Noname manuscript No.
(will be inserted by the editor)

An improved approximation algorithm for the stable marriage problem with one-sided ties

Chien-Chung Huang · Telikepalli Kavitha

Received: date / Accepted: date

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.

A preliminary version of this work appears in the 17th Conference on Integer Programming and Combinatorial Optimization (IPCO 2014). Part of this work was done when the second author visited the Max-Planck-Institut für Informatik, Saarbrücken under the IMPECS program.

Chien-Chung Huang
Chalmers University of Technology, Sweden
E-mail: huangch@chalmers.se

Telikepalli Kavitha
Tata Institute of Fundamental Research, India
E-mail: kavitha@tcs.tifr.res.in

Keywords Stable Matching · Approximation Algorithms

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 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:

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

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 to keep the graph G' simple. 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 $|\mathsf{OPT}|/|M'| \leq 10/7 \approx 1.4286$ 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."

We present our algorithm for one-sided ties in Section 2 its analysis in Section 2.1. Section 3 has our algorithm for instances with two-sided ties, where each tie has length 2.

2 Algorithm for one-sided ties

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. In the following, we show that M is stable and it is a $22/15$ approximation, thereby proving Theorem 1.

2.1 Analysis of the above algorithm

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.*

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.*

Lemma 1 *The matching M is stable in $G = (A \cup B, E)$.*

Proof Let $(a, b) \in E \setminus M$. Suppose a is unmatched in M. Then by (1)-(2) of Proposition 2, a has to be 2-promoted and all women on his list rejected at least one of his proposals. As this includes b, we know from Proposition 1 that b has two proposals from men ranking at least as high as a on her list and she is matched in M with one such man. So (a, b) does not block M.

Now suppose a is matched in M. Let $(a, b') \in M$. It follows from our algorithm that man a must have been rejected by all women that rank higher than b' on his list. Let b be such a woman. By Proposition 1, till the end of the algorithm, b has two proposals from men that rank at least as high as a on her list and b is matched in M with one such man. So (a, b) does not block M. □

In the rest of the discussion, 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.

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 2 will be crucial in our analysis.

Lemma 2 *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 2. □

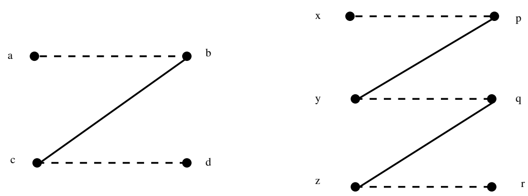

Fig. 2 On the left we have a length-3 augmenting path and on the right we have the length-5 augmenting path ρ_i with respect to M in $M \oplus \mathsf{OPT}$.

Let $R = \{\rho_1, \ldots, \rho_t\}$ denote the set of length-5 augmenting paths in $M \oplus \mathsf{OPT}$. Lemma 3 lists properties of vertices in a length-5 augmenting path ρ_i (see Fig. 2).

Lemma 3 *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.*

Proof It is easy to see that (1) and (3) follow from Proposition 2.

We now show (2). Suppose a_1^i is 2-promoted. To become 2-promoted, a_1^i must have been rejected by b_1^i as a 1-promoted man. By Proposition 1, as $(a_2^i, b_1^i) \in G'$, either b_1^i prefers a_2^i to a_1^i, or they are tied in her preference list and a_2^i is also at least

1-promoted. In the former case, (a_2^i, b_1^i) blocks OPT while in the latter case, (3) is contradicted. This proves the first part of (2). For the second part of (2), suppose a_1^i prefers b_0^i to b_1^i. Then b_0^i is indifferent between a_0^i and a_1^i, otherwise either (a_0^i, b_0^i) blocks M or (a_1^i, b_0^i) blocks OPT. Since a_0^i is unmatched in M, by Proposition 2.2, a_0^i must have been rejected by b_0^i as a 2-promoted man. Then Proposition 1 implies that a_1^i is also 2-promoted, which contradicts the first part of (2). This completes the proof of (2).

To show (4), observe that if b_1^i is not indifferent between a_1^i and a_2^i, then either (a_1^i, b_1^i) blocks M or (a_2^i, b_1^i) blocks OPT.

We now show (5). Since a_1^i is not 2-promoted, by (1)-(2) of Proposition 2, both of his proposals get accepted in the algorithm. As a_0^i has been rejected by b_0^i as a 2-promoted man, Proposition 1 states that b_0^i has two proposals till the end of the algorithm. So either both proposals made by a_1^i get accepted by b_0^i (then both a_1^i and b_0^i have degree 1 in G') or exactly one proposal of a_1^i is accepted by b_0^i, in which case, a_1^i's other proposal is accepted by some other woman and b_0^i has received a proposal from some other man (so both a_1^i and b_0^i have degree 2 in G').

We now show (6). Since a_2^i is not 2-promoted, by (1)-(2) of Proposition 2, both of his proposals get accepted in the algorithm. Note that by (2), a_1^i must have been rejected by b_1^i, so b_1^i has at least two proposals till the end of the algorithm. So either both proposals made by a_2^i get accepted by b_1^i (then both a_2^i and b_1^i have degree 1 in G') or exactly one proposal of a_2^i is accepted by b_1^i, in which case, a_2^i's other proposal is accepted by some other woman and b_1^i has received a proposal from some other man (so both a_2^i and b_1^i have degree 2 in G'). □

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 $i = 1, \ldots, t$. 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 $i = 1, \ldots, t$.

Note that (5)-(6) of Lemma 3 imply that $\deg_H(x_i), \deg_H(y_i) \in \{0, 2\}$ for $1 \leq i \leq t$, 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 4 *For any $1 \leq i, j \leq t$, 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 $1 \leq i, j, \leq t$. 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 2, 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 4 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 4 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 Lemmas 5 and 6) that every critical arc is in a distinct good path. It follows from Lemma 5 that every good path has at most one critical arc. Lemma 7 is the main technical lemma here. It shows there are *enough* critical arcs in H.

Lemma 5 *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 3.2). We know a_1^i is not 2-promoted by Lemma 3.2. We now contradict Lemma 2, by substituting $a = a_0^j$, $b = b_0^j$, $a' = a_1^i$, and $b' = b_1^i$. $\square$

Lemma 6 *Every critical arc is in some good path and every pair of good paths is vertex-disjoint.*

Proof Let (x_i, z) be a critical arc in H. We know from Lemma 5 that z is either a blue woman w or a y-node y_j. Suppose z is a blue woman w. Then w becomes one endpoint of a good path containing (x_i, z). Since $\deg_H(x_i) \neq 0$, it has to be 2. So there is an arc (u_0, x_i) in H and we know from Lemma 4 that u_0 cannot be a y-node. Hence u_0 is either a blue man m (in which case we have the desired good path) or $u_0 = x_{i'}$ for some i' – then there is an arc $(u_1, x_{i'})$ in H and thus chasing these arcs backwards, we have to finally reach a blue man m. This yields a good path $m - \cdots - x_{i'} - x_i - w$ that contains (x_i, w).

Suppose $z = y_j$ for some j. Now we also chase arcs forwards, starting from y_j. Following these arcs, we have to reach a blue woman w since there is no arc from a y-node to a x-node (by Lemma 4). So we get a good path $m - \cdots - x_{i'} - x_i - y_j - y_{j'} - \cdots - w$ containing (x_i, y_j).

We now show that any pair of good paths has to be vertex-disjoint. As every vertex in G' has degree at most 2, the intermediate vertices in any two good paths are disjoint. Suppose vertex u is an endpoint of two different good paths. By definition of good paths, u is then incident on two edges of G', neither of which is in M. This contradicts the assumption that all degree 2 vertices of G' are matched in M. $\square$

Lemma 7 *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 3, 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 3.2), 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 3.4) 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 7.1.
(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 7.2). 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 7.2). 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. We will show later that such a charging scheme exists. For now, we 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:

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

We use $(\sum_i s_i)/(\sum_i t_i) \leq \max_i s_i/t_i$ 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 $|q| = 2\ell_q + 3 \geq 7$. Note that since augmenting paths in Q have length ≥ 7, this forces $\ell_q \geq 2$ 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 and this proves Theorem 1.

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

Proof Let $p \in \mathcal{P}$ be a type-1 path. So p has the form $m - y_j - \cdots - y_{j'} - w$, where m and w are blue vertices. There is only one element in $[t]$ whose f-image is p and that is the index j', where $y_{j'}$ is the last y-node in p. Since p has no x-nodes, there is no critical arc in p. So p cannot be the f-image of any other index in $[t]$. Hence in this case, we have $\mathsf{cost}(p) = 1$.

Let $p \in \mathcal{P}$ be a type-2 path. So p has the form $m - x_i - \cdots - x_{i'} - w$, where m and w are blue vertices. The arc $(x_{i'}, w)$ could be a critical arc and no other arc can be critical (by Lemma 5). If $(x_{i'}, w)$ is not critical, then p cannot be the f-image of any element in $[t]$ and $\mathsf{cost}(p) = 0$.

If $(x_{i'}, w)$ is critical, then p can be the f-image of at most 2 elements in $[t]$. These are the index i' and possibly another index i'' if there exists an arc $(y_{i''}, y_{i'})$ in a good path or a cycle of y-nodes. No other index in $[t]$ can be mapped to p. Thus we have $\mathsf{cost}(p) \leq 2$ here.

Let $p \in \mathcal{P}$ be a type-3 path. So p has the form $m - x_i - \cdots - x_{i'} - y_j - \cdots - y_{j'} - w$, where m and w are blue vertices. It follows from the discussion in the earlier cases that there are at most 3 elements in $[t]$ that can get mapped to p. These are j' (the index of the last y-node), i' (if $(x_{i'}, y_j)$) is critical), and index i'' (if $(x_{i'}, y_j)$ is critical and there is an arc $(y_{i''}, y_{i'})$ in a good path or a cycle of y-nodes). Thus we have $\mathsf{cost}(p) \leq 3$ here. □

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 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, either both u_p and r_p are in A or they both are in B. In other words, exactly one of r_p, s_p is a woman (recall $s_p = \{a_p, b_p\} \setminus \{u_p\}$).

– 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.

Lemma 9 *The function* $\mathsf{charge}(\cdot)$ *satisfies properties (I) and (II) stated earlier.*

Proof It is immediate from Lemma 8 that $\mathsf{charge}(u) \leq 1.5$ for all $u \in A \cup B$. Consider any $p \in \mathcal{P}$ and let its endpoints be a_p and b_p. It follows from Lemma 6 that every path in $\mathcal{P}$ has its own distinct endpoints. When both a_p and b_p are matched in M, then no other vertex gets charged for p. When only one of them is matched, then another vertex r_p could also get charged – this is the degree 1 endpoint of the alternating path L_p from the unmatched vertex in $\{a_p, b_p\}$. Observe that r_p cannot be the endpoint of any $p' \in \mathcal{P}$ since each of $a_{p'}$ and $b_{p'}$ need to have an unmatched edge incident on them, which is not true for r_p.

Also, $r_p \neq r_{p'}$ for any other $p' \in \mathcal{P}$: if L_p and $L_{p'}$ are the alternating paths corresponding to distinct paths $p, p' \in \mathcal{P}$, since the endpoints of L_p and $L_{p'}$ are degree 1 vertices while every intermediate vertex has degree 2 in L_p and in $L_{p'}$, the paths L_p and $L_{p'}$ have to be vertex-disjoint since the maximum degree in G' is 2. Thus any vertex gets charged for at most one $p \in \mathcal{P}$. Hence $\sum_{u \in A \cup B} \mathsf{charge}(u) = \sum_{p \in \mathcal{P}} \mathsf{cost}(p) = t$. Thus property (I) holds.

We next show property (II). It is also straightforward from our method of resetting charge values that it is only vertices that are matched in M that get assigned positive charge. Also each such vertex is outside R since the edges of $M \cap R$ are contracted to red nodes in H while the vertices that get assigned positive charge here are a_p, b_p which are blue vertices and r_p which is a degree 1 vertex in G' that is reachable by an alternating path from an unmatched vertex – hence r_p cannot be any of $b^i_0, a^i_1, b^i_1, a^i_2$ by parts (5) and (6) of Lemma 3. Summarizing, each vertex with positive charge is a vertex matched in M that belongs to some $q \in Q$.

What is left to show is that if $q \in Q$ is an augmenting path on $2\ell + 3 \geq 7$ edges, then at most 2ℓ vertices of q get assigned a positive charge. Let q be the path $\alpha_0 - \beta_0 - \alpha_1 - \cdots - \beta_\ell - \alpha_{\ell+1} - \beta_{\ell+1}$ where the edges $(\alpha_i, \beta_{i-1}) \in M$ for $i = 1, \ldots, \ell + 1$. As our charging scheme assigns positive charge only to matched vertices, both α_0 and β_{l+1} have charge 0. We will now show that neither β_0 nor $\alpha_{\ell+1}$ can be assigned positive charge. We first show that neither β_0 nor $\alpha_{\ell+1}$ can be the vertex r_p for any $p \in \mathcal{P}$. We claim that if β_0 has degree 1 in G', then $M(\beta_0) = \alpha_1$ also has degree 1; similarly, if $\alpha_{\ell+1}$ has degree 1 in G', then $M(\alpha_{\ell+1}) = \beta_\ell$ also has degree 1. Assuming this claim, it is easy to see that neither β_0 nor $\alpha_{\ell+1}$ can be the degree 1 vertex which is the matched endpoint of an alternating path starting from an unmatched vertex.

We now prove the above claim. We know that β_0 must have rejected at least one proposal from the unmatched man α_0, so β_0 must have 2 proposals at the end of the algorithm. We know that β_0 has at least one proposal from α_1 since (α_1, β_0) is in G'. So if β_0 has degree 1 in G', then both her proposals must be from α_1, in other words, α_1 also has degree 1 in G'. Similarly, $\alpha_{\ell+1}$ has an unmatched woman $\beta_{\ell+1}$ on his preference list – so he must have both of his proposals accepted at the end of the algorithm. We know that β_ℓ has accepted at least one of these proposals since $(\alpha_{\ell+1}, \beta_\ell)$ is in G'. So if $\alpha_{\ell+1}$ has degree 1 in G', then both of his proposals have been accepted by β_ℓ, in other words, β_ℓ also has degree 1 in G'.

We now show that if $\beta_0 = b_p$ for some $p \in \mathcal{P}$, then $\mathsf{cost}(p) = 0$. If $\beta_0 = b_p$, then we cannot have the arc (y_i, β_0) in H for any y-node y_i since such an arc in H implies the edge (a_2^i, β_0) is in G' and we contradict Lemma 2 by substituting $a = \alpha_0$, $b = \beta_0$, $a' = a_2^i$, and $b' = b_2^i$. So if $\beta_0 = b_p$ for some $p \in \mathcal{P}$, then p has no y-nodes (it is a type-2 path) and hence has the structure: $a_p - x_i - \cdots - x_j - \beta_0$. We now argue that there cannot be a critical arc in p. The only candidate for a critical arc here is (x_j, β_0) and for this to be critical, we need a_1^j to prefer β_0 to b_1^j. However this contradicts Lemma 2 by substituting $a = \alpha_0$, $b = \beta_0$, $a' = a_1^j$, and $b' = b_1^j$. Thus p cannot the f-image of any element in $[t]$. Hence we have $\mathsf{charge}(\beta_0) = 0$.

We now show that if $\alpha_{\ell+1} = a_p$ for some $p \in \mathcal{P}$, then $\mathsf{charge}(a_p) = 0$. If $\alpha_{\ell+1} = a_p$, then we cannot have the arc $(\alpha_{\ell+1}, x_i)$ in H for any x-node x_i since such an arc in H implies the edge $(\alpha_{\ell+1}, b_0^i)$ is in G' and we contradict Lemma 2 by substituting $a = a_0^i$, $b = b_0^i$, $a' = \alpha_{\ell+1}$, and $b' = \beta_{\ell+1}$. So if $\alpha_{\ell+1} = a_p$ for some $p \in \mathcal{P}$, then p has no x-nodes, i.e., it is a type-1 path. Recall that for a type-1 path p, the entire cost of p was assigned completely to the *woman* associated with p, i.e., to b_p if this is a matched vertex, else to r_p. Thus $\mathsf{charge}(\alpha_{\ell+1}) = 0$. This finishes the proof of Lemma 9. □

3 Algorithm for two-sided ties of length 2

We now move to the domain of two-sided ties, i.e., both men and women in G may have ties in their preference lists; however each tie has length 2. We now present our algorithm to compute a stable matching M' in G such that $|\mathsf{OPT}|/|M'| \leq 10/7$, where OPT is a maximum cardinality stable matching in G.

This algorithm bears similarity to our previous algorithm, however there are some important differences as well. We inherit the notions of basic/1-promoted/2-promoted levels in status for men from our previous algorithm (see the early part of Section 2). Our algorithm describing how men propose and women decide is given in Fig. 3.

How men propose. As before, every $a \in A$ has two proposals p_a^1 and p_a^2 and a makes each of these proposals to the women on his preference list in a round-robin manner. We again use a rejection history r_a to record the women in a's list who have rejected his proposals in his current status. We now describe how proposals are made to the two women of the same rank in a's list.

Let t_a^i be the rank of the next target woman for proposal p_a^i. If there is exactly one woman of rank t_a^i who is not in the rejection history r_a, then proposal p_a^i is made to this woman. If there are two women b and b' of rank t_a^i and neither is in the rejection history r_a, then one of them, say b, is chosen arbitrarily. In our algorithm, it is possible that b *bounces* or *forwards* a's proposal to b' without actually rejecting him. This step is described in detail below.

How women decide. A woman accepts a new proposal unconditionally if she currently has less than two proposals. Suppose a woman b receives a third proposal p_a^i when she already has two proposals, call these $p_{a'}^{i'}$ and $p_{a''}^{i''}$. Recall that in the algorithm in Fig. 1, b rejected the *least desirable* among these three proposals. In our current algorithm, b does not do this step right away. She first runs the *bounce* step and if this is not successful, then she runs the *forward* step.

- The bounce step: this works for $\alpha \in \{a, a', a''\}$ if there is a woman b' tied with b in α's preference list such that b' has less than two proposals. The woman b checks if the bounce step works for any of a, a', a'' (in no particular order). If so, then b *bounces* that man's proposal to b' and the bounce step is successful.
- The forward step: for this to work, it is necessary that among the 3 proposals that b currently has, two of them are from the same man α. If there is a woman b' tied with b in α's preference list such that $b' \notin r_\alpha$, then the forward step is successful, i.e., b *forwards* the proposal p_α^1 to b' and the algorithm continues by letting man α make proposal p_α^1 to b'.

It is important to note that in both the bounce step and the forward step, woman b does not actually reject α, i.e., b is not added to the rejection history of r_α. If neither the bounce step nor the forward step works for any of a, a', a'', then b rejects any of the least desirable proposals among $p_a^i, p_{a'}^{i'}, p_{a''}^{i''}$. Here the definition of "least desirable" is the same as in Definition 1, except that part (4) of this definition does not apply here.

In the special case when a, a', a'' are tied in b's preference list, then it has to be the case that two of these 3 men are the same, i.e., $a = a'$ or $a' = a''$ or $a = a''$, since each tie has length 2; in this case we let b reject one of the two proposals from the same man.

We now describe what it means for a proposal to be rejected. Say the proposal p_α^j is rejected by woman b. Then the following steps are run:

- b is added to the rejection history r_α of α.
- In case $r_\alpha = N(\alpha)$ (the entire neighborhood of α in G)
  - if α is basic then he becomes 1-promoted; $r_\alpha = \emptyset$
  - if α is 1-promoted then he becomes 2-promoted; $r_\alpha = \emptyset$
  - if α is 2-promoted then he gives up.

This finishes the description of our algorithm. Since a bounce step causes a woman who currently has less than two proposals to receive a new proposal, it is easy to see that the number of bounce steps is at most $2|B|$. The number of reject steps is also linear. Lemma 10 bounds the total number of forward steps in the algorithm.

Lemma 10 *The total number of forward steps in the entire algorithm is* $O(|E|)$.

Proof A forward step implies a man α has to transfer his proposal p_α^1 from woman b to woman b' (who already has 2 proposals). Currently both p_α^1 and p_α^2 are made to b.

For all men $a \in A$, $t_a^1 := t_a^2 := 1$; $r_a := \emptyset$.
{r_a *is the rejection history of man a;* t_a^i *is the rank of the next woman targeted by proposal* p_a^i.}
while some man $a \in A$ has his proposal p_a^i not accepted by a woman and he has not given up **do**
  if there is a woman b of rank t_a^i on a's list and $b \notin r_a$ **then**
    – propose(p_a^i,b)
  else
    – if t_a^i is the worst rank on a's list, then set $t_a^i = 1$; else set $t_a^i = t_a^i + 1$
  end if
end while

Procedure propose(p_a^i, b)
– a makes his proposal p_a^i to b
if b has at most two proposals now (incl. p_a^i) **then**
  – b accepts p_a^i
else
  – let these three proposals with b be p_a^i, $p_{a'}^{i'}$ and $p_{a''}^{i''}$
  if the *bounce* step works for some $\alpha \in \{a, a', a''\}$ **then**
    – propose(p_α^j, b')
    {$\exists b'$ *tied with b on* α*'s list with less than 2 proposals – now* b' *receives* $p_\alpha^j \in \{p_a^i, p_{a'}^{i'}, p_{a''}^{i''}\}$}
  else if two of a, a', a'' are the same man (call him α) and $\exists b'$ tied with b on α's list and $b' \notin r_\alpha$ **then**
    – propose(p_α^1, b')
    {*this is the "forward" step:* b' *receives the proposal* p_α^1}
  else
    if all 3 proposals are "least desirable" **then**
      – reject p_α^1 where $\{p_\alpha^1, p_\alpha^2\} \subset \{p_a^i, p_{a'}^{i'}, p_{a''}^{i''}\}$
      {*two of these 3 equally least desirable proposals have to be from the same man* α *and* p_α^1 *is rejected*}
    else
      – reject any of the least desirable proposals
    end if
  end if
end if

Fig. 3 A description of proposals/disposals in our algorithm when ties have length 2

Once p_α^1 is transferred to b', it will not be possible for b to run the forward step on α again unless α's status changes. Since b will continue to have 2 proposals henceforth, it is not possible for b' to *bounce* p_α^1 to b. For b' to *forward* p_α^1 back to b, it is necessary for b' to have both p_α^1 and p_α^2. Since b cannot bounce p_α^2 to b', it means that b should have rejected p_α^2 at some point earlier, which means $b \in r_\alpha$ and so p_α^1 cannot be forwarded to b while a's status remains the same.

It is possible that b' rejects p_α^1 and this proposal is again made to b – however b cannot forward p_α^1 to b' now since $b' \in r_\alpha$. Thus for every $(\alpha, b) \in E$, there are at most 3 forward steps (corresponding to basic/1-promoted/2-promoted levels in α's status). □

Let the vector $x = (x_e, e \in E) \in \{0, 0.5, 1\}^{|E|}$ be defined as before, i.e., $x_{ab} = 1$ or 0.5 or 0 if the number of proposals that b accepts from a is 2 or 1 or 0, respectively. Let G' be the subgraph of G containing all those edges e such that $x_e > 0$. The maximum degree in G' is 2.

Let M' be a maximum cardinality matching in G' that matches all degree 2 vertices. It is easy to see that given G', the matching M' can be computed in linear time.

Thus it follows that our entire algorithm to compute M' in G runs in linear time. In the following, we show that M' is stable and it is a 10/7 approximation, thereby proving Theorem 2.

3.1 Analysis of the above algorithm

Propositions 3 and 4 follow from our algorithm. The stability of M' (shown in Lemma 11) follows from these propositions.

Proposition 3 *If woman b rejects proposal p_a^i at some point in the algorithm, 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 $a' = a''$) that rank at least as high as man a in b's list. In particular,*

1. *when p_a^i is made to b, if man a is ℓ-promoted (ℓ is either 1 or 2) and a' (similarly, a'') is tied with man a in b's list, then a' (resp. a'') is $\geq \ell$-promoted at that point.*
2. *if a' (similarly, a'') has a woman b' tied with b on the list of a' (resp. a''), then b' currently has two proposals.*
3. *if $a' = a''$ and a' has a woman b' tied with b on the list of a', then b' is currently in the rejection history $r_{a'}$.*

Proposition 4 *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, if a man has less than two proposals accepted, then he is 2-promoted. Also, he must have been rejected by all women on his list as a 2-promoted man.*
3. *If a woman b on the list of the man a is unmatched in M', then a is basic and he does not prefer b to the women accepting his proposals.*

Lemma 11 *The matching M' is stable in $G = (A \cup B, E)$.*

Proof Let $(a, b) \in M' \setminus E$. Suppose man a is unmatched in M'. By (1)-(2) of Proposition 4, a is 2-promoted and every woman on his list rejected at least one of his proposals. As this includes b, we know from Proposition 3 that b has two proposals from men ranking at least as high as a on her list and b is matched in M' with one of them. So (a, b) does not block M'.

Suppose a is matched in M' and let $(a, b') \in M'$. It follows from our algorithm that a must have been rejected by all women b ranking higher than b' on his list. By Proposition 3.1, b has two proposals from men ranking at least as high as a and she is matched in M' with one of them. So (a, b) does not block M'. $\square$

Bounding the size of M'

Let OPT be a maximum cardinality stable matching in G. Our method to bound $|M|$ in terms of $|\mathsf{OPT}|$ is similar to the method used in Section 2.1. We can however strengthen several of the lemmas to obtain an improved approximation ratio here.

We start with Lemma 12. Observe that this lemma immediately implies that there is no length-3 augmenting path in $M' \oplus \mathsf{OPT}$.

Lemma 12 *Suppose (a, b) and (a', b') are in OPT, where a is unmatched in M' and a' is not 2-promoted. If b' is unmatched in M' or a' prefers b to b', then (a', b) cannot be in G'.*

Proof We prove this lemma by contradiction. Suppose $(a', b) \in G'$. By (1)-(2) of Proposition 4, a has been rejected by b as a 2-promoted man. Then from the presence of (a', b) in G', Proposition 3, and the fact that a' is not 2-promoted, it follows that b prefers a' to a. If a' prefers b to b', then (a', b) blocks OPT, a contradiction.

So suppose b' is unmatched in M' and b' ranks at least as high as b on the list of a'. Using Proposition 4.3 and the presence of (a', b) in G', it follows that a' is indifferent between b and b'. Since a is unmatched in M', we know that b has rejected a proposal from a. Using Proposition 3.2, if b' and b are tied on the list of a', then b' has two proposals till the end of the algorithm. Then Proposition 4.1 implies that b' is matched in M', a contradiction. □

Let $R = \{\rho_1, \ldots, \rho_t\}$ be the set of all the length-5 augmenting paths in $M' \oplus \mathsf{OPT}$, where ρ_i is $a_0^i - b_0^i - a_1^i - b_1^i - a_2^i - b_2^i$, for $1 \leq i \leq t$. Lemma 13 lists properties of vertices in ρ_i, for any i.

Lemma 13 *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 ranks b_1^i at least as high as b_0^i.*
3. *a_2^i is basic and he ranks b_1^i at least as high as b_2^i.*
4. *In G', b_0^i has degree 1 if and only if a_1^i has degree 1.*
5. *In G', b_1^i has degree 1 if and only if a_2^i has degree 1.*

Proof Parts (1) and (3) follow from Proposition 4.

To show the first part of (2), suppose a_1^i is 2-promoted. To become 2-promoted, a_1^i must have been rejected by b_1^i as a 1-promoted man. We know b_2^i has received at most one proposal (by Proposition 4.1). It follows from Proposition 3.2 that a_2^i prefers b_1^i to b_2^i (otherwise a *bounce* step would have happened). It follows from Proposition 3 that either b_1^i prefers a_2^i to a_1^i or they are tied and a_2^i is also at least 1-promoted. In the former case, (a_2^i, b_1^i) blocks OPT; in the latter case, a_2^i is not basic, contradicting part (3). This establishes the first part of (2).

For the second part of (2), suppose a_1^i prefers b_0^i to b_1^i. Then b_0^i is indifferent between a_0^i and a_1^i, otherwise either (a_0^i, b_0^i) blocks M' or (a_1^i, b_0^i) blocks OPT. As a_0^i has been rejected by b_0^i as a 2-promoted man, Proposition 3.1 implies that a_1^i is also 2-promoted, a contradiction to the first part of (2). This completes the proof of (2).

For (4) and (5), it is enough to argue that each of a_1^i, a_2^i, b_0^i and b_1^i either has two proposals accepted or accept two proposals. Since b_0^i rejected a proposal from a_0^i (by part (1)), it follows from Proposition 3 that b_0^i receives two proposals. Next observe that b_0^i receives a proposal from a_1^i and the latter ranks b_1^i at least as high as b_0^i. If a_1^i prefers b_1^i to b_0^i, then b_1^i must have rejected a proposal from a_1^i. Then by Proposition 4, b_1^i must have two proposals. If b_1^i and b_0^i are tied on the list of a_1^i, the fact that b_0^i rejects a_0^i while ending up with a proposal from a_1^i implies that b_1^i has two proposals as well (Proposition 3.2). Finally, we know that neither a_1^i nor a_2^i is 2-promoted. It now follows from Proposition 4.2 that both a_1^i and a_2^i have two proposals each accepted. $\square$

As done in Section 2.1, we form the directed graph H from G' by first directing all edges from A to B and then contracting each (a_1^i, b_0^i) into a single node x_i and each (a_2^i, b_1^i) into a single node y_i. Parts (4) and (5) of Lemma 13 imply that $\deg_H(x_i), \deg_H(y_i)$ are in $\{0, 2\}$ for all i.

It is easy to see that Lemma 4 from Section 2.1 holds here as well, i.e., H has no (y_i, x_j) arc for any y-node y_i and x-node x_j, by substituting $a = a_0^j$, $b = b_0^j$, $a' = a_2^i$, and $b' = b_2^i$ in Lemma 12.

We use the notion of *critical arc* that we used in Section 2.1. Our definition now is a little stricter than Definition 3.

Definition 4 Call an arc (x_i, z) in H *critical* if a_1^i prefers z to b_1^i (in case z is an x-node/y-node, we mean a_1^i prefers the woman in z to b_1^i).

Lemma 5 in Section 2.1 holds here as well: if (x_i, z) is critical in H, then z cannot be an x-node.

Proof of Lemma 5 here. For any $1 \le i, j \le t$, 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 otherwise. It follows from the construction of H that $j \neq i$. We now contradict Lemma 12, by substituting $a = a_0^j$, $b = b_0^j$, $a' = a_1^i$, and $b' = b_1^i$. Note that we know a_1^i is not 2-promoted by Lemma 13.2. $\square$

Lemma 14 *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).
(2) If (y_j, y_i) is an arc, then there exists a critical arc (x_i, z).
(3) If (x_j, y_i) is a critical arc, then there exists a critical arc (x_i, z).

Proof Let y_i be an isolated vertex in H. This means that b_1^i receives two proposals from a_2^i. We first claim that b_1^i never rejects a_1^i. Suppose otherwise. We know from Proposition 3 and the fact that b_1^i did not bounce a_2^i's proposal to b_2^i that a_2^i prefers b_1^i to b_2^i. So b_1^i is indifferent between a_1^i and a_2^i, otherwise either (a_2^i, b_1^i) blocks OPT, or Proposition 3 is contradicted (since b_1^i rejected a_1^i).

At the point when b_1^i rejected a_1^i, it could not be the case that b_1^i's other two proposals were from a_2^i, since our algorithm ensures that a proposal from a_2^i gets rejected in such a case. As ties are of length 2, b_1^i at that point has a proposal from man a' who is ranked higher than a_1^i and a_2^i on her list. By Proposition 3.2, if a' has another woman b' tied on his list with b_1^i, then b' has two proposals since b_1^i did not bounce a''s proposal to b'. For the forward step to apply, it has to be the case both of a''s proposals are with b_1^i. In any case, a proposal of a' must have been rejected by b_1^i at some point, since she later accepts two proposals from a_2^i. This contradicts Proposition 3 as a' ranks higher than a_2^i on b_1^i's list.

Thus we have shown that b_1^i never rejects a_1^i. Since at least one proposal of a_1^i is accepted by b_0^i and b_1^i ranks at least as high as b_0^i on the list of a_1^i (Lemma 13.3), it must be the case that b_1^i and b_0^i are tied on a_1^i's list (otherwise a_1^i has to get rejected by b_1^i before proposing to b_0^i). Suppose both of a_1^i's proposals are accepted by b_0^i. We know that a_0^i has been rejected by b_0^i. By Proposition 3.3, b_1^i has to be in the rejection history of a_1^i at the point when a_0^i is rejected by b_0^i, contradicting that b_1^i never rejects a_1^i. So exactly one of the proposals of a_1^i is accepted by b_0^i. Since ties are of length 2, there has to be a woman w who ranks higher than b_1^i on a_1^i's list, who accepts a proposal of a_1^i. Thus there is a critical arc (x_j, z) in H. This completes the proof of (1).

To prove (2), we again claim that a_1^i is never rejected by b_1^i. Suppose otherwise. We know that b_2^j and b_2^i receive less than two proposals each at the end of the algorithm since they are left unmatched in M'. Hence it follows from Proposition 3.2 that a_2^j prefers b_1^i to b_2^j and a_2^i prefers b_1^i to b_2^i. Furthermore, a_2^i and a_2^j rank at least as high as a_1^i on the list of b_1^i (by Proposition 3). As ties are of length 2 on the women's side, either a_2^i or a_2^j ranks higher than a_1^i on the list of b_1^i. Then one of (a_2^i, b_1^i), (a_2^j, b_1^i) blocks OPT, a contradiction. Thus b_1^i never rejects a_1^i and the rest of the proof is identical to the proof of part (1) above.

For (3), note that $j \neq i$ since (x_j, y_i) is a critical arc. The proof of (3) is similar to (2), so we only sketch the outline: b_1^i never rejects a_1^i, otherwise one of (a_1^j, b_1^i), (a_2^i, b_1^i) blocks OPT. So b_1^i and b_0^i are tied on the list of a_1^i. It cannot happen that both proposals of a_1^i are accepted by b_0^i, since a_0^i is rejected by b_0^i. This implies (3). $\square$

We use the notion of *good path* from Section 2.1 (see Definition 2). It is easy to see that Lemma 6 still holds here: that is, every critical arc belongs to some good path and no two good paths intersect. Let $\mathcal{P}$ be the set of all good paths in H. Recall that there are only 3 possibilities in H for each $y_i \in \{y_1, \ldots, y_t\}$: (1) y_i is isolated, (2) y_i is in a cycle of y-nodes, (3) y_i is in a good path.

We now define $f : [t] \to \mathcal{P}$ as follows. For any $i \in [t]$:

- If y_i is either isolated or in a cycle of y-nodes, then set $f(i) = p$, where $p \in \mathcal{P}$ contains the critical arc (x_i, z). We know in both these cases (isolated/cycle of y-nodes), there is a critical arc (x_i, z) in H by parts (1) and (2) of Lemma 14, respectively.
- Otherwise y_i is in a good path p'. If H contains a critical arc (x_i, z), then set $f(i) = p$, where $p \in \mathcal{P}$ contains (x_i, z). Otherwise we set $f(i) = p'$.

  [note that if there is no critical arc (x_i, z) in H, then parts (2)-(3) of Lemma 14 imply that either no x-node/y-node precedes y_i in p' or the path p' has an arc (x_j, y_i) that is *not* critical – in either case p' has no critical arc]

Lemma 15 *The function f is one-to-one.*

Proof Consider any $p \in \mathcal{P}$. By Lemma 5, p can contain at most one critical arc (x_i, z) where z is either a y-node or a blue woman. If p contains a critical arc (x_i, z), then $f(i) = p$ and no other index is mapped to p.

Suppose p contains no critical arc. Additionally, if p has no y-nodes, then p has no pre-image, else let y_j be the first y-node in p. The only index that can get mapped to p is j; recall that we have $f(j) = p$ if there is no critical (x_j, z) in H. □

Let $Q = (M' \oplus \mathsf{OPT}) \setminus R$. The above lemma allows us to show a charging scheme where for any $q \in Q$, if q is 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, then q pays for $c_q \leq \ell_q$ paths in R.

We will show later such a charging scheme exists. For now, we show why this implies that $|\mathsf{OPT}|/|M'| \leq 10/7$. Note that all the paths in R have been paid for in our charging scheme. So we have:

$$\frac{|\mathsf{OPT}|}{|M'|} = \frac{\sum_{q \in Q}(|\mathsf{OPT} \cap q| + 3c_q)}{\sum_{q \in Q}(|M' \cap q| + 2c_q)} \leq \max_{q \in Q} \frac{|\mathsf{OPT} \cap q| + 3c_q}{|M' \cap q| + 2c_q} \leq \max_{\ell_q \geq 2} \frac{4\ell_q + 2}{3\ell_q + 1} \leq \frac{10}{7}.$$

As done in Section 2.1, we get the maximum value of the above ratio for any $q \in Q$ by setting c_q to its largest value of ℓ_q and letting q be an augmenting path of length $2\ell_q + 3 \geq 7$ so that $|\mathsf{OPT} \cap q| = \ell_q + 2$ and $|M' \cap q| = \ell_q + 1$. Then the ratio becomes $(\ell_q + 2 + 3\ell_q)/(\ell_q + 1 + 2\ell_q)$, which is maximized by setting $\ell_q = 2$. Thus we get 10/7 as an upper bound for $|\mathsf{OPT}|/|M'|$ and Theorem 2 is proved.

Our charging scheme

For any $p \in \mathcal{P}$, let $\mathsf{cost}(p) = 1$ if there is some $i \in [t]$ such that $f(i) = p$, else let $\mathsf{cost}(p) = 0$. We define a function $\mathsf{charge} : A \cup B \to \{0, 1/2\}$ now. Set $\mathsf{charge}(u) = 0$ for all $u \in A \cup B$ and reset charge values for some vertices as follows. For each $p \in \mathcal{P}$ with $\mathsf{cost}(p) = 1$, do the following:

Let a_p and b_p be the endpoints of p. Since M' is a maximum cardinality matching in G', at least one of a_p, b_p has to be matched in M'. If both are matched in M', then assign $\mathsf{charge}(a_p) = \mathsf{charge}(b_p) = 1/2$. Otherwise form the alternating path L_p in G' with the unmatched vertex in $\{a_p, b_p\}$ as one endpoint. Let r_p be the other endpoint of L_p; we know that r_p has to be matched in M'. Assign $\mathsf{charge}(r_p) = \mathsf{charge}(s_p) = 1/2$ where s_p is the matched vertex in $\{a_p, b_p\}$.

Since paths in $\mathcal{P}$ are vertex-disjoint, every $p \in \mathcal{P}$ has its own distinct endpoints. Also $r_p \neq r_{p'}$ for any distinct $p, p' \in \mathcal{P}$ (the same argument as in Lemma 9). Hence any vertex gets charged for at most one path in $\mathcal{P}$ and we have $\sum_{u \in A \cup B} \mathsf{charge}(u) = t$. It is also clear that only vertices matched in M' get assigned positive charge.

Recall that R is the set of length-5 augmenting paths in $M' \oplus \mathsf{OPT}$. Our charging scheme implies that for any $q \in Q$, where $Q = (M' \oplus \mathsf{OPT}) \setminus R$, if q is 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'), then q pays for $c_q \leq 2\ell_q/2 = \ell_q$ paths in R. We will now show that if $q = \alpha_0 - \beta_0 - \alpha_1 - \cdots - \beta_\ell - \alpha_{\ell+1} - \beta_{\ell+1}$ is an augmenting path on $2\ell + 3 \geq 7$ edges, then $\mathsf{charge}(\beta_0) = 0$ and $\mathsf{charge}(\alpha_{\ell+1}) = 0$. Thus our charging scheme ensures that such a q also pays for $c_q \leq 2\ell_q/2 = \ell_q$ paths in R.

We first show that neither β_0 nor $\alpha_{\ell+1}$ can be the vertex r_p for any $p \in \mathcal{P}$. As done in the proof of Lemma 9, the following claim holds: if β_0 has degree 1 in G', then $M'(\beta_0) = \alpha_1$ also has degree 1; similarly, if $\alpha_{\ell+1}$ has degree 1 in G', then $M'(\alpha_{\ell+1}) = \beta_\ell$ also has degree 1. This claim immediately implies that neither β_0 nor $\alpha_{\ell+1}$ can be the degree 1 vertex which is the matched endpoint of an alternating path starting from an unmatched vertex.

Suppose $\beta_0 = b_p$ for some $p \in \mathcal{P}$. We will show that $\mathsf{cost}(p) = 0$ then. For any y-node y_j, H cannot have the arc (y_j, β_0). To admit such an arc in H, we need the edge (a_2^j, β_0) in G'. This means that a_2^j ranks β_0 at least as high as b_2^j (by Proposition 4.3). This contradicts Lemma 12 by substituting $a = \alpha_0$, $b = \beta_0$, $a' = a_2^j$ and $b' = b_2^j$. So if $\beta_0 = b_p$ for some $p \in \mathcal{P}$, then p's structure has to be $a_p - x_i - \cdots - x_j - \beta_0$, i.e., it has no y-nodes. For such a p to have positive cost, there needs to be critical arc in p. So the arc (x_j, β_0) has to be critical, in other words, a_1^j prefers β_0 to b_1^j. We again contradict Lemma 12 by substituting $a = \alpha_0$, $b = \beta_0$, $a' = a_1^j$ and $b' = b_1^j$. Hence $\mathsf{charge}(\beta_0) = \mathsf{cost}(p) = 0$.

Suppose $\alpha_{\ell+1} = a_p$ for some $p \in \mathcal{P}$. For any x-node x_j, H cannot have the arc $(\alpha_{\ell+1}, x_j)$ as that means $(\alpha_{\ell+1}, b_0^j)$ is in G'. This contradicts Lemma 12 by substituting $a = a_0^j$, $b = b_0^j$, $a' = \alpha_{\ell+1}$ and $b' = \beta_{\ell+1}$. So if $\alpha_{\ell+1} = a_p$ for some $p \in \mathcal{P}$, then p has no x-nodes; so p's structure is $\alpha_{\ell+1} - y_j - \cdots - b_p$. For such a path p to have positive cost, it has to be the case that there is *no* critical arc (x_j, z) in H. We will show such an arc, in fact, exists, hence $\mathsf{cost}(p) = 0$.

The edge $(\alpha_{\ell+1}, b_1^j)$ has to be in G' since $(\alpha_{\ell+1}, y_j)$ is in H. If b_1^j ever rejects a_1^j, then both a_2^j and $\alpha_{\ell+1}$ rank at least as high as a_1^j on b_1^j's list (by Proposition 3). As ties are of length 2, b_1^j prefers at least one of them to a_1^j. Note that both $\beta_{\ell+1}$ and b_2^i have less than two proposals as they are unmatched in M'. Then Proposition 3 and Proposition 4.3 imply that $\alpha_{\ell+1}$ prefers b_1^j to $\beta_{\ell+1}$ and a_2^j prefers b_1^j to b_2^j. So one of $(\alpha_{\ell+1}, b_1^j)$, (a_2^j, b_1^j) blocks OPT, a contradiction.

So b_1^j never rejects a_1^j. The rest of the proof is similar to the argument seen in the proof of Lemma 14.1: so b_1^j and b_0^j are tied on the list of a_1^j, it cannot happen that both proposals of a_1^j are accepted by b_0^j, since a_0^j is rejected by b_0^j. Hence there is a woman w ranked better than b_1^j who accepts a proposal of a_1^j. Thus there is a critical arc (x_j, z) in H, so $\mathsf{charge}(\alpha_{\ell+1}) = \mathsf{cost}(p) = 0$. This completes the proof of correctness of our charging scheme.

References

1. G. Askalidis, N. Immorlica, A. Kwanashie, D. Manlove, and E. Pountourakis. Socially Stable matchings in the Hospitals / Residents problem. In *13th WADS*, 2013.
2. D. Gale and L. S. Shapley. College admissions and the stability of marriage. American Math. Monthly, Vol. 69: 9–15, 1962.
3. D. Gale and M. Sotomayer. Some remarks on the stable marriage problem. Discrete Applied Mathematics, Vol. 11: 223–232, 1985.
4. D. Gusfield and R. W. Irving. The Stable Marriage Problem: Structure and Algorithms. MIT Press, Boston, MA 1989.

5. M. M. Halldórsson, K. Iwama, S. Miyazaki, and H. Yanagisawa. Randomized approximation of the stable marriage problem. Theoretical Computer Science, Vol. 325(3): 439–465, 2004.
6. R. W. Irving. Stable marriage and indifference. Discrete Applied Mathematics, Vol. 48: 261–272, 1994.
7. R. W. Irving and D. F. Manlove. Approximation algorithms for hard variants of the stable marriage and hospitals/residents problems. Journal of Combinatorial Optimization, Vol. 16(3): 279–292, 2008.
8. K. Iwama, D. F. Manlove, S. Miyazaki, and Y. Morita. Stable marriage with incomplete lists and ties. In *36th ICALP*: 443–452, 1999.
9. K. Iwama, S. Miyazaki, and N. Yamauchi. A 1.875-approximation algorithm for the stable marriage problem. In *18th SODA*: 288–297, 2007.
10. K. Iwama, S. Miyazaki, and H. Yanagisawa. A 25/17-Approximation Algorithm for the Stable Marriage Problem with One-Sided Ties. In *18th ESA*: 135–146, 2010.
11. Z. Király. Better and simpler approximation algorithms for the stable marriage problem. Algorithmica: Vol. 60(1): 3–20, 2011.
12. Z. Király. Linear time local approximation algorithm for maximum stable marriage. Algorithms, Vol. 6(3): 471-484, 2013.
13. D. Knuth. Mariages stables et leurs relations avec d'autre problèmes. Les Presses de l'université de Montréal, 1976.
14. D. Manlove. Algorithmics of Matching Under Preferences World Scientific Publishing Company Incorporated, 2013.
15. D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. Theoretical Computer Science, Vol. 276(1-2): 261–179, 2002.
16. E. J. McDermid. A 3/2 approximation algorithm for general stable marriage. In *36th ICALP*: 689–700, 2009.
17. K. Paluch. Faster and simpler approximation of stable matchings. In *9th WAOA*: 176-187, 2011.
18. A. Roth and M. Sotomayor. Two-sided matching: a study in game-theoretic modeling and analysis. Cambridge university press, 1992.
19. C.-P. Teo, J. Sethuraman, and W. P. Tan. Gale-Shapley stable marriage problem revisited: strategic issues and applications. In *7th IPCO*: 429–438, 1999.
20. J. Vande Vate. Linear Programming Brings Marital Bliss. In *Operation Research Letters* 8: 147-153, 1989.
21. H. Yanagisawa. Approximation Algorithms for Stable Marriage Problems. Ph.D. Thesis, Kyoto University, 2007.