An Improved Approximation Algorithm for the Stable Marriage Problem with One-Sided Ties

Chien-Chung Huang[1] and Telikepalli Kavitha[2,⋆]

[1] Chalmers University, Sweden
huangch@chalmers.se
[2] Tata Institute of Fundamental Research, India
kavitha@tcs.tifr.res.in

Abstract. We consider the problem of computing a large stable matching in a bipartite graph $G = (A \cup B, E)$ where each vertex $u \in A \cup B$ ranks its neighbors in an order of preference, perhaps involving ties. A matching M is said to be *stable* if there is no edge (a, b) such that a is unmatched or prefers b to $M(a)$ and similarly, b is unmatched or prefers a to $M(b)$. While a stable matching in G can be easily computed in linear time by the Gale-Shapley algorithm, it is known that computing a maximum size stable matching is APX-hard.

In this paper we consider the case when the preference lists of vertices in A are *strict* while the preference lists of vertices in B may include ties. This case is also APX-hard and the current best approximation ratio known here is 25/17 ≈ 1.4706 which relies on solving an LP. We improve this ratio to 22/15 ≈ 1.4667 by a simple linear time algorithm.

We first compute a half-integral stable matching in $\{0, 0.5, 1\}^{|E|}$ and round it to an integral stable matching M. The ratio $|\mathsf{OPT}|/|M|$ is bounded via a payment scheme that charges other components in $\mathsf{OPT} \oplus M$ to cover the costs of length-5 augmenting paths. There will be no length-3 augmenting paths here.

We also consider the following special case of two-sided ties, where every tie length is 2. This case is known to be UGC-hard to approximate to within 4/3. We show a 10/7 ≈ 1.4286 approximation algorithm here that runs in linear time.

1 Introduction

The stable marriage problem is a classical and well-studied matching problem in bipartite graphs. The input here is a bipartite graph $G = (A \cup B, E)$ where every $u \in A \cup B$ ranks its neighbors in an order of preference and ties are permitted in preference lists. It is customary to refer to the vertices in A and B as *men* and *women*, respectively. Preference lists may be incomplete: that is, a vertex need not be adjacent to all the vertices on the other side.

A matching is a set of edges, no two of which share an endpoint. An edge (a, b) is said to be a *blocking edge* for a matching M if either a is unmatched or prefers b to its partner in M, i.e., $M(a)$, and similarly, b is unmatched or prefers a to its partner $M(b)$. A matching that admits no blocking edges is said to be *stable*. The problem of

⋆ Part of this work was done while visiting the Max-Planck-Institut für Informatik, Saarbrücken under the IMPECS program.

J. Lee and J. Vygen (Eds.): IPCO 2014, LNCS 8494, pp. 297–308, 2014.

computing a stable matching in G is the stable marriage problem. A stable matching always exists and can be computed in linear time by the well-known Gale-Shapley algorithm [2].

Several real-world assignment problems can be modeled as the stable marriage problem; for instance, the problems of assigning residents to hospitals [4] or students to schools [19]. The input instance could admit many stable matchings and the desired stable matching in most real-world applications is a maximum cardinality stable matching. When preference lists are *strict* (no ties permitted), it is known that all stable matchings in G have the same size and the set of vertices matched in every stable matching is the same [3]. However when preference lists involve ties, stable matchings can vary in size.

Consider the following simple example, where $A = \{a_1, a_2\}$ and $B = \{b_1, b_2\}$ and let the preference lists be as follows:

$a_1 : b_1;$ $a_2 : b_1, b_2;$ $b_1 : \{a_1, a_2\};$ and $b_2 : a_2.$

The preference list of a_1 consists of just b_1 while the preference list of a_2 consists of b_1 followed by b_2. The preference list of b_1 consists of a_1 and a_2 *tied* as the top choice while the preference list of b_2 consists of the single vertex a_2. There are 2 stable matchings here: $\{(a_2, b_1)\}$ and $\{(a_1, b_1), (a_2, b_2)\}$. Thus the sizes of stable matchings in G could differ by a factor of 2 and it is easy to see that they cannot differ by a factor more than 2 since every stable matching has to be a *maximal* matching. As stated earlier, the desired matching here is a maximum size stable matching. However it is known that computing such a matching is NP-hard [8,15].

Iwama et al. [9] showed a $15/8 = 1.875$-approximation algorithm for this problem using a local search technique. The next breakthrough was due to Király [11], who introduced the simple and effective technique of "promotion" to break ties in a modification of the Gale-Shapley algorithm. He improved the approximation ratio to $5/3$ for the general case and to 1.5 for *one-sided ties*, i.e., the preference lists of vertices in A have to be strict while ties are permitted in the preference lists of vertices in B. McDermid [16] then improved the approximation ratio for the general case also to 1.5. For the case of one-sided ties, Iwama et al. [10] showed a $25/17 \approx 1.4706$-approximation.

On the inapproximability side, the strongest hardness results are due to Yanagisama [21] and Iwama et al. [9]. In [21], the general problem was shown to be NP-hard to approximate to within $33/29$ and UGC-hard to approximate to within $4/3$; the case of one-sided ties was considered in [9] and shown to be NP-hard to approximate to within $21/19$ and UGC-hard to approximate to within $5/4$.

In this paper we focus mostly on the case of one-sided ties. The case of one-sided ties occurs frequently in several real-world problems; for instance, in the Scottish Foundation Allocation Scheme (SFAS), the preference lists of applicants have to be strictly ordered while the preference lists of positions can admit ties [7]. Let OPT be a maximum size stable marriage in the given instance. We show the following result here.

Theorem 1. *Let $G = (A \cup B, E)$ be a stable marriage instance where vertices in A have strict preference lists while vertices in B are allowed to have ties in preference lists. A stable matching M in G such that $|\mathsf{OPT}|/|M| \leq 22/15 \approx 1.4667$ can be computed in linear time.*

Techniques. Our algorithm constructs a *half-integral* stable matchings using a modified Gale-Shapley algorithm: each man can make two proposals and each woman can accept two proposals. How the proposals are made by men and how women accept these proposals forms the core part of our algorithms. In our algorithms, after the proposing phase is over, we have a half-integral vector x, where $x_{ab} = 1$ (similarly, 1/2 or 0) if b accepts 2 (respectively, 1 or 0) proposals from a. We then build a subgraph G' of G by retaining an edge e only if $x_e > 0$. Our solution is a maximum cardinality matching in G' where every degree 2 vertex gets matched.

In the original Gale-Shapley algorithm, when two proposals are made to a woman from men that are tied on her list, she is forced to make a blind choice since she has no way of knowing which is a better proposal (i.e., it leads to a larger matching) to accept. Our approach to deal with this issue is to let her accept both proposals. Since neither proposer is fully accepted, each of them has to propose down his list further and get another proposal accepted. Essentially, our strategy of letting men make multiple proposals and letting women accept multiple proposals is a way of coping with their lack of knowledge about the best decision at any point in time. Note that we limit the number of proposals a man makes/a woman accepts to be 2 because we want the graph G' to have a simple structure. In our algorithms, every vertex in G' has degree at most 2 and this allows us to bound our approximation guarantees.

We first show that there are no length-3 augmenting paths in $M \oplus \mathsf{OPT}$ using the idea of *promotion* introduced by Király [11] to break ties in favor of those vertices rejected once by all their neighbors. This idea was also used by McDermid [16] and Iwama et al. [10]. This idea essentially guarantees an approximation factor of 1.5 by eliminating all length-3 augmenting paths in $M \oplus \mathsf{OPT}$. In order to obtain an approximation ratio < 1.5, we use a new combinatorial technique that makes components other than augmenting paths of length-5 in $M \oplus \mathsf{OPT}$ pay for augmenting paths of length-5.

Let R denote the set of augmenting paths of length-5 in $M \oplus \mathsf{OPT}$ and let $Q = (M \oplus \mathsf{OPT}) \setminus R$. Suppose $q \in Q$ is an augmenting path on $2\ell + 3 \geq 7$ edges or an alternating cycle/path on 2ℓ edges or an alternating path on $2\ell - 1$ edges (with ℓ edges of M). In our algorithm for one-sided ties, q will be charged for $\leq 3\ell$ elements in R and this will imply that $|\mathsf{OPT}|/|M| \leq 22/15$.

For the case of one-sided ties, to obtain an approximation guarantee < 1.5, the algorithm by Iwama et al. [10] formulates the maximum cardinality stable matching problem as an integer program and solves its LP relaxation. This optimal LP-solution guides women in accepting proposals and leads to a 25/17-approximation.

It was also shown in [10] that for two-sided ties, the integrality gap of a natural LP for this problem (first used in [20]) is $1.5 - \Theta(1/n)$. As mentioned earlier, McDermid [16] gave a 1.5-approximation algorithm here; Király [12] and Paluch [17] have shown linear time algorithms for this ratio. A variation of the general problem was recently studied by Askalidis et al. [1].

Since no approximation guarantee better than 1.5 is known for the general case of two-sided ties while better approximation algorithms are known for the one-sided ties case, as a first step we consider the following variant of two-sided ties where each tie length is 2. This is a natural variant as there are several application domains where ties are permitted but their length has to be small. We show the following result here.

Theorem 2. *Let $G = (A \cup B, E)$ be a stable marriage instance where vertices in $A \cup B$ are allowed to have ties in preference lists, however each tie has length 2. A stable matching M' in G such that $|\mathsf{OPT}|/|M'| \le 10/7 \approx 1.4286$ can be computed in linear time.*

Currently, this is the only case with approximation ratio better than 1.5 for any special case of the stable marriage problem where ties can occur on *both* sides of G. Interestingly, in the hardness results shown in [21] and [9], it is assumed that each vertex has at most one tie in its preference list, and such a tie is of length 2. Thus if the general case really has higher inapproximability, say 1.5 as previously conjectured by Király [11], then the reduction in the hardness proof needs to use longer ties.

We also note that the ratio of $10/7$ we achieve in this special case coincides with the ratio attained by Halldórsson et al. [5] for the case that ties only appear on women's side and each tie is of length 2.

The stable marriage problem is an extensively studied subject on which several monographs [4,13,14,18] are available. The generalization of allowing ties in the preference lists was first introduced by Irving [6]. There are several ways of defining stability when ties are allowed in preference lists. The definition, as used in this paper, is Irving's "weak-stability."

Due to the space limit, we only present our algorithm for one-sided ties in Section 2 and its analysis in Section 3. Some missing proofs, along with the algorithm for two-sided ties where each tie has length 2, can be found in the full version.

2 Our Algorithm

Our algorithm produces a fractional matching $x = (x_e, e \in E)$ where each $x_e \in \{0, 1/2, 1\}$. The algorithm is a modification of the Gale-Shapley algorithm in $G = (A \cup B, E)$. We first explain how men propose to women and then how women decide (see Fig. 1).

How men propose. Every man a has two proposals p_a^1 and p_a^2, where each proposal p_a^i (for $i = 1, 2$) goes to the women on a's preference list in a round-robin manner. Initially, the target of both proposals p_a^1 and p_a^2 is the first woman on a's list. For any i, at any point, if p_a^i is rejected by the woman who is ranked k-th on a's list (for any k), then p_a^i goes to the woman ranked $(k+1)$-st on a's list; in case the k-th woman is already the last woman on a's list, then the proposal p_a^i is again made to the first woman on a's list.

A man has three possible levels in status: *basic*, *1-promoted*, or *2-promoted*. Every man a starts out basic with rejection history $r_a = \emptyset$. Let $N(a)$ be the set of all women on a's list. When $r_a = N(a)$, then a becomes 1-promoted. Once he becomes 1-promoted, r_a is reset to the empty set. If $r_a = N(a)$ after a becomes 1-promoted, then a becomes 2-promoted and r_a is reset once again to the empty set. After a becomes 2-promoted, if $r_a = N(a)$, then a gives up.

To illustrate promotions, consider the following example: man a has only two women b_1 and b_2 on his list. He starts as a basic man and makes his proposals p_a^1 and p_a^2 to b_1.

Suppose b_1 rejects both. Then a makes both these proposals to b_2. Suppose b_2 accepts p_a^1 but rejects p_a^2. Then a becomes 1-promoted since $r_a = \{b_1, b_2\}$ now and r_a is reset to $\emptyset$. Note that for a to become 2-promoted, we need r_a to become $\{b_1, b_2\}$ once again. Similarly, a 2-promoted man a gives up only when his rejection history r_a becomes $\{b_1, b_2\}$ *after* he becomes 2-promoted.

– For every $a \in A$, $t_a^1 := t_a^2 := 1$; $r_a := \emptyset$.
$\{r_a$ *is the rejection history of man a;* t_a^i *is the rank of the next woman targeted by the proposal* p_a^i.$\}$
while some $a \in A$ has his proposal p_a^i (i is 1 or 2) not accepted by any woman and he has not given up **do**
  – a makes his proposal p_a^i to the t_a^i-th woman b on his list.
  if b has at most two proposals now (incl. p_a^i) **then**
    – b accepts p_a^i
  else
    – b rejects any of her "least desirable" (see Definition 1) proposals $p_{a'}^j$
    if $t_{a'}^j$ = number of women on the list of a' **then**
      $t_{a'}^j := 1$ $\{$*the round-robin nature of proposing*$\}$
    else
      $t_{a'}^j := t_{a'}^j + 1$
    end if
    – $r_{a'} := r_{a'} \cup \{b\}$
    if $r_{a'}$ = the entire set of neighbors of a' **then**
      if a' is basic **then**
        a' becomes 1-promoted and $r_{a'} := \emptyset$
      else if a' is 1-promoted **then**
        a' becomes 2-promoted and $r_{a'} := \emptyset$
      else if a' is 2-promoted **then**
        a' gives up
      end if
    end if
  end if
end while

Fig. 1. A description of proposals/disposals in our algorithm with one-sided ties

Our algorithm terminates when each $a \in A$ satisfies one of the following conditions: (1) both his proposals p_a^1 and p_a^2 are accepted, (2) he gives up. Note that when (2) happens, the man a must be 2-promoted.

How women decide: A woman can accept up to two proposals. The two proposals can be from the same man. When she currently has less than two proposals, she unconditionally accepts the new proposal. If she has already accepted two proposals and is faced with a third one, then she rejects one of her "least desirable" proposals (see Definition 1 below).

Definition 1. *For a woman b, proposal p_a^i is* superior *to $p_{a'}^{i'}$ if on b's list:*

(1) a ranks better than a'.
(2) a and a' are tied; a is currently 2-promoted while a' is currently 1-promoted or basic.
(3) a and a' are tied; a is currently 1-promoted while a' is currently basic.
(4) a and a' are tied and both are currently basic; moreover, woman b has already rejected one proposal of a while so far she has not rejected any of the proposals of a'.

Let p_a^i be among the three proposals that a woman has and suppose it is not superior to either of the other two proposals. Then p_a^i is a least desirable *proposal.*

The reasoning behind the rules of a woman's decision can be summarized as follows.

– Proposals from higher-ranking men should be preferred, as in the Gale-Shapley algorithm.
– When a woman receives proposals from men who are tied in her list, she prefers the man who has been promoted: a 1-promoted (similarly, 2-promoted) man having been rejected by the entire set of women on his list once (resp. twice) should be preferred, since he is more desperate and deserves to be given a chance.
– When two basic men of the same rank propose to a woman, she prefers the one who has been rejected by her before. The intuition again is that he is more desperate—though he has not been rejected by all women on his list yet (otherwise he would have been 1-promoted).

It is easy to see that the algorithm in Fig. 1 runs in linear time. When it terminates, for each edge $(a, b) \in E$, we set $x_{ab} = 1$ or 0.5 or 0 if the number of proposals that woman b accepts from man a is 2 or 1 or 0, respectively. Let $G' = (A \cup B, E')$ be the subgraph where an edge $e \in E'$ if and only if $x_e > 0$. It is easy to see that in G', the maximum degree of any vertex is 2.

There is a maximum cardinality matching in G' where all degree 2 vertices are matched; moreover, such a matching can be computed in linear time. Let M be such a matching. We first show that M is stable and then prove it is a $22/15$ approximation. Propositions 1 and 2 follow easily from our algorithm and lead to the stability of M.

Proposition 1. *Let woman b reject proposal p_a^i from man a. Then from this point till the end of the algorithm, b has two proposals $p_{a'}^{i'}$ and $p_{a''}^{i''}$ from men a' and a'' (it is possible that $a' = a''$) who rank at least as high as man a on b's list. In particular, if a' (similarly, a'') is tied with man a on the list of b, then at the time a proposed to b:*

1. *if a is ℓ-promoted (ℓ is either 1 or 2), then man a' (resp. a'') has to be $\geq \ell$-promoted.*
2. *if a is basic and his other proposal is already rejected by b, then it has to be the case that either a' (resp. a'') is not basic or b has already rejected his other proposal.*

In the rest of the paper, unless we specifically state the time point, when we say a man is basic/1-promoted/2-promoted, we mean his status when the algorithm terminates.

Proposition 2. *The following facts hold:*

1. *If a man (similarly, a woman) is unmatched in M, then he has at most one proposal accepted by a woman (resp., she receives at most one proposal) during the entire algorithm.*
2. *At the end of the algorithm, every man with less than two proposals accepted is 2-promoted. Furthermore, he must have been rejected by all women on his list as a 2-promoted man.*
3. *If woman b on the list of the man a is unmatched in M, then man a has to be basic and he does not prefer b to the women who accepted his proposals.*

3 Bounding the Size of M

Let OPT be an optimal stable matching. We now need to bound $|\mathsf{OPT}|/|M|$. Whenever we refer to an augmenting path in $M \oplus \mathsf{OPT}$, we mean the path is augmenting with respect to M. Lemma 1 will be crucial in our analysis.

Lemma 1. *Suppose (a, b) and (a', b') are in OPT where man a' is not 2-promoted and a' prefers b to b'. If a is unmatched in M, then (a', b) cannot be in G'.*

Proof. We prove this lemma by contradiction. Suppose $(a', b) \in G'$. If b prefers a' to a, then (a', b) blocks OPT. On the other hand, if b prefers a to a', then this contradicts the fact that b rejected at least one proposal from a (by Proposition 2.1) while b has a proposal from a', who is ranked worse on b's list, at the end of the algorithm since $(a', b) \in G'$.

So the only option possible is that a' and a are tied on b's list. Since a is unmatched in M, it follows from (1)-(2) of Proposition 2 that a has been rejected by b as a 2-promoted man. Since $(a', b) \in G'$, Proposition 1 implies that a' has to be 2-promoted. This however contradicts the lemma statement that a' is not 2-promoted. □

Corollary 1. *There is no length-3 augmenting path $M \oplus \mathsf{OPT}$.*

Proof. If such a path $a - b - a' - b'$ exists (see Fig. 2), then $(a', b) \in G'$ since it is in M. As b' is unmatched in M, a' is basic and prefers b to b' (by Proposition 2.3). This contradicts Lemma 1. □

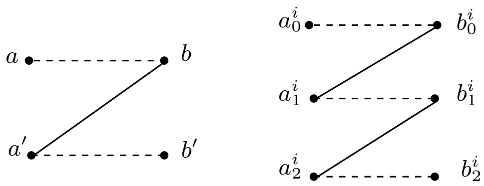

Fig. 2. On the left we have a length-3 augmenting path and on the right we have the length-5 augmenting path ρ_i with respect to M in $M \oplus \mathsf{OPT}$

Let $R = \{\rho_1, \dots, \rho_t\}$ denote the set of length-5 augmenting paths in $M \oplus \mathsf{OPT}$. Lemma 2 lists properties of vertices in a length-5 augmenting path ρ_i (Fig. 2).

Lemma 2. *If $\rho_i = a_0^i - b_0^i - a_1^i - b_1^i - a_2^i - b_2^i$ is a length-5 augmenting path in $M \oplus \mathsf{OPT}$, then*

1. *a_0^i is 2-promoted and has been rejected by b_0^i as a 2-promoted man.*
2. *a_1^i is not 2-promoted and he prefers b_1^i to b_0^i.*
3. *a_2^i is basic and he prefers b_1^i to b_2^i.*
4. *b_1^i is indifferent between a_1^i and a_2^i.*
5. *In G', b_0^i has degree 1 if and only if a_1^i has degree 1.*
6. *In G', b_1^i has degree 1 if and only if a_2^i has degree 1.*

Recall that G' is a subgraph of G and every vertex has degree at most 2 in G'. We form a directed graph H from G' as follows: first orient all edges in the graph G' from A to B; then contract each edge of $M \cap \rho_i$ for $i = 1, \dots, t$. That is, if $\rho_i = a_0^i - b_0^i - a_1^i - b_1^i - a_2^i - b_2^i$, then in H, the edge (a_1^i, b_0^i) gets contracted into a single node (call it x_i) and similarly the edge (a_2^i, b_1^i) gets contracted into a single node (call it y_i) and this happens for all $i = 1, \dots, t$.

Note that (5)-(6) of Lemma 2 imply that $\deg_H(x_i), \deg_H(y_i) \in \{0, 2\}$ for $1 \le i \le t$, where $\deg_H(v) = 2$ means in H in-degree(v) = out-degree$(v) = 1$. The following lemma rules out the possibility of certain arcs in H.

Lemma 3. *For any $1 \le i, j \le t$, there is no arc from y_i to x_j in H.*

Proof. Suppose there is an arc in H from y_i to x_j for some $1 \le i, j, \le t$. That is, G' contains the edge (a_2^i, b_0^j). Since the woman b_2^i is unmatched, we use Proposition 2.3 to conclude that a_2^i is basic and he prefers b_0^j to b_2^i. This contradicts Lemma 1, by substituting $a = a_0^j$, $b = b_0^j$, $a' = a_2^i$, and $b' = b_2^i$. □

We now define a "good path" in H. In H, let us refer to the x-nodes and y-nodes as *red* and let the other vertices be called *blue*.

Definition 2. *A directed path in H is* good *if its end vertices are blue while all its intermediate vertices are red. Also, we assume there is at least one intermediate vertex in such a path.*

Lemma 3 implies that every good path looks as follows: a blue man, followed by some x-nodes (possibly none), followed by some y-nodes (possibly none), and a blue woman.

For any y-node y_i, if $\deg_H(y_i) \neq 0$, using Lemma 3 we can conclude that y_i is either in a cycle of y-nodes or in a good path. In other words, there are only 3 possibilities in H for each y_i: (1) y_i is an isolated node, (2) y_i is in a cycle of y-nodes, (3) y_i is in a good path.

We next define a *critical arc* in H. We will use critical arcs to show that H has enough good paths. Since the endpoints of a good path are vertices outside R, this bounds $|\mathsf{OPT}|/|M|$.

Definition 3. *Call an arc (x_i, z) in H* critical *if either a_1^i prefers z to b_1^i or $z = b_1^i$.*

In case z is a red node, let w be the woman in z – in Definition 3, we mean either $w = b_1^i$ or a_1^i prefers w to b_1^i. We show (via Lemma 4 and Claim 1) that every critical arc is in a distinct good path. It follows from Lemma 4 that every good path has at most one critical arc. Lemma 5 is the main technical lemma here. It shows there are *enough* critical arcs in H.

Lemma 4. *For any i, if (x_i, z) is critical, then z is not an x-node, i.e., $z \neq x_j$ for any j.*

Proof. For any $1 \leq i, j \leq t$, if a proposal of a_1^i is accepted by a woman w that a_1^i prefers to b_1^i, then we need to show that w cannot be b_0^j. Suppose $w = b_0^j$ for some j. In the first place, $j \neq i$ since we know a_1^i prefers b_1^i to b_0^i (by Lemma 2.2). We know a_1^i is not 2-promoted by Lemma 2.2. We now contradict Lemma 1, by substituting $a = a_0^j$, $b = b_0^j$, $a' = a_1^i$, and $b' = b_1^i$. □

Claim 1. *Every critical arc is in some good path and every pair of good paths is vertex-disjoint.*

Lemma 5. *In the graph H, the following statements hold:*

(1) If y_i is an isolated node, then there exists a critical arc (x_i, z) in H.
(2) If (y_i, y_j) is an arc, then there exists a critical arc (x_i, z) or a critical arc (x_j, z') (or both).

Proof. We first show part (1) of this lemma. Suppose y_i is an isolated node in H. By parts (2) and (6) of Lemma 2, the woman b_1^i accepts both proposals from a_2^i and she rejects a_1^i at least once. Suppose b_1^i rejects a_1^i exactly once. This means that one proposal of a_1^i (other than the one accepted by b_0^i) has been accepted by a woman w that a_1^i prefers to b_1^i. That is, there is a critical arc (x_i, z) in H.

So suppose b_1^i rejects a_1^i more than once. Then either a_1^i has both of his proposals rejected by b_1^i while he was basic, or he was rejected by b_1^i as a 1-promoted man. In both cases we have a contradiction to Proposition 1 since b_1^i has accepted both proposals from a_2^i, who is basic and is tied with a_1^i.

We now show part (2) of this lemma. Suppose a_1^i prefers b_1^i to the women accepting his proposals and a_1^j prefers b_1^j to the women accepting his proposals. Note that this includes the possibility that both of a_1^i's proposals are accepted by b_0^i and the possibility that both of a_1^j's proposals are accepted by b_0^j. The first observation is that a_1^j could *not* have proposed to b_1^j as a 1-promoted man, as it would contradict Proposition 1 otherwise (recall a_2^j is basic and a_1^j, a_2^j are tied on the list of b_1^j). For the same reason, a_1^i never proposed to b_1^i as a 1-promoted man.

Since we assumed that a_1^j prefers b_1^j to the women accepting his proposals and he never proposed to b_1^j as a 1-promoted man, it must be the case that both of his proposals were rejected by b_1^j when he was still basic. The edge $(a_2^i, b_1^j) \in G'$ since (y_i, y_j) is in H. We now claim this implies a_2^i is tied with a_1^j on the list of b_1^j. If b_1^j prefers a_2^i to a_1^j, then (a_2^i, b_1^j) blocks OPT, since Proposition 2.3 states that a_2^i prefers b_1^j to b_2^i. Now suppose b_1^j prefers a_1^j to a_2^i. Since a_1^j prefers b_1^j to b_0^j (by Lemma 2.2), he must have been rejected by b_1^j before he proposed to b_0^j, implying a contradiction to Proposition 1.

We also know that a_1^j is tied with a_2^j on the list of b_1^j (by Lemma 2.4) and that a_2^i is basic. Since we know that both of a_1^j's proposals were rejected by b_1^j, it has to be the case that while b_1^j accepted one proposal of a_2^i, she rejected his other proposal (by Proposition 1.2). This other proposal of a_2^i was at some point accepted by b_1^i. So it follows that b_1^j *ranks higher than* b_1^i *on the list of* a_2^i*, furthermore,* b_1^i *never rejects a proposal from* a_2^i*.*

Since we assumed that a_1^i prefers b_1^i to the women accepting his proposals and he never proposed to b_1^i as a 1-promoted man, it follows that both of his proposals were rejected by b_1^i when he was basic. This, combined with the fact that b_1^i never rejects a proposal from a_2^i, contradicts Proposition 1.2. Thus either one proposal of a_1^i has been accepted by a woman w that is b_1^i or better than b_1^i in a_1^i's list or one proposal of a_1^j has been accepted by a woman w' that a_1^j prefers to b_1^j. Hence there is a critical arc (x_i, z) or a critical arc (x_j, z') in H. □

We define a function $f : [t] \to \mathcal{P}$, where $\mathcal{P}$ is the set of all good paths in H and $[t] = \{1, \ldots, t\}$. For any $i \in [t]$, $f(i)$ is defined as follows:

(1) Suppose y_i is isolated. Then let $f(i) = p$, where $p \in \mathcal{P}$ contains the critical arc (x_i, z). We know there is such an arc in H by Lemma 5.1.
(2) Suppose y_i belongs to a cycle C of y-nodes, so there is an arc (y_i, y_j) in C. We know H has a critical arc (x_i, z) or (x_j, z') (by Lemma 5.2). Then let $f(i) = p$, where $p \in \mathcal{P}$ contains such a critical arc.
(3) Suppose y_i belongs to a good path p'. If y_i is the *last* y-node in p', then let $f(i) = p'$. Otherwise there is an arc (y_i, y_j) in p' and we know H has a critical arc (x_i, z) or (x_j, z') (by Lemma 5.2). Then let $f(y_i) = p$, where $p \in \mathcal{P}$ contains such a critical arc.

For any $p \in \mathcal{P}$, let $\mathsf{cost}(p) =$ the number of pre-images of p under f. We now show a charging scheme that distributes $\mathsf{cost}(p)$, for each $p \in \mathcal{P}$, among the vertices in G so that the following properties hold. Let $Q = (M \oplus \mathsf{OPT}) \setminus R$.

(I) Each $v \in A \cup B$ is assigned a charge of at most 1.5 and the sum of all vertex charges is t.
(II) Every vertex that is assigned a positive charge must be matched in M and is in some $q \in Q$. Moreover, if $q \in Q$ is an augmenting path on $2\ell_q + 3 \geq 7$ edges, then at most $2\ell_q$ vertices in q will be assigned a positive charge.

Note that a vertex not assigned a positive charge has charge 0 by default.

Suppose there is such a charging scheme, we now show why this implies $|\mathsf{OPT}|/|M|$ is at most $22/15$. Let $q \in Q$ be an alternating cycle/path on $2\ell_q$ edges or an alternating path on $2\ell_q - 1$ edges (with ℓ_q edges from M) or an augmenting path on $2\ell_q + 3 \geq 7$ edges. It follows from (I) and (II) that the total charge assigned to vertices in q is at most $1.5(2\ell_q) = 3\ell_q$, i.e., if the vertices in q are being charged for c_q augmenting paths of length-5 in $M \oplus \mathsf{OPT}$, then $c_q \leq 3\ell_q$.

Since $\sum_{q \in Q} c_q = t$, all the paths in R are paid for in this manner. So we have:

$$|\mathsf{OPT}| = \sum_{q \in Q} (|\mathsf{OPT} \cap q| + 3c_q) \quad \text{and} \quad |M| = \sum_{q \in Q} (|M \cap q| + 2c_q),$$

because there are $3c_q$ edges of OPT in the c_q augmenting paths of length-5 covered by q and $2c_q$ edges of M in the c_q augmenting paths of length-5 covered by q. Thus we have:

$$\frac{|\mathsf{OPT}|}{|M|} \le \max_{q \in Q} \frac{|\mathsf{OPT} \cap q| + 3c_q}{|M \cap q| + 2c_q} \le \max_{l_q \ge 2} \frac{10\ell_q + 2}{7\ell_q + 1} \le \frac{22}{15}.$$

We use $(\sum_i s_i)/(\sum_i t_i) \le \max_i s_i/t_i$ in the first inequality. The above ratio gets maximized for any $q \in Q$ by setting c_q to its largest value of $3\ell_q$ and letting q be an augmenting path so that $|\mathsf{OPT} \cap q| > |M \cap q|$.

This yields $(\ell_q + 2 + 3 \cdot 3\ell_q)/(\ell_q + 1 + 2 \cdot 3\ell_q)$, where $|q| = 2\ell_q + 3 \ge 7$. Note that since augmenting paths in Q have length ≥ 7, this forces $\ell_q \ge 2$ in this ratio. Setting $\ell_q = 2$ maximizes the ratio $(10\ell_q + 2)/(7\ell_q + 1)$. Thus our upper bound is 22/15.

Ensuring properties (I) and (II). We now show a charging scheme that defines a function $\mathsf{charge} : A \cup B \to [0, 1.5]$ such that $\sum_u \mathsf{charge}(u) = \sum_{p \in \mathcal{P}} \mathsf{cost}(p) = t$, where the sum is over all $u \in A \cup B$. We start with $\mathsf{charge}(u) = 0$ for all $u \in A \cup B$. Our task now is to reset charge values for some vertices so that properties (I) and (II) are satisfied.

Each $p \in \mathcal{P}$ is one of the following three types: (1) *type-1* path: this has no x-nodes, (2) *type-2* path: this has no y-nodes, and (3) *type-3* path: this has both x-nodes and y-nodes. The following lemma will be useful later in our analysis.

Lemma 6. *For any $p \in \mathcal{P}$ and $k = 1, 2, 3$, if p is a type-k path, then* $\mathsf{cost}(p) \le k$.

Consider any $p \in \mathcal{P}$. Though p was defined as a good path in H, we now consider p as a path in the graph G'. Since each intermediate node of p is an edge of M, p is an alternating path in G'. Let a_p (man) and b_p (woman) be the endpoints of the path p.

If both a_p and b_p are unmatched in M, then the path p becomes an augmenting path in G'. Since M is a maximum cardinality matching in G', there cannot be an augmenting path with respect to M in G'; hence at least one of a_p, b_p has to be matched in M.

Case 1. Suppose both a_p and b_p are matched. If p is a type-1 path, then reset $\mathsf{charge}(b_p) = \mathsf{cost}(p)$, i.e., the entire cost associated with p is assigned to the woman who is an endpoint of p. If p is a type-k path for $k = 2$ or 3, then reset $\mathsf{charge}(a_p) = \mathsf{charge}(b_p) = \mathsf{cost}(p)/2$.

Case 2. Suppose exactly one of a_p, b_p is matched: call the matched vertex s_p and the unmatched vertex u_p. Construct the alternating path with respect to M in G' with u_p as the starting vertex. The vertex u_p has degree 1 since it is unmatched, also the maximum degree of any vertex in G' is 2. So there is only one such alternating path in G'. This path continues till it encounters a degree 1 vertex, call it r_p.

Note that r_p has to be matched, otherwise there is an augmenting path in G' between u_p and r_p. Since r_p is reached via a matched edge on this path, both u_p and r_p are either in A or in B. In other words, exactly one of r_p, s_p (recall $s_p = \{a_p, b_p\} \setminus \{u_p\}$) is a woman. If p is a type-1 path, then we reset $\mathsf{charge}(w) = \mathsf{cost}(p)$, where w is the woman in $\{r_p, s_p\}$. If p is a type-k path, where $k = 2$ or 3, then we reset $\mathsf{charge}(s_p) = \mathsf{charge}(r_p) = \mathsf{cost}(p)/2$. This concludes the description of our charging scheme.

References

1. Askalidis, G., Immorlica, N., Kwanashie, A., Manlove, D.F., Pountourakis, E.: Socially stable matchings in the hospitals / residents problem. In: Dehne, F., Solis-Oba, R., Sack, J.-R. (eds.) WADS 2013. LNCS, vol. 8037, pp. 85–96. Springer, Heidelberg (2013)
2. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. American Math. Monthly 69, 9–15 (1962)
3. Gale, D., Sotomayer, M.: Some remarks on the stable marriage problem. Discrete Applied Mathematics 11, 223–232 (1985)
4. Gusfield, D., Irving, R.W.: The Stable Marriage Problem: Structure and Algorithms. MIT Press, Boston (1989)
5. Halldórsson, M.M., Iwama, K., Miyazaki, S., Yanagisawa, H.: Randomized approximation of the stable marriage problem. Theoretical Computer Science 325(3), 439–465 (2004)
6. Irving, R.W.: Stable marriage and indifference. Discrete Applied Mathematics 48, 261–272 (1994)
7. Irving, R.W., Manlove, D.F.: Approximation algorithms for hard variants of the stable marriage and hospitals/residents problems. Journal of Combinatorial Optimization 16(3), 279–292 (2008)
8. Iwama, K., Manlove, D.F., Miyazaki, S., Morita, Y.: Stable marriage with incomplete lists and ties. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 443–452. Springer, Heidelberg (1999)
9. Iwama, K., Miyazaki, S., Yamauchi, N.: A 1.875-approximation algorithm for the stable marriage problem. In: 18th SODA, pp. 288–297 (2007)
10. Iwama, K., Miyazaki, S., Yanagisawa, H.: A 25/17-approximation algorithm for the stable marriage problem with one-sided ties. In: de Berg, M., Meyer, U. (eds.) ESA 2010, Part II. LNCS, vol. 6347, pp. 135–146. Springer, Heidelberg (2010)
11. Király, Z.: Better and simpler approximation algorithms for the stable marriage problem. Algorithmica 60(1), 3–20 (2011)
12. Király, Z.: Linear time local approximation algorithm for maximum stable marriage. Algorithms 6(3), 471–484 (2013)
13. Knuth, D.: Mariages stables et leurs relations avec d'autre problèmes. Les Presses de l'université de Montréal (1976)
14. Manlove, D.: Algorithmics of Matching Under Preferences. World Scientific Publishing Company Incorporated (2013)
15. Manlove, D.F., Irving, R.W., Iwama, K., Miyazaki, S., Morita, Y.: Hard variants of stable marriage. Theoretical Computer Science 276(1-2), 261–279 (2002)
16. McDermid, E.: A 3/2 approximation algorithm for general stable marriage. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part I. LNCS, vol. 5555, pp. 689–700. Springer, Heidelberg (2009)
17. Paluch, K.: Faster and simpler approximation of stable matchings. In: Solis-Oba, R., Persiano, G. (eds.) WAOA 2011. LNCS, vol. 7164, pp. 176–187. Springer, Heidelberg (2012)
18. Roth, A., Sotomayor, M.: Two-sided matching: a study in game-theoretic modeling and analysis. Cambridge University Press (1992)
19. Teo, C.-P., Sethuraman, J., Tan, W.P.: Gale-Shapley stable marriage problem revisited: strategic issues and applications. In: Cornuéjols, G., Burkard, R.E., Woeginger, G.J. (eds.) IPCO 1999. LNCS, vol. 1610, pp. 429–438. Springer, Heidelberg (1999)
20. Vande Vate, J.: Linear Programming brings marital bliss. Operation Research Letters 8, 147–153 (1989)
21. Yanagisawa, H.: Approximation algorithms for stable marriage problems. Ph.D. Thesis, Kyoto University (2007)