemma 3, there is one way of receiving such S if $|M| \leq \frac{|M_{opt}|}{2} + c \log N$.) Given S , let S_1, S_2, \dots, S_n be all subsets of S of size $|S|/4$. Fix S_i . Since each edge in S_i is good, for each person p in S_i , his/her partner in M_{opt} is single in M . We divorce all pairs of S_i , and then, make them find a partner who is single in M (by the method described in Fig. 4). They may lose to find a partner, but we can prove that if we apply the above procedure to all S_1, S_2, \dots, S_n , then at least one execution will give us a good result, i.e., there exists an S_i such that every person in S_i finds a partner who is at least as good as the partner in M_{opt} (Lemma 5). Let L_i be the set of newly added edges. Then, it is not hard to see that $|L_i| = 2|S_i|$, and hence we can increase the size of M by $|S_i|$. (See Fig. 3 (a).)

In the latter half of the algorithm, we do the following: If there is a blocking pair (m, w) such that both m and w have a partner, say, w' and m' , respectively, then we can prove that exactly one of (m, w') or (m', w) is in L_i (Lemma 6 (2)). We then remove one which is *not* in L_i . (See Fig. 3 (b).) This process may decrease the size of a matching, but we prove that its size-decrease is less than $|S_i|$. In total, we can increase the size of matching by at least one. The full description of algorithm INCREASE is given in Fig. 4. (For the Gale-Shapley algorithm, see [11] for example.)

4.1 Correctness of INCREASE

We give a sufficient condition for INCREASE to achieve a successful computation.

Lemma 4: If S consists of only good edges, and if $|M| \leq \frac{|M_{opt}|}{2} + c \log N$, then there is at least one i (at line 7) such that INCREASE(M, S) succeeds.

The proof of this lemma uses a series of lemmas. In

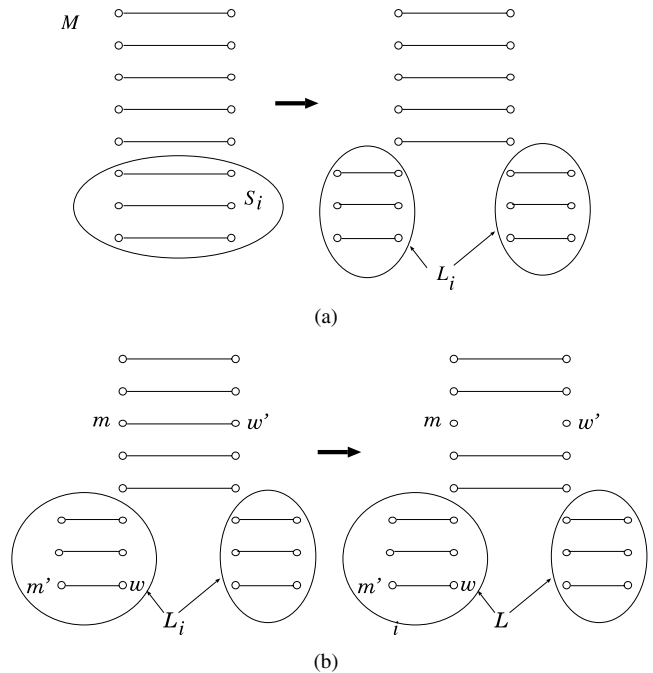


Fig. 3 Execution of INCREASE .

the following lemmas, we assume the same assumptions in Lemma 4, namely, S consists of only good edges, and $|M| \leq \frac{|M_{opt}|}{2} + c \log N$, even if they are not explicitly stated in the statement of lemmas.

Lemma 5: There exists i^* such that, after executing the Gale-Shapley algorithm (at lines 9 and 10 of Fig. 4), every person in $S^{m^*} \cup S^{w^*}$ is matched with a partner who is at least as good as his/her partner in M_{opt} .

Proof. Consider the following procedure (note that we consider this procedure only for the proof of this lemma, and is not a part of INCREASE): Break all ties in preference lists of persons in $S^m \cup S^w \cup F^m \cup F^w$ in the same way as in the execution of line 5 of INCREASE. Furthermore, in each man $m(\in S^m)$'s new list, remove all women below $M_{opt}(m)$. Similarly, in each woman $w(\in S^w)$'s new list, remove all men below $M_{opt}(w)$. (It should be noted that for any person p in $S^m \cup S^w$, $M_{opt}(p)$ can be defined since any element of S is a good edge. Furthermore, $M_{opt}(p)$ is single in M , namely $M_{opt}(p)$ is in $F^m \cup F^w$.)

Apply the men-propose Gale-Shapley algorithm to the subinstance defined by S^m and F^w . It is not hard to see that at least half of S^m are matched at the termination of the Gale-Shapley. To see this, assume the contrary, and let $A \subseteq S^m$ be the set of single men ($|A| > |S^m|/2$). Then, each man $m(\in A)$ was rejected by all women on his modified list, especially, he was rejected by $M_{opt}(m)$. (Recall that $M_{opt}(m)$ stays in m 's modified list.) When $M_{opt}(m)$ rejected m , $M_{opt}(m)$ was matched with someone better than m , and by the property of the Gale-Shapley algorithm, she never becomes single after that. So, at the termination, at least $|A|(> |S^m|/2)$ women are matched, but this means that more than $|S^m|/2$ men are matched, a contradiction.

Procedure INCREASE(M, S)

```

1:  $F^m :=$  set of single men in  $M$ ;
2:  $F^w :=$  set of single women in  $M$ ;
3:  $S^m :=$  set of men in  $S$ ;
4:  $S^w :=$  set of women in  $S$ ;
5: Break all ties in preference lists of  $S^m \cup S^w \cup F^m \cup F^w$ 
   in an arbitrary way;
6: Let  $S_1, S_2, \dots, S_n$  be all subsets of  $S$  of size  $|S|/4$ ;
7: for  $i := 1$  to  $n$ 
8:    $\{S_i^m :=$  set of men in  $S_i$ ;
      $S_i^w :=$  set of women in  $S_i$ ;
9:   Find a matching between  $S_i^m$  and  $F^w$  using the
     men-propose Gale-Shapley algorithm;
10:  Find a matching between  $S_i^w$  and  $F^m$  using the
     women-propose Gale-Shapley algorithm;
11:  if  $(\exists p \in S_i^m \cup S_i^w$  s.t.  $p$  remains single after the
     Gale-Shapley algorithm)
12:    exit for-loop;
     /* The current  $i$  was not a good choice */
13:  else
14:     $\{L_i :=$  the set of all pairs obtained by the
     Gale-Shapley algorithm;
15:     $M_i := M - S_i \cup L_i$ ;
16:    while  $(\exists (m, w) \in BP(M_i)$  s.t. both  $m$  and  $w$ 
     have a partner in  $M_i$ )
17:      if  $((m, M_i(m)) \in L_i$  and  $(M_i(w), w) \in L_i)$ 
18:        exit for-loop;
        /* The current  $i$  was not a good choice */
19:      if  $((m, M_i(m)) \in M_i - L_i$ 
     and  $(M_i(w), w) \in M_i - L_i)$ 
20:        exit for-loop;
        /* The current  $i$  was not a good choice */
21:      if  $((m, M_i(m)) \in M_i - L_i$ 
     and  $(M_i(w), w) \in L_i)$ 
22:         $M_i := M_i - \{(m, M_i(m))\}$ ;
23:      if  $((m, M_i(m)) \in L_i$ 
     and  $(M_i(w), w) \in M_i - L_i)$ 
24:         $M_i := M_i - \{(M_i(w), w)\}$ ;
25:      } /* end while */
26:    if  $(|M_i| > |M|)$ 
27:      output  $M_i$  and terminate;
28:    else exit for-loop;
     /* The current  $i$  was not a good choice */
29:  } /* end else */
30: } /* end for */
31: output "error" and terminate;

```

Fig. 4 Procedure INCREASE .

Now, if $m \in S^m$ has a partner after the execution of the Gale-Shapley algorithm, call m a *successful man*. Call a woman in S^w a *successful woman* if and only if her partner in M is a successful man (there are at least $|S|/2$ successful women). Now, apply the women-propose Gale-Shapley algorithm to the subinstance defined by successful women in S^w and F^m . If a successful woman gets a partner in the resulting matching, call her a *super-successful woman*. For the same reason as above, at least half of successful women are super-successful. Call a pair $(m, w) \in S$ a *super-successful pair* if and only if w is a super-successful woman. There are at least $|S|/4$ super-successful pairs.

Since S_1, S_2, \dots, S_n are all subsets of S of size exactly $|S|/4$, there exists at least one i such that S_i consists of only super-successful pairs. Let i^* be any such i , and consider

the execution of INCREASE at lines 7 through 10 for i^* . It is not hard to see that after INCREASE completes the Gale-Shapley algorithm (at lines 9 and 10), each person in $S_{i^*}^m \cup S_{i^*}^w$ gets a partner not worse than the one obtained by the above procedure. This implies that every person in $S_{i^*}^m \cup S_{i^*}^w$ is matched with a partner not worse than the partner in M_{opt} (with respect to preference lists after ties are broken at line 5 of INCREASE). To complete the proof, observe that this fact also holds in terms of the original preference lists before ties are broken. \square

In the following lemmas, i^* always denotes the one that satisfies the condition of Lemma 5.

Lemma 6: M_{i^*} at line 15 of Fig. 4 satisfies following (1) and (2): (1) $|M_{i^*}| = |M| + \frac{k}{4} \log N$. (2) Consider an arbitrary blocking pair $(m, w) \in BP(M_{i^*})$ such that both m and w are matched in M_{i^*} . Then, one of $(m, M_{i^*}(m))$ and $(M_{i^*}(w), w)$ is in $M_{i^*} - L_{i^*}$ and the other is in L_{i^*} .

Proof. (1) Recall that $|S_{i^*}| = |S|/4 = \frac{k}{4} \log N$ and $|L_{i^*}| = 2|S_{i^*}|$. Then, $|M_{i^*}| = |M| - |S_{i^*}| + |L_{i^*}| = |M| + |S_{i^*}| = |M| + \frac{k}{4} \log N$.

(2) First, suppose that both $(m, M_{i^*}(m))$ and $(M_{i^*}(w), w)$ are in $M_{i^*} - L_{i^*}$. Observe that, by the construction of M_{i^*} , both of these two pairs are also in M . This means that $(m, w) \in BP(M)$, which contradicts the stability of M .

Next, suppose that both $(m, M_{i^*}(m))$ and $(M_{i^*}(w), w)$ are in L_{i^*} . We have four cases to consider: (i) $m \in F^m, w \in F^w$, (ii) $m \in S_{i^*}^m, w \in F^w$, (iii) $m \in F^m, w \in S_{i^*}^w$ and (iv) $m \in S_{i^*}^m, w \in S_{i^*}^w$.

Case (i): By the definition of F^m and F^w , both m and w are single in M . But since (m, w) forms a blocking pair for M_{i^*} , m and w are acceptable to each other. Then, $(m, w) \in BP(M)$, which contradicts the stability of M .

Case (ii): By the assumption that (m, w) is a blocking pair for M_{i^*} , $w \succ_m M_{i^*}(m)$. Observe that $w \succ_m M_{i^*}(m)$ holds even after ties in m 's list is broken at line 5 of INCREASE. Then, since $w \in F^w$, during the execution of the Gale-Shapley algorithm at line 9, m proposed to w but w rejected m . Hence, $M_{i^*}(w) \succ_w m$ with respect to w 's list after ties are broken. (This comes from the fact that when w rejected m , w was matched with a man better than m , and in the rest of the execution, she never changes partner to a worse man.) Then, $M_{i^*}(w) \geq_w m$ with respect to w 's original list. Hence, (m, w) cannot block M_{i^*} , a contradiction.

Case (iii): Similar to Case (ii).

Case (iv): Since (m, w) is a blocking pair for M_{i^*} , $w \succ_m M_{i^*}(m)$ and $m \succ_w M_{i^*}(w)$. But by Lemma 5, $M_{i^*}(m) \geq_m M_{opt}(m)$ and $M_{i^*}(w) \geq_w M_{opt}(w)$. Then, $w \succ_m M_{opt}(m)$ and $m \succ_w M_{opt}(w)$, which means that (m, w) is a blocking pair for M_{opt} , a contradiction. \square

The proof of Lemma 4 is completed by the following lemma, which guarantees the size of $|M_{i^*}|$ at line 26 of Fig. 4.

Lemma 7: M_{i^*} at line 26 of INCREASE satisfies $|M_{i^*}| > |M|$.

Proof. First of all, it should be noted that INCREASE never

fails on i^* at lines 11 and 12 by Lemma 5. Also, during the execution of the while-loop (starting from line 16) on i^* , INCREASE never fails by Lemma 6 (2). By Lemma 6 (1), we know that $|M_{i^*}| = |M| + \frac{k}{4} \log N$. However, during the execution of the while-loop, some pairs may be removed from $M_{i^*} - L_{i^*}$, which may decrease the size of M_{i^*} . Note that all pairs in $M_{i^*} - L_{i^*}$ are pairs in M . In the following, we show that if a pair in $M_{i^*} - L_{i^*}$ is removed during the while-loop, then the pair must be a bad edge of M . If this is true, the number of removed pairs in the while-loop is at most $4c \log N$ by Lemma 2, and thus $|M_{i^*}| \geq |M| + \frac{k}{4} \log N - 4c \log N > |M|$. (Recall that $c < \frac{k}{16}$.)

Suppose that during the while-loop of INCREASE, some pair is removed from M_{i^*} . Then, there is a blocking pair (m, w) for M_{i^*} and both m and w are matched in M_{i^*} . We have two cases: (1) $(m, M_{i^*}(m)) \in L_{i^*}$ and $(M_{i^*}(w), w) \in M_{i^*} - L_{i^*}$ (and hence $(M_{i^*}(w), w)$ is removed). (2) $(m, M_{i^*}(m)) \in M_{i^*} - L_{i^*}$ and $(M_{i^*}(w), w) \in L_{i^*}$ (and hence $(m, M_{i^*}(m))$ is removed). We consider only Case (1). (Case (2) can be treated similarly.) Now, suppose that the removed pair $(M_{i^*}(w), w)$ is a good edge of M . We will show a contradiction. We further consider two subcases: (1-1) $m \in F^m$ and (1-2) $m \in S_{i^*}^m$.

Case (1-1): Note that m is single in M since $m \in F^m$. Now observe that, as $(M_{i^*}(w), w) \in M_{i^*} - L_{i^*}$, w and $M_{i^*}(w)$ are matched in M , namely, $M_{i^*}(w) = M(w)$. Since $(m, w) \in BP(M_{i^*})$, it results that $(m, w) \in BP(M)$, which contradicts the stability of M . (In this case, we can have a contradiction without assuming that $(M_{i^*}(w), w)$ is a good edge of M .)

Case (1-2): Since we assume that $(M_{i^*}(w), w)$ is a good edge of M , $M(w) \succeq_w M_{opt}(w)$ by Lemma 1. For the same reason as above, $M_{i^*}(w) = M(w)$. So, $M_{i^*}(w) \succeq_w M_{opt}(w)$. As (m, w) is a blocking pair for M_{i^*} , it results that $m \succ_w M_{i^*}(w) \succeq_w M_{opt}(w)$. Next, consider the man m which we assumed to be in $S_{i^*}^m$. By Lemma 5, $M_{i^*}(m) \succeq_m M_{opt}(m)$. Again, as (m, w) is a blocking pair for M_{i^*} , $w \succ_m M_{i^*}(m)$. So, $w \succ_m M_{i^*}(m) \succeq_m M_{opt}(m)$. Consequently, (m, w) is in $BP(M_{opt})$, a contradiction. \square

5. Procedure STABILIZE(M_0)

STABILIZE takes a matching M_0 with property Π and makes it stable without decreasing the size. Recall that for any blocking pair (m, w) for M_0 , at least m or w is single in M_0 . For a matching M , define $BP_{s,m}(M) \subseteq BP(M)$ to be the set of blocking pairs (m, w) for M such that m is single in M and w is matched in M . Similarly, $BP_{m,s}(M)$ ($BP_{s,s}(M)$ and $BP_{m,m}(M)$, respectively) denotes the set of blocking pairs (m, w) for M such that m is matched and w is single (both m and w are single, and both m and w are matched, respectively) in M . Define $BP_{-,s}(M) = BP_{m,s}(M) \cup BP_{s,s}(M)$, and $BP_{-,m}(M) = BP_{m,m}(M) \cup BP_{s,m}(M)$. Figure 5 shows the procedure STABILIZE.

5.1 Correctness of STABILIZE

Lemma 8: Suppose that an application of line 4 of

Procedure STABILIZE(M_0)

```

1: while (  $BP_{s,m}(M_0) \neq \emptyset$  )
2:   {Select  $(m, w) \in BP_{s,m}(M_0)$ ;
3:    $w^* :=$  woman s.t.  $(m, w^*) \in BP_{s,m}(M_0)$  and
     there is no  $(m, w') \in BP_{s,m}(M_0)$  s.t.  $w' \succ_m w^*$ ;
4:    $M_0 := M_0 - \{(M_0(w^*), w^*)\} \cup \{(m, w^*)\}$ ;
5:   }
6: while (  $BP_{-,s}(M_0) \neq \emptyset$  )
7:   {Select  $(m, w) \in BP_{-,s}(M_0)$ ;
8:    $m^* :=$  man s.t.  $(m^*, w) \in BP_{-,s}(M_0)$  and
     there is no  $(m', w) \in BP_{-,s}(M_0)$  s.t.  $m' \succ_w m^*$ ;
9:   if ( $m^*$  is matched in  $M_0$ )
10:     $M_0 := M_0 - \{(m^*, M_0(m^*))\} \cup \{(m^*, w)\}$ ;
11:   else
12:     $M_0 := M_0 \cup \{(m^*, w)\}$ ;
13:   }
```

Fig. 5 Procedure STABILIZE .

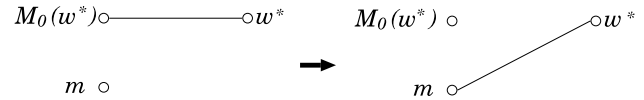


Fig. 6 An update by STABILIZE .

STABILIZE updates M_0 as follows. (see Fig. 6).

$$M'_0 := M_0 - \{(M_0(w^*), w^*)\} \cup \{(m, w^*)\}.$$

Then, following (1) through (3) hold. (1) $M'_0(w^*) \succ_w M_0(w^*)$ and for any $w(\neq w^*)$, $M'_0(w) = M_0(w)$. (2) $|M'_0| = |M_0|$. (3) If $BP_{m,m}(M_0) = \emptyset$, then $BP_{m,m}(M'_0) = \emptyset$.

Proof. (1) Since (m, w^*) is in $BP(M_0)$, $m \succ_w M_0(w^*)$. So, $M'_0(w^*) \succ_w M_0(w^*)$ because $M'_0(w^*) = m$. The latter part of (1) is trivial because, among all women, only w^* changed a partner.

(2) This is easy because we removed one edge and added one edge.

(3) Recall that $BP_{m,m}(M'_0) \subseteq BP(M'_0)$. To prove that $BP_{m,m}(M'_0)$ is empty, we show that any blocking pair in $BP(M'_0)$ is not in $BP_{m,m}(M'_0)$.

First, we will show that any element in $BP(M'_0) - BP(M_0)$ (namely, a pair non-blocking for M_0 but blocking for M'_0) is not in $BP_{m,m}(M'_0)$. Observe that three persons changed the partner by updating from M_0 to M'_0 : w^* obtained a better partner, m became matched from single, and $M_0(w^*)$ became single from matched. So, any blocking pair arising by changing from M_0 to M'_0 is associated with the man $M_0(w^*)$. But, since $M_0(w^*)$ is single in M'_0 , that pair cannot be in $BP_{m,m}(M'_0)$.

Next, consider $(\tilde{m}, \tilde{w}) \in BP(M'_0) \cap BP(M_0)$. Since $BP_{m,m}(M_0) = \emptyset$, at least one of \tilde{m} and \tilde{w} is single in M_0 . Recall that only m changed the status from single to matched. So if $\tilde{m} \neq m$, $(\tilde{m}, \tilde{w}) \notin BP_{m,m}(M'_0)$. Hence, it remains to consider the case $\tilde{m} = m$.

Then, consider a blocking pair $(m, \tilde{w}) \in BP(M'_0) \cap BP(M_0)$. If \tilde{w} was single in M_0 , she is also single in M'_0 and hence $(m, \tilde{w}) \notin BP_{m,m}(M'_0)$. So assume that \tilde{w} was matched in M_0 . In this case, both (m, w^*) and (m, \tilde{w}) were

in $BP_{s,m}(M_0)$. So, both w^* and \tilde{w} were candidates for being matched with m in M'_0 . But since w^* was selected, it must be the case that $w^* \succeq_m \tilde{w}$. Hence (m, \tilde{w}) cannot block M'_0 , a contradiction. \square

Lemma 9: Suppose that an application of lines 10 and 12 of STABILIZE updates M_0 as follows.

(Line 10) $M'_0 := M_0 - \{(m^*, M_0(m^*))\} \cup \{(m^*, w)\}$.

(Line 12) $M'_0 := M_0 \cup \{(m^*, w)\}$.

Then, following (1) through (3) hold. (1) In case of executing line 10, $M'_0(m^*) \succ_{m^*} M_0(m^*)$ (in case of executing line 12, m^* becomes matched in M'_0), and for any $m(\neq m^*)$, $M'_0(m) = M_0(m)$. (2) $|M'_0| \geq |M_0|$. (3) If $BP_{-m}(M_0) = \emptyset$, then $BP_{-m}(M'_0) = \emptyset$.

Proof. Proofs are similar to that of Lemma 8, and we omit (1) and (2).

(3) Observe that three or two persons changed the partner by updating from M_0 to M'_0 : m^* obtained a better partner or became matched from single, w became matched from single, and $M_0(m^*)$, if exists, became single from matched. So, in case of line 12, there arises no new blocking pair, and hence $BP(M'_0) - BP(M_0)$ is empty. In case of line 10, any new blocking pair is associated with the woman $M_0(m^*)$. Since $M_0(m^*)$ is single in M'_0 , any pair in $BP(M'_0) - BP(M_0)$ is not in $BP_{-m}(M'_0)$.

Next, consider $(\tilde{m}, \tilde{w}) \in BP(M'_0) \cap BP(M_0)$. Since $BP_{-m}(M_0) = \emptyset$, \tilde{w} is single in M_0 . Recall that, among all women, only w changed the status from single to matched. So, if $\tilde{w} \neq w$, $(\tilde{m}, \tilde{w}) \notin BP_{-m}(M'_0)$.

Now consider a blocking pair $(\tilde{m}, w) \in BP(M'_0) \cap BP(M_0)$. In this case, both (m^*, w) and (\tilde{m}, w) were in $BP_{-s}(M_0)$. So, both m^* and \tilde{m} were candidates for being matched with w in M'_0 . But since m^* was selected, it must be the case that $m^* \succeq_w \tilde{m}$. Hence (\tilde{m}, w) cannot block M'_0 , a contradiction. \square

Using these two lemmas, we prove the correctness of STABILIZE.

Lemma 10: Let M' be the output of STABILIZE. Then M' is stable and $|M'| \geq |M_0|$.

Proof. Consider an application of line 4 of STABILIZE. By Lemma 8 (1), one woman gets better off and all other women do not change the marital status. Since there are N women, each with a preference list of length at most N , the number of repetitions of the first while-loop is at most N^2 . Let M'' be the matching just before STABILIZE starts the second while-loop. Then $BP_{s,m}(M'')$ is empty. (This is the condition for STABILIZE to exit from the first while-loop.) Since $BP_{m,m}(M_0)$ is empty, we can show that $BP_{m,m}(M'')$ is empty by applying Lemma 8 (3) repeatedly. Combining these two facts, it results that $BP_{-m}(M'')$ is empty. Also, by Lemma 8 (2), $|M''| = |M_0|$.

Similarly as above, each application of line 10 or 12 would make men better off (Lemma 9 (1)), and hence the number of repetitions of the second while-loop is at most N^2 . Since $BP_{-m}(M'') = \emptyset$, we can show that, $BP_{-m}(M') =$

\emptyset using Lemma 9 (3) repeatedly. However, the termination condition of STABILIZE says that $BP_{-s}(M') = \emptyset$. Consequently, $BP(M')$ is empty and hence M' is stable. By Lemma 9 (2), $|M'| \geq |M''|$. So, $|M'| \geq |M_0|$. \square

Acknowledgment

This research was supported in part by Scientific Research Grant, Ministry of Education, Japan, 16300002, 16092101, 16092215, 15700010 and 17700015.

References

- [1] D.J. Abraham, K. Cechlárová, D.F. Manlove, and K. Mehlhorn, "Pareto optimality in house allocation problems," Proc. ISAAC 2004, LNCS 3341, pp.3–15, 2004.
- [2] D.J. Abraham, R.W. Irving, T. Kavitha, and K. Mehlhorn, "Popular matchings," Proc. SODA 2005, pp.424–432, 2005.
- [3] V. Bansal, A. Agrawal, and V. Malhotra, "Stable marriages with multiple partners: Efficient search for an optimal solution," Proc. ICALP 2003, LNCS 2719, pp.527–542, 2003.
- [4] R. Bar-Yehuda and S. Even, "A local-ratio theorem for approximating the weighted vertex cover problem," in Analysis and Design of Algorithms for Combinatorial Problems, Annals of Disc. Math. vol.25, pp.27–46, Elsevier Science, Amsterdam, 1985.
- [5] P. Berman and T. Fujito, "On the approximation properties of independent set problem in degree 3 graphs," Proc. WADS 95, LNCS 955, pp.449–460, 1995.
- [6] K. Cechlárová, "On the complexity of exchange-stable roommates," Discrete Appl. Math., vol.116, pp.279–287, 2002.
- [7] T. Elde, "Assignment engine," Senior Honours Project Report, Department of Computing Science, University of Glasgow, 2000.
- [8] T. Fleiner, "A fixed-point approach to stable matchings and some applications," Mathematics of Operations Research, vol.28, no.1, pp.103–126, 2003.
- [9] D. Gale and L.S. Shapley, "College admissions and the stability of marriage," Amer. Math. Monthly, vol.69, pp.9–15, 1962.
- [10] D. Gale and M. Sotomayor, "Some remarks on the stable matching problem," Discrete Appl. Math., vol.11, pp.223–232, 1985.
- [11] D. Gusfield and R. Irving, The Stable Marriage Problem: Structure and Algorithms, MIT Press, Boston, MA, 1989.
- [12] M. Halldórsson, R. Irving, K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, and S. Scott, "Approximability results for stable marriage problems with ties," Theor. Comput. Sci., vol.306, pp.431–447, 2003.
- [13] M. Halldórsson, K. Iwama, S. Miyazaki and H. Yanagisawa, "Randomized approximation of the stable marriage problem," Proc. COCOON 2003, LNCS 2697, pp.339–350, 2003.
- [14] M. Halldórsson, K. Iwama, S. Miyazaki, and H. Yanagisawa, "Improved approximation of the stable marriage problem," Proc. ESA 2003, LNCS 2832, pp.266–277, 2003.
- [15] E. Halperin, "Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs," Proc. SODA 2000, pp.329–337, 2000.
- [16] R. Irving, "Stable marriage and indifference," Discrete Appl. Math., vol.48, pp.261–272, 1994.
- [17] R. Irving, "Matching medical students to pairs of hospitals: A new variation on an old theme," Proc. ESA 98, LNCS 1461, pp.381–392, 1998.
- [18] R. Irving, D. Manlove, and S. Scott, "The hospital/residents problem with ties," Proc. SWAT 2000, LNCS 1851, pp.259–271, 2000.
- [19] R. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch, "Rank-maximal matchings," Proc. SODA 2004, pp.68–75, 2004.
- [20] R.W. Irving, D.F. Manlove, and S. Scott, "Strong stability in the hospitals/residents problem," Proc. STACS 2003, LNCS 2607, pp.439–

450, 2003.

- [21] K. Iwama, D. Manlove, S. Miyazaki, and Y. Morita, "Stable marriage with incomplete lists and ties," Proc. ICALP 99, LNCS 1644, pp.443–452, 1999.
- [22] G. Karakostas, "A better approximation ratio for the vertex cover problem," ECCO Report, TR04-084, 2004.
- [23] M. Karpinski, "Polynomial time approximation schemes for some dense instances of NP-hard optimization problems," Proc. RANDOM 97, LNCS 1269, pp.1–14, 1997.
- [24] T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch, "Strongly stable matchings in time $O(nm)$ and extension to the H/R problem," Proc. STACS 2004, 2004.
- [25] T. Le, V. Bhushan, and C. Amin, First Aid for the Match, second ed., McGraw-Hill, 2001.
- [26] D. Manlove, R. Irving, K. Iwama, S. Miyazaki, and Y. Morita, "Hard variants of stable marriage," Theor. Comput. Sci., vol.276, no.1-2, pp.261–279, 2002.
- [27] B. Monien and E. Speckenmeyer, "Ramsey numbers and an approximation algorithm for the vertex cover problem," Acta Inf., vol.22, pp.115–123, 1985.
- [28] H. Nagamochi, Y. Nishida, and T. Ibaraki, "Approximability of the minimum maximal matching problem in planar graphs," IEICE Trans. Fundamentals, vol.E86-A, no.12, pp.3251–3258, Dec. 2003.
- [29] C.P. Teo, J.V. Sethuraman, and W.P. Tan, "Gale-Shapley stable marriage problem revisited: Strategic issues and applications," Proc. IPCO 99, pp.429–438, 1999.
- [30] M. Zito, Randomized techniques in combinatorial algorithmics, PhD thesis, Dept. of Computer Science, University of Warwick, 1999.



Kazuya Okamoto is a student of Graduate School of Informatics, Kyoto University, Kyoto, Japan. He received BE and ME degrees from Kyoto University in 2004 and 2006, respectively. His research interests include algorithms.



Kazuo Iwama is a professor of Graduate School of Informatics, Kyoto University, Kyoto, Japan. He received BE, ME, and Ph.D. degrees in Electrical Engineering from Kyoto University in 1978, 1980 and 1985, respectively. His research interests include algorithms, complexity theory and quantum computation.



Shuichi Miyazaki is an associate professor of Academic Center for Computing and Media Studies, Kyoto University, Kyoto, Japan. He received BE, ME, and Ph.D. degrees from Kyushu University in 1993, 1995 and 1998, respectively. His research interests include algorithms and complexity theory.