

Critical Relaxed Stable Matchings with Two-Sided Ties

Meghana Nasre¹, Prajakta Nimbhorkar², and Keshav Ranjan¹

¹ IIT Madras, Chennai, India

² Chennai Mathematical Institute and UMI ReLaX, India

Abstract. We consider the stable marriage problem in the presence of ties in preferences and critical vertices. The input to our problem is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where \mathcal{A} and \mathcal{B} denote sets of vertices which need to be matched. Each vertex has a preference ordering over its neighbours possibly containing ties. In addition, a subset of vertices in $\mathcal{A} \cup \mathcal{B}$ are marked as critical and the goal is to output a matching that matches as many critical vertices as possible. Such matchings are called critical matchings in the literature and in our setting, we seek to compute a matching that is critical as well as optimal with respect to the preferences of the vertices.

Stability, which is a well-accepted notion of optimality in the presence of two-sided preferences, is generalized to weak-stability in the presence of ties. It is well known that in the presence of critical vertices, a matching that is critical as well as weakly stable may not exist. Popularity is another well-investigated notion of optimality for the two-sided preference list setting, however, in the presence of ties (even with no critical vertices), a popular matching need not exist. We, therefore, consider the notion of relaxed stability which was introduced and studied by Krishnaa et. al. (SAGT 2020). We show that a critical matching which is relaxed stable always exists in our setting although computing a maximum-sized relaxed stable matching turns out to be NP-hard. Our main contribution is a $3/2$ approximation to the maximum-sized critical relaxed stable matching for the stable marriage problem with two-sided ties and critical vertices.

Keywords: Stable Matching · Two-Sided Ties · Critical · Relaxed Stable · Approximation Algorithm.

1 Introduction

We study the stable marriage problem in the presence of *ties* in preferences and *critical vertices*. Formally, the input to our problem is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$, where \mathcal{A} and \mathcal{B} are two sets of vertices and E denotes the set of all the acceptable vertex-pairs. Each vertex $u \in \mathcal{A} \cup \mathcal{B}$ ranks a subset of vertices in the other partition (its neighbours in G) in order of its preference possibly involving ties – this ordering is denoted as $\text{Pref}(u)$. For a vertex u let v_1 and v_2 be its neighbours in G . The vertex u strictly prefers v_1 over v_2 (denoted as $v_1 \succ_u v_2$)

if the rank of the edge (u, v_1) is smaller than the rank of the edge (u, v_2) . The vertex u is tied between v_1 and v_2 (denoted as $v_1 =_u v_2$) if the ranks on the edges (u, v_1) and (u, v_2) are the same. We use $v_1 \succeq_u v_2$ to denote that the rank of v_1 is at least as good as the rank of v_2 in $\text{Pref}(u)$. In addition, the input consists of a set $\mathcal{C} \subseteq (\mathcal{A} \cup \mathcal{B})$ of *critical vertices*. Critical vertices are a generalization of lower-quota vertices that *must* be matched in any assignment. In our setting, critical vertices can be left unassigned, however, we wish to minimize the number of critical vertices left unassigned.

A *matching* $M \subseteq E$ in G is a set of edges that do not share an end-point. For each vertex $u \in \mathcal{A} \cup \mathcal{B}$, we denote by $M(u)$, the neighbour of u that is assigned to u in M . In the presence of critical vertices, the most important attribute of any matching is to match as many critical vertices as possible. A matching M is *critical* [11] if there is no matching that matches more critical vertices than M . In this work, we are interested in computing a critical matching that is *optimal* with respect to the preferences of the vertices in an instance of our setting.

Lower-quotas or critical vertices/positions naturally arise in applications like Hospital-Residents problem [7] where rural hospitals must be prioritized to ensure sufficient staffing. Another example is the problem of assigning sailors to billets [26] in the US Navy where some critical billets cannot be left vacant [23,27]. Ties in preferences is yet another important practical consideration in matching problems and has been extensively investigated in the literature [17,18,13,22,2,9,8]. However, there is a limited investigation of matching problems with ties as well critical vertices [5] and ours is the first work that allows ties as well as critical vertices on both sides of the bipartition.

Stability which is the de-facto notion of optimality for two-sided preferences is defined by the absence of a blocking pair. Informally, an assignment is stable if no unassigned pair wishes to deviate from it.

Definition 1 (Stable Matchings). *Given a matching M , a pair $(a, b) \in E \setminus M$ is called blocking pair w.r.t. M if (i) either a is unmatched or $b \succ_a M(a)$ and (ii) either b is unmatched or $a \succ_b M(b)$. The matching M is stable if there is no blocking pair w.r.t. M .*

When all preferences are strict, that is there are no ties, every instance of the stable marriage problem admits a stable matching and it can be computed using the famous Gale and Shapley algorithm [3]. In addition, it is well known [24,25] that all stable matchings have the same size.

Stable matching in the presence of ties: When preferences are allowed to have ties, the notion of stability defined above is referred to as “weak stability” (referred to as stability in the rest of the paper). We remark that for a pair (a, b) to block a matching M , both a and b prefer each other *strictly* over their current partners in M . Every instance of the stable marriage problem with ties admits a stable matching and it can be efficiently computed. However, unlike in the case of strict lists, all stable matchings need not have the same size and the problem of computing a maximum or minimum size stable matching is NP-hard [17] under severe restrictions – the ties occur at the end of preference lists and on one side of bipartition only, there is at most one tie per list, and each tie is of length two.

Stable/popular matching in the presence of critical vertices: When preferences of all the vertices are strict and we have critical vertices as a part of the input, a stable matching always exists. However, stable matching which is also critical may not exist (see Figure 1). Since stability and criticality are not simultaneously guaranteed, an alternate notion of optimality, namely *popularity* [4] is extensively investigated in the literature [19,20,11]. The goal is to compute *popular* amongst the set of critical matchings. Informally, a matching M is *popular* in a set of matchings if no majority of vertices wish to deviate from M to any other matching in that set. It is known [11,20] that an instance with strict preference lists *always* admits a matching which is popular amongst critical matchings and such a matching can be computed efficiently. Hence, it is natural to consider popularity in the presence of critical vertices and ties.

However, even when ties are present in the preferences only on one side of the bipartition (even with no critical vertices), popular matchings are not guaranteed to exist [1], and deciding whether a popular matching exists is NP-Hard. In light of this, we explore the notion of *relaxed stability*.

Relaxed stability in the presence of ties and critical vertices: The notion of relaxed stability was introduced and studied by Krishnaa *et al.* [14] for the Hospital-Residents problem with lower quotas (HRLQ). In their setting, preferences are assumed to be strict. The HRLQ setting is a many-to-one matching problem where a hospital h can accept at most $q^+(h)$ many residents and has $q^-(h) \leq q^+(h)$ many critical positions. To satisfy the critical positions at a hospital, certain residents may be *forced* to be matched to the hospital. The notion of relaxed stability allows only such residents to participate in blocking pairs. In addition, if a resident matched to h participates in a blocking pair then the hospital h should not be *surplus*, that is $|M(h)| \leq q^-(h)$.

In the HRLQ setting, preferences are strict, hospitals have capacities as well as critical positions are allowed only for hospitals. In contrast, we allow ties in preferences as well as critical vertices to appear on *both sides* of the bipartition. However, our setting is one-to-one.

We now define the notion of relaxed stability RSM for our setting. Intuitively, a matching M is a RSM if every blocking pair (a, b) w.r.t. M is *justified* by either the a endpoint or by the b endpoint. A vertex a justifies the blocking pair if $M(a)$ is a critical vertex. That is, $M(a)$ forces a to be matched to a lower-preferred vertex than b . Similarly, the vertex b can justify the blocking pair (a, b) .

Definition 2 (Relaxed stability in our setting). A matching M is RSM if for every blocking pair (a, b) w.r.t. M one of the following holds:

1. a is matched and $b' = M(a)$ is critical, or
2. b is matched and $a' = M(b)$ is critical.

A matching M is called critical relaxed stable matching (CRITICAL-RSM) if it is *critical* as well as *relaxed stable*. In the instance shown in Figure 1, the matching M_1 is critical but not RSM whereas M_2 is CRITICAL-RSM.

Our first contribution is to show that a CRITICAL-RSM always exists in our setting. We remark that when $\mathcal{C} = \emptyset$, an instance of our setting is the same

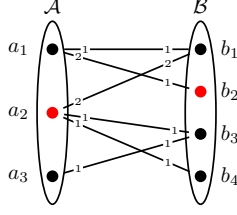


Fig. 1: Red vertices are critical, black vertices are non-critical. The numbers on the edges denote the ranks of the respective end-points. The instance does not admit any critical stable matching because b_2 remains unmatched in every stable matching. $M_1 = \{(a_1, b_2), (a_2, b_1), (a_3, b_3)\}$ is critical but not RSM because the blocking edge (a_2, b_4) is not justified. $M_2 = \{(a_1, b_2), (a_2, b_4), (a_3, b_3)\}$ is CRITICAL-RSM because the only blocking edge (a_1, b_1) is justified.

as stable marriage setting with ties but without critical vertices, and hence the set of CRITICAL-RSM is the same as the set of stable matchings. This immediately implies that computing a maximum size critical RSM is NP-Hard [17] and hard to approximate [6]. For the problem of computing a maximum sized stable matching in the presence of two-sided ties, the current best approximation factor [13,18,22] is $\frac{3}{2}$. The main result (Theorem 1) provides the same approximation size guarantee for a maximum sized CRITICAL-RSM in our setting.

Theorem 1. *Let $G = (\mathcal{A} \cup \mathcal{B}, E)$ be an instance of the stable marriage problem with two-sided ties and two-sided critical vertices. Then G always admits a matching M such that M is CRITICAL-RSM and can be computed efficiently. Moreover, $|M| \geq \frac{2}{3}|M'|$, where M' is a maximum size CRITICAL-RSM in G .*

Related work: As mentioned earlier, the generalizations of the stable marriage problem to allow one of ties in preferences or critical vertices/lower-quota positions has been extensively investigated. The only work which we are aware of allows both ties and critical vertices is a recent work by Goko *et al.* [5]. They study the Hospital-Residents problem with lower-quotas with ties on both sides. However, one side of the bipartition cannot have critical vertices. Furthermore, their results are for complete preference, a restricted setting. Goko *et al.* define the maximum satisfaction ratio which for our one-to-one setting coincides with the definition of critical matchings. However, their goal is to compute amongst all stable matchings the one that achieves criticality.

For strict preferences and lower-quotas / critical vertices, various notions like envyfreeness [28,15], popularity [19,11,20,21], and relaxed stability [14,15] have been studied. Relaxed stability and popularity do not define the set of matchings even in the one-to-one strict-list setting and critical vertices restricted to one side only (see Appendix A). Hamada *et al.* [7] consider the problem of computing a matching with minimum number of blocking pairs or blocking residents.

For the stable marriage problem with ties (without critical vertices) there is a long line of investigation [12,18,13,22,10,2,9] in order to improve the approximation ratio under various restricted settings. The best known approximation algorithm for the case of one-sided ties is by Lam and Plaxton [16] whereas the best known for the case of two-sided ties is by [13,18,22]. We use Király's algorithm [13] in our work.

2 Preliminaries

Our algorithm described in the next section combines the ideas in (i) Király's algorithm [13] for computing a stable matching in the presence of two-sided ties and (ii) Multi-level algorithm for computing popular critical matching [21] for strict preferences. We give an overview of the algorithms and also define terminology useful for our algorithm.

Overview of Király's algorithm [13]. Király's algorithm [13] is a proposal-based algorithm where vertices in \mathcal{A} propose and vertices in \mathcal{B} accept or reject. We need the term *uncertain proposal* from [13] which is defined below.

Definition 3 (Uncertain Proposal). *Let b be some k^{th} rank neighbour of a in $\text{Pref}(a)$. During the course of the algorithm, the proposal from a to b is uncertain if there exists another k^{th} rank neighbour b' of a which is unproposed by a and unmatched in the matching. Once a proposal (a, b) is uncertain, it remains uncertain until b rejects a .*

Each time an $a \in \mathcal{A}$ proposes to its *favourite* neighbour b (we define it formally in Definition 4), the vertex b accepts/rejects as follows:

1. If b is unmatched then b immediately accepts the proposal.
2. If b is matched, say to a' , and (a', b) is an uncertain proposal, then b rejects a' and accepts the proposal from a , irrespective of the ranks of a and a' in $\text{Pref}(b)$. In this case, b is *marked* by a' .
3. If b is matched, say to a' , and (a', b) is not an uncertain proposal, then
 - (i) if $a \succ_b a'$ then b rejects a' and accepts the proposal from a , or
 - (ii) if $a' \succeq_b a$ then b rejects a .

The reason for a' marking the vertex b in (2) is as follows: In this case, b rejects the uncertain proposal from a' and accepts a *irrespective* of the preference of b' between a and a' . Later, when a' gets its chance to propose, and if none of the neighbours of a' at the rank of b accept the proposal from a' then a' will propose to the marked vertex b before proposing to the next lower-ranked neighbours. In contrast in (3)(i) above, when the proposal (a', b) is not uncertain and $a \succ_b a'$ then a' does not mark b . Note that a vertex $b \in \mathcal{B}$ can be part of an uncertain proposal at most once. Once a vertex receives its first proposal it will remain matched and thereafter cannot be part of any uncertain proposal. Thus, any $b \in \mathcal{B}$ can be marked at most once during the course of the algorithm.

Now, we define the favourite neighbour of a vertex a , which is an adaptation of the definition in [13].

Definition 4 (Favourite neighbour of a). Assume that k is the best rank at which some unproposed or marked neighbours of a exist in $\text{Pref}(a)$. Then b is the favourite neighbour of a if one of the following conditions holds:

- (i) there exists at least one unmatched neighbour of a at the k^{th} rank and b has the lowest index among all such unmatched neighbours, or
- (ii) all the k^{th} -ranked neighbours of a are matched and b is the lowest index among all such neighbours which are unproposed by a , or
- (iii) all the k^{th} -ranked neighbours are already proposed by a and b has the lowest index among all the vertices which are marked by a .

Király's algorithm begins with every vertex $a \in \mathcal{A}$ being active. As long as there exists an active vertex which is unmatched and has not exhausted its preference list, the vertex proposes to its favourite neighbour. If $a \in \mathcal{A}$ remains unmatched after exhausting its preference list, it achieves a '*' status and starts proposing to vertices in $\text{Pref}(a)$ with * status. The * status of a vertex a can be interpreted as improving the rank of a in $\text{Pref}(b)$ by 0.5 for any neighbour b of a . Thus, the * status vertex is used to decide between vertices in a tie, but does not affect strict preferences. It is shown in [13] that the resulting matching is a $\frac{3}{2}$ -approximation of a maximum size stable matching.

Overview of the popular critical matching algorithm [21]. Now, we briefly describe the algorithm in [21] for computing the maximum size popular critical matching in the one-to-one strict list setting. Let s and t denote the number of critical vertices in \mathcal{A} and \mathcal{B} , respectively. The algorithm in [21] is a multi-level algorithm which first matches as many critical vertices from \mathcal{B} as possible by allowing each unmatched $a \in \mathcal{A}$ to propose only critical vertices on the \mathcal{B} -side at levels $0, \dots, t-1$. At the level t each vertex $a \in \mathcal{A}$ is allowed to propose *all* its neighbours. If a vertex $a \in \mathcal{A}$ remains unmatched even after exhausting its preference list at level t , a raises its level to $t+1$ and proposes to its neighbours until it is matched or it exhausts its preference list at the level $t+1$. If a critical vertex a remains unmatched then a raises its level above $t+1$ and continues proposing to all its neighbours until it is matched or it exhausts its preference list at the highest level $s+t+1$. A vertex b which receives the proposal always prefers a higher level vertex a over any lower level vertex a' irrespective of the ranks of a and a' in $\text{Pref}(b)$. It is shown in [21] that the resulting matching is a maximum size popular matching among all the critical matchings.

3 Algorithm for computing CRITICAL-RSM

Our algorithm (see Algorithm 1) is a combination of Király's algorithm and the popular critical matching algorithm discussed in the previous section. In each level, vertices in \mathcal{A} propose and vertices in \mathcal{B} accept or reject. The set of vertices that $a \in \mathcal{A}$ proposes to depends on its level. Furthermore, depending on the level of a , the preference list at that level may be strict or may contain ties. Throughout Algorithm 1, b uses its original preference list $\text{Pref}(b)$ which possibly contains ties. For a vertex $a \in \mathcal{A}$, let $\text{PrefS}(a)$ denote a *strict* preference

list obtained by breaking ties in $\text{Pref}(a)$ so that the vertices in ties are ordered by increasing order of their indices. Furthermore, let $\text{PrefSC}(a)$ be the strict list obtained from $\text{PrefS}(a)$ by omitting all the non-critical vertices from $\text{PrefS}(a)$. For example, assume $\text{Pref}(a) = (b_2, b_1), b_5, (b_3, b_4)$ where b_4 and b_5 are critical vertices. Here, a ranks b_1 and b_2 as rank-1, b_5 as rank-2 and b_3 and b_4 as rank-3. We have $\text{PrefS}(a) = b_1, b_2, b_5, b_3, b_4$ and $\text{PrefSC}(a) = b_5, b_4$ where comma separated vertices denote a strict ordering.

Initially, all the vertices in \mathcal{A} have their levels set to 0. A vertex a at level ℓ is denoted as a^ℓ . At a level less than t (see Line 4–Line 8 of Algorithm 1), each $a \in \mathcal{A}$ proposes to vertices in $\text{PrefSC}(a)$. Each time it remains unmatched, it proposes to its *most preferred* neighbour b . The most preferred neighbour in $\text{PrefSC}(a)$ or $\text{PrefS}(a)$ is the best-ranked neighbour b to whom a has not yet proposed at the current level. If a remains unmatched after proposing to all its neighbours in $\text{PrefSC}(a)$ at a level $\ell < t - 1$, then a raises its level to $\ell + 1$ and again proposes to vertices in $\text{PrefSC}(a)$. In this part of the algorithm, we invoke **CriticalPropose()** which encodes the level-based accept/reject by b . A vertex $b \in \mathcal{B}$ prefers a_i^ℓ over $a_j^{\ell'}$ if :

- (i) either $\ell > \ell'$ (ranks of a_i and a_j in $\text{Pref}(b)$ do not matter) or
- (ii) $\ell = \ell'$ and $a_i \succ_b a_j$.

If vertex a remains unmatched after exhausting $\text{PrefSC}(a)$ at level $t - 1$, a attains level t where it uses its original preference list $\text{Pref}(a)$ which may contain ties. At level t our algorithm executes Király's algorithm [13]. This corresponds to Line 10–Line 13 of Algorithm 1. Király's algorithm is encoded in the procedure **TiesPropose()**. Since we have two-sided ties at this level, we need the notion of a favourite neighbour and uncertain proposal defined in Section 2. If the vertex a remains unmatched after exhausting $\text{Pref}(a)$ at level t , it attains the $*$ status, and for this, we have the sub-level t^* . The interpretation of the $*$ status is the same as discussed in Section 2.

If a *critical* vertex a remains unmatched after exhausting its preference list $\text{Pref}(a)$ at level t^* , a raises its level to $t + 1$, and starts proposing to vertices in $\text{PrefS}(a)$ (see Line 16–Line 20 of Algorithm 1). It continues doing so until either it is matched or it has exhausted $\text{PrefS}(a)$ at level $s + t$. In contrast, if a non-critical vertex a remains unmatched after exhausting its preference list $\text{Pref}(a)$ at level t^* , a does not propose any further. Recall that $\text{PrefS}(a)$ is a strict preference list containing all the neighbours (not restricted to critical vertices). Here, Algorithm 1, again invokes **CriticalPropose()** for the level-based accept/reject by b . The algorithm terminates when either (i) all the vertices in \mathcal{A} are matched or (ii) all unmatched critical $a \in \mathcal{A}$ have exhausted $\text{PrefS}(a)$ at level $s + t$ and all unmatched non-critical $a \in \mathcal{A}$ have exhausted $\text{Pref}(a)$ at level t^* . We note that $s + t = |\mathcal{C}| = O(n)$, where $n = |\mathcal{A} \cup \mathcal{B}|$ and each edge of G is explored at most $s + t + 3$ times (at most three times at level t , the Király's step, and at most once at every other level). Thus, the running time of our algorithm is $O(n \cdot |E|)$.

Algorithm 1: Critical relaxed stable matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$

```

1 Set  $M = \emptyset$ , Initialize a queue  $Q = \{a^0 : a \in \mathcal{A}\}$ 
2 while  $Q$  is not empty do
3   Let  $a^\ell = \text{dequeue}(Q)$  /*  $a$  is unmatched */
4   if  $\ell < t$  then
5     if  $a^\ell$  has not exhausted  $\text{PrefSC}(a)$  then
6       |  $\text{CriticalPropose}(a^\ell, \text{PrefSC}(a), M, Q)$ 
7     else
8       |  $\ell = \ell + 1$  and add  $a^\ell$  to  $Q$ 
9   else if  $\ell == t$  or  $\ell == t^*$  then
10    if  $\exists b' \in \text{Pref}(a)$  which is marked/unproposed by  $a^\ell$  then
11      |  $\text{TiesPropose}(a^\ell, \text{Pref}(a), M, Q)$ 
12    else
13      | if  $\ell == t$  then  $\ell = t^*$  and add  $a^\ell$  to  $Q$ 
14      | if  $\ell == t^*$  and  $a$  is critical then  $\ell = t + 1$  and add  $a^\ell$  to  $Q$ 
15    else
16      if  $a^\ell$  has not exhausted  $\text{PrefS}(a)$  then
17        |  $\text{CriticalPropose}(a^\ell, \text{PrefS}(a), M, Q)$ 
18      else
19        | if  $\ell < s + t$  and  $a$  is critical then
20        |  $\ell = \ell + 1$  and add  $a^\ell$  to  $Q$ 
21 return  $M$ 

```

4 Correctness of our algorithm

We prove that the matching M output by Algorithm 1 is

- (I) Critical as well as relaxed stable (RSM) and
- (II) A $\frac{3}{2}$ approximation to the maximum size CRITICAL-RSM in G .

The proofs of claims marked with \star are deferred to Appendix.

The partition of vertices defined below based on the levels of vertices in \mathcal{A} and the matching M is useful for us.

Procedure $\text{CriticalPropose}(a^\ell, \text{List}(a), M, Q)$

```

1 Let  $b$  be the most preferred unproposed vertex by  $a^\ell$  in  $\text{List}(a)$ 
2 if  $b$  is unmatched in  $M$  then
3   |  $M = M \cup \{(a^\ell, b)\}$ 
4 else
5   | Let  $a_j^y = M(b)$ 
6   | if  $(\ell > y)$  or  $(\ell == y$  and  $a \succ_b a_j)$  then
7   | |  $M = M \setminus \{(a_j^y, b)\} \cup \{(a^\ell, b)\}$  and add  $a_j^y$  to  $Q$ 
8   | else add  $a^\ell$  to  $Q$ 

```

Partition of vertices: The vertex set \mathcal{A} is partitioned into $\mathcal{A}_0 \cup \mathcal{A}_1 \cup \dots \cup \mathcal{A}_t \cup \dots \cup \mathcal{A}_{s+t}$, and the vertex set \mathcal{B} is partitioned into $\mathcal{B}_0 \cup \mathcal{B}_1 \cup \dots \cup \mathcal{B}_t \cup \dots \cup \mathcal{B}_{s+t}$. For every matched vertex $a \in \mathcal{A}$ there exists $x \in \{0, \dots, s+t\}$ such that $(a^x, b) \in M$. We use x to partition the vertex set. Note that if $(a^{t^*}, b) \in M$ then for the purpose of partitioning we consider $t^* = t$ as t^* is a sub-level of the level t .

- **Matched vertices in $\mathcal{A} \cup \mathcal{B}$:** Let $a \in \mathcal{A}, b \in \mathcal{B}$ and $(a^x, b) \in M$ for some $x \in \{0, \dots, s+t\}$. Then we add a to \mathcal{A}_x and b to \mathcal{B}_x .
- **Unmatched vertices in $\mathcal{A} \cup \mathcal{B}$:**
 - If a non-critical vertex $a \in \mathcal{A}$ is unmatched in M then we add a to \mathcal{A}_t .
 - If a critical vertex $a \in \mathcal{A}$ is unmatched in M then we add a to \mathcal{A}_{s+t} .
 - If a non-critical vertex $b \in \mathcal{B}$ is unmatched in M then we add b to \mathcal{B}_t .
 - If a critical vertex $b \in \mathcal{B}$ is unmatched in M then we add b to \mathcal{B}_0 .

Procedure TiesPropose(a^ℓ , List(a), M , Q)

```

1 Let  $b$  be the favourite neighbour of  $a$  in Pref( $a$ ) at rank  $k$ 
2 if  $b$  was marked by  $a$  then  $a$  unmarks  $b$ 
3 if  $b$  is unmatched then
4    $M = M \cup \{(a^\ell, b)\}$ 
5   if there exists an unmatched  $b''$  at rank  $k$  in Pref( $a$ ) then
6     Set  $(a^\ell, b)$  as uncertain proposal
7 else if  $b$  is part of an uncertain proposal  $(a_j^y, b)$  then
8    $M = (M \setminus \{(a_j^y, b)\}) \cup \{(a^\ell, b)\}$ 
9    $a_j^y$  marks  $b$  and add  $a_j^y$  to  $Q$ 
10 else if  $b$  is not part of an uncertain proposal then
11   Let  $a_j^y = M(b)$ 
12   if  $\ell == t$  then
13     if  $(y < t)$  or  $((y == t \text{ or } y == t^*) \text{ and } a \succ_b a_j)$  then
14        $M = M \setminus \{(a_j^y, b)\} \cup \{(a^\ell, b)\}$  and add  $a_j^y$  to  $Q$ 
15     else add  $a^\ell$  to  $Q$ 
16   if  $\ell == t^*$  then
17     if  $(y < t)$  or  $(y == t \text{ and } a \succeq_b a_j)$  or  $(y == t^* \text{ and } a \succ_b a_j)$ 
18       then
19          $M = M \setminus \{(a_j^y, b)\} \cup \{(a^\ell, b)\}$  and add  $a_j^y$  to  $Q$ 
20     else add  $a^\ell$  to  $Q$ 

```

It is convenient to visualize the partitions as shown in Figure 2. This particular drawing of the graph G is denoted by G_M throughout the rest of the section. It is useful to assume that the edges in G_M are implicitly directed from \mathcal{A} to \mathcal{B} . By construction, the edges of M (shown in blue colour) are horizontal whereas the unmatched edges (shown as solid black edges) can be horizontal, upwards or downwards.

We state the properties of the vertices and edges in G_M with respect to this partition in Property 1. We briefly justify the properties in Property 1 as follows. Only critical vertices in \mathcal{A} attain levels above t . This implies that the partition set $\mathcal{A}_{t+1}, \dots, \mathcal{A}_{s+t}$ contain only critical $a \in \mathcal{A}$. Thus, we have Property 1(1). Since each $a \in \mathcal{A}$ at a level at most $t-1$ does not propose to any non-critical vertex, the matched partner of each a^x for $x \leq t-1$ is a critical vertex. Also, all the unmatched non-critical vertices on \mathcal{B} -side are only in \mathcal{B}_t . This implies that the partition set $\mathcal{B}_0, \dots, \mathcal{B}_{t-1}$ contain only critical $b \in \mathcal{B}$. Thus, we have Property 1(2). If a vertex a remains unmatched in M then by the design of our algorithm it must have exhausted its preference list at level $s+t$ (if it is a critical vertex) or at level t , more specifically t^* , (if it is a non-critical vertex) and got rejected by each of its neighbours. Recall that each b prefers a higher level a over any lower level a' irrespective of the ranks of a and a' in $\text{Pref}(b)$. Thus, Property 1(3) and Property 1(4) hold in G_M . Observe that if any vertex $b \in \mathcal{B}$ receives a proposal then it cannot remain unmatched in M . This implies, if a critical b is unmatched then none of its neighbours has proposed it at level 0, which further implies that they have not exhausted $\text{PrefSC}(a)$ at level 0. By construction, b is in \mathcal{B}_0 . Hence, we have Property 1(5). Similarly, if b is non-critical and unmatched then none of its neighbours can go to level $t+1$ or above. By construction, b is in \mathcal{B}_t . Hence, we have Property 1(6).

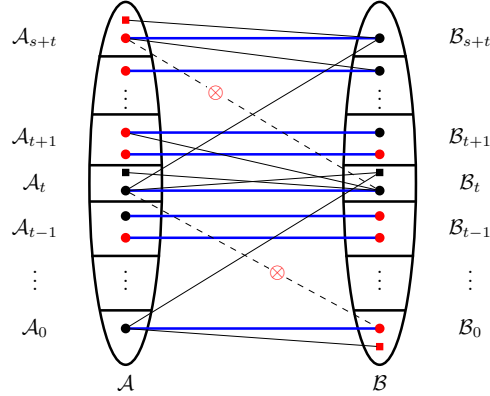


Fig. 2: The graph G_M . Red vertices are critical and black vertices are non-critical. Matched vertices are represented by circles and unmatched vertices are represented by squares. The blue horizontal lines represent matched edges in M . Solid black lines represent edges which are not matched in M . Dashed black lines marked with crossed red circles represent steep downward edges that are not present in G_M (Lemma 1).

Property 1. Let $a \in \mathcal{A}$ and $b \in \mathcal{B}$. Then the following hold in graph G_M .

1. If $a \in \bigcup_{x=t+1}^{t+s} \mathcal{A}_x$ then a is critical. Thus, $|\bigcup_{x=t+1}^{t+s} \mathcal{A}_x| \leq s$.

2. If $b \in \bigcup_{x=0}^{t-1} \mathcal{B}_x$ then b is critical. Thus, $|\bigcup_{x=0}^{t-1} \mathcal{B}_x| \leq t$.
3. If a is critical and is unmatched in M then $a \in \mathcal{A}_{s+t}$ and all the neighbours of a are matched and present in \mathcal{B}_{s+t} only.
4. If a is not critical and is unmatched in M then $a \in \mathcal{A}_t$ and all the neighbours of a are matched and present in \mathcal{B}_x for $x \geq t$.
5. If b is critical and is unmatched in M then $b \in \mathcal{B}_0$ and all the neighbours of b are present in \mathcal{A}_0 only.
6. If b is not critical and is unmatched in M then $b \in \mathcal{B}_t$ and all the neighbours of b are present in \mathcal{A}_x for $x \leq t$.

Let $(a, b) \in E$ be an edge such that $a \in \mathcal{A}_x$ and $b \in \mathcal{B}_y$. We say that such an edge is of the form $\mathcal{A}_x \times \mathcal{B}_y$. Lemma 1 below gives an important property about the edges which cannot be present in G_M . An edge of the form $\mathcal{A}_x \times \mathcal{B}_y$ with $x > y + 1$ is referred to as a *steep downward* edge.

Lemma 1. *The graph G_M does not contain steep downward edges. That is, there is no edge in G_M of the form $\mathcal{A}_x \times \mathcal{B}_y$ such that $x > y + 1$.*

Proof. Let (a, b) be any edge in G_M . If b is unmatched then irrespective of whether b is critical or not by Property 1(5) and Property 1(6), we have $x \leq y$. Now suppose $(a', b) \in M$. If $a = a'$ then by construction of G_M , $(a, b) \in \mathcal{A}_x \times \mathcal{B}_x$ for some $x \in \{0, \dots, s+t\}$. If $a \neq a'$ then we use the following claim (Claim 1). It is immediate from this claim that b is in \mathcal{B}_z for $z \geq x - 1$.

Claim 1 *Let $(a, b) \in E \setminus M$ and b be matched in M to \tilde{a} at level y , that is, $M(b) = \tilde{a}^y$. If the level x of a is at least 2 then $y \geq x - 1$.*

Proof of Claim 1: Suppose for contradiction that there exists $\tilde{a} \in \mathcal{A}$ such that $(\tilde{a}^y, b) \in M$ for $y < x - 1$. The fact that $(a, b) \in E$ and a achieves the level x implies that a remains unmatched after a^{x-1} exhausted its preference list $\text{Pref}(a)$, $\text{PrefS}(a)$ or $\text{PrefSC}(a)$ as appropriate. Note that if b receives a proposal from a vertex $\tilde{a} \in \mathcal{A}$ at levels below $x - 1$ then b is also available to receive proposals from vertices in \mathcal{A} at levels $\geq x - 1$. This is because when a vertex in \mathcal{A} transitions to a higher level, it proposes to possibly a superset of vertices that it proposes to in the lower level (recall that $\text{Pref}(a)$ and $\text{PrefS}(a)$ is a superset of $\text{PrefSC}(a)$). Furthermore, if a vertex in \mathcal{B} receives a proposal from some $a' \in \mathcal{A}$ at a level z then it is available to receive a proposal from all its neighbours proposing at level z .

Since b is matched to a vertex at level $y < x - 1$, it must be the case that b has received a proposal from a^{x-1} and it accepted this proposal by rejecting \tilde{a}^y because $y < x - 1$. Recall that a vertex $b \in \mathcal{B}$ always prefers a over \tilde{a} if a is at a higher level than that of \tilde{a} . Thus, $(a, b) \in M$ and we get a contradiction to the fact that $(a, b) \notin M$. □

This completes the proof of Lemma 1. □

Lemma 2. *Let (a, b) be a blocking pair w.r.t. M . Then the corresponding edge in G_M is an upward edge.*

Proof. For the blocking pair (a, b) let a and b be at levels x and y , respectively. First, suppose that b is a critical vertex. Since (a, b) is a blocking pair, irrespective of whether a is matched or unmatched, a^x must have proposed to critical b . Thus, b cannot remain unmatched. Thus, $M(b)$ exists. Since $(a, b) \notin M$, it must be the case that b rejected a^x . The fact that $a \succ_b M(b)$ and b rejected a^x implies that $M(b)$ is at a level $y > x$. Thus, the (a, b) edge is an upward edge in G_M .

Now, suppose that b is a non-critical vertex. Then by the construction of G_M , $b \in \mathcal{B}_y$ for $y \geq t$. If $x < t$ then we are done. So, assume that $x \geq t$. Since $x \geq t$, a^x is allowed to propose to *all* of its neighbours. Again, since (a, b) is a blocking pair, irrespective of whether a is matched or unmatched, a^x must have proposed b . Thus, $M(b)$ exists. Since $(a, b) \notin M$, b rejected a^x for $x \geq t$. The fact that $a \succ_b M(b)$ implies $M(b)$ must be at a level y such that $y > x$. Thus, (a, b) edge is an upward edge in G_M . \square

Now, we show that the matching M output by Algorithm 1 is critical. To prove the criticality of M , we use a property of an arbitrary critical matching N which is given in Claim 2. Basically, Claim 2 states that no matching matches more critical vertices from a particular side \mathcal{A} or \mathcal{B} than a critical matching. In other words, the number of critical vertices matched from \mathcal{A} -side or \mathcal{B} -side is optimum in any critical matching. This also implies that the number of critical vertices matched from \mathcal{A} -side or \mathcal{B} -side is invariant across all critical matchings. That is, if a critical matching M_1 matches p number of critical vertices from \mathcal{A} and q number of critical vertices from \mathcal{B} then any other critical matching, say M_2 , also matches p many critical vertices from \mathcal{A} and q many critical vertices from \mathcal{B} . This claim is similar to the one in [21].

Claim 2 *Let N be any critical matching and M be any matching in G . Then the number of critical vertices matched in N from \mathcal{A} is at least the number of critical vertices matched in M from \mathcal{A} . Similarly, the number of critical vertices matched in N from \mathcal{B} is at least the number of critical vertices matched in M from \mathcal{B} .*

Proof. Here we will prove the first statement, that is, we show that the number of critical vertices matched in a critical matching N from \mathcal{A} is at least the number of critical vertices matched in any matching M from \mathcal{A} . The proof for the second statement is symmetric.

Consider the symmetric difference $N \oplus M$. Suppose for contradiction that N matches strictly less number of critical vertices from \mathcal{A} than that of M . This implies there must exist a maximal alternating path $\rho = \langle u, u' \dots, v \rangle$ starting at an unmatched critical vertex $u \in \mathcal{A}$ in N such that using ρ we obtain a matching $N' = N \oplus \rho$ which matches strictly more number of critical vertices from \mathcal{A} than N on ρ . Note that ρ is a maximal M - N alternating path starting with an M edge (u, u') such that critical u is unmatched in N but matched in M . We consider the two cases below depending on the parity of the length of ρ .

If ρ is of odd length then it is an augmenting path with respect to N . That is, critical u becomes matched in N' from unmatched in N whereas the matched/unmatched status of all other vertices on ρ , except v , remains the same

as in N . Note that ρ is an augmenting path for N and hence v gets matched in N' from unmatched in N . Thus, N' matches more number of critical vertices from \mathcal{A} than that of N . Also, note that all the vertices from \mathcal{B} , except v , remain matched/unmatched as they were in N . That is, the number of matched critical vertices from \mathcal{B} either increases by 1 (when v is critical) or remains the same. Thus N' matches more number of critical vertices overall than that of a critical matching N – a contradiction.

If ρ is of even length then due to the maximality of ρ , the other endpoint $v \in \mathcal{A}$ is unmatched in M and hence, ρ ends with an N -edge. Note that $v \in \mathcal{A}$ is unmatched in M and u is critical but matched in M . If v is a critical vertex then the number of critical vertices matched in N' and N remain the same. This contradicts the selection of our path ρ . Recall ρ is an alternating path such that $N' = N \oplus \rho$ matches strictly more number of critical vertices from \mathcal{A} . Thus, v is not a critical vertex. But then the number of critical vertices matched in N' is strictly less than that of N . This contradicts the selection of our path ρ .

Thus, we conclude that such a path ρ does not exist and hence the number of critical vertices matched in N from \mathcal{A} is at least the number of critical vertices matched in M from \mathcal{A} . \square

Lemma 3. *The output matching M is critical for G .*

Proof. We prove the criticality of M by using the level structure of the graph G_M . The idea is to show that there is no alternating path ρ in G_M w.r.t. M such that $M \oplus \rho$ results in more critical vertices matched than in M . We prove the criticality in two parts. First, we prove (\mathcal{A} -part) where we show that M matches the maximum possible critical vertices from the set $\mathcal{A} \cap \mathcal{C}$ and then we prove (\mathcal{B} -part) where we show that M matches the maximum possible critical vertices from the set $\mathcal{B} \cap \mathcal{C}$. Thus, by using Claim 2 above, we conclude that M is critical. Let N be any critical matching in G .

Proof of (\mathcal{A} -part): Suppose for contradiction that M does not match the maximum possible critical vertices from the set $\mathcal{A} \cap \mathcal{C}$. This implies that there exists an alternating path ρ in $M \oplus N$ such that N matches more critical vertices from \mathcal{A} on ρ than in M . Let $\rho = \langle u_0, v_1, u_1, v_2, u_2, \dots, v_k, u_k, \dots \rangle$ where $(v_i, u_i) \in M$ and the other edges of ρ are in the matching N . Furthermore, assume that the first vertex u_0 represents a vertex $a \in \mathcal{A}$ such that critical a is matched in N but unmatched in M . Since critical a is unmatched in M , by Property 1(3), $a \in \mathcal{A}_{s+t}$. Thus, ρ starts at level $s+t$ in G_M , that is, $u_0 \in \mathcal{A}_{s+t}$. Since $u_0 = a$ is critical and unmatched in M , by Property 1(3), $v_1 \in \mathcal{A}_{s+t}$ and $u_1 = M(v_1)$ is in \mathcal{A}_{s+t} . The other end of ρ can be in \mathcal{A} or in \mathcal{B} . We consider both these cases below.

The path ρ ends at a vertex in \mathcal{A} : Suppose that the path ends at a vertex in \mathcal{A}_x for $x > t$. By Property 1(1), all the vertices in \mathcal{A}_x for $x > t$ are critical. Thus, if ρ ends at a vertex u_i such that $u_i \in \mathcal{A}_x$ for $x > t$ then $N \oplus \rho$ matches the same number of critical vertices from \mathcal{A} . This contradicts the choice of our path ρ (recall that we selected ρ such that N matches more critical vertices from \mathcal{A} on ρ than in M). This implies that the other endpoint of ρ must be in \mathcal{A}_x

for $x \leq t$. Lemma 1 implies that if $u_i \in \mathcal{A}_x$ and $u_{i+1} \in \mathcal{A}_y$ then $|y - x| \leq 1$ for all indices i on ρ . Hence, ρ must contain at least one vertex from each \mathcal{A}_x for $t + 1 \leq x \leq s + t$. We observe that ρ contains at least two vertices u_0 and u_1 from \mathcal{A}_{s+t} , and at least one vertex from each \mathcal{A}_x for $t + 1 \leq x \leq s + t$. Thus, ρ contains at least $s + 1$ many critical vertices from $\bigcup_{x=t+1}^{s+t} \mathcal{A}_x$. By Property 1(1), the total number of vertices accommodated in these levels is at most s . Thus, we get a contradiction.

The path ρ ends at some vertex in \mathcal{B} : Note that ρ has even length and hence the last vertex, say v_{k+1} , on ρ remains unmatched in M . By construction, an unmatched vertex $b \in \mathcal{B}$ are in $\mathcal{B}_t \cup \mathcal{B}_0$. Thus, by Property 1(5) and Property 1(6), $u_k \in \mathcal{A}_x$ for $x \leq t$. Since ρ contains some vertex in \mathcal{A}_x for $x \leq t$, by using the same argument as in the previous case, we show that ρ contains at least $s + 1$ many critical vertices from $\bigcup_{x=t+1}^{s+t} \mathcal{B}_x$ to get a contradiction. Hence, we conclude that such a path ρ cannot exist.

Proof of (\mathcal{B} -part): Suppose for contradiction that M does not match the maximum possible critical vertices from the set $\mathcal{B} \cap \mathcal{C}$. This implies that there exists an alternating path ρ in $M \oplus N$ such that N matches more critical vertices from \mathcal{B} on ρ than in M . Let $\rho = \langle v_0, u_1, v_1, u_2, v_2, \dots, u_k, v_k, \dots \rangle$ where $(u_i, v_i) \in M$ and the other edges of ρ are in the matching N . Furthermore, assume that the first vertex v_0 represents a vertex $b \in \mathcal{B}$ such that critical b is matched in N but unmatched in M . Since b is critical and unmatched in M , by Property 1(5), $b \in \mathcal{B}_0$. Thus, ρ starts at level 0 in G_M , that is, $v_0 \in \mathcal{B}_0$. Since $v_0 = b$ is critical and unmatched in M , by Property 1(5), $u_1 \in \mathcal{A}_0$ and $v_1 = M(u_1)$ is in \mathcal{B}_0 . The other end of ρ can be in \mathcal{B} or in \mathcal{A} . We consider both these cases below.

The path ρ ends at a vertex in \mathcal{B} : Suppose that the path ends at a vertex in \mathcal{B}_x for $x < t$. By Property 1(2), all the vertices in \mathcal{B}_x for $x < t$ are critical. Thus, if ρ ends at a vertex v_i such that $v_i \in \mathcal{B}_x$ for $x < t$ then $N \oplus \rho$ matches the same number of critical vertices from \mathcal{B} . This contradicts the choice of our path ρ (recall that we selected ρ such that N matches more critical vertices from \mathcal{B} on ρ than in M). This implies that the other endpoint of ρ must be in \mathcal{B}_x for $x \geq t$. Lemma 1 implies that if $v_i \in \mathcal{B}_x$ and $v_{i+1} \in \mathcal{B}_y$ then $y - x \leq 1$ for all indices i on ρ . Hence, ρ must contain at least one vertex from each \mathcal{B}_x for $1 \leq x \leq t - 1$. We observe that ρ contains at least two vertices v_0 and v_1 from \mathcal{B}_0 , and at least one vertex from each \mathcal{B}_x for $1 \leq x \leq t - 1$. Thus, ρ contains at least $t + 1$ many critical vertices from $\bigcup_{x=0}^{t-1} \mathcal{B}_x$. By Property 1(2), the total number of vertices accommodated in these levels is at most t . Thus, we get a contradiction.

The path ρ ends at some vertex in \mathcal{A} : Note that ρ has even length and hence the last vertex, say u_{k+1} , on ρ remains unmatched in M . By construction, an unmatched vertex $a \in \mathcal{A}$ are in $\mathcal{A}_t \cup \mathcal{A}_{s+t}$. Thus, by Property 1(4) and Property 1(3), $v_k \in \mathcal{B}_x$ for $x \geq t$. Since ρ contains some vertex in \mathcal{B}_x for $x \geq t$, by using the same argument as in the previous case, we show that ρ contains at least $t + 1$ many critical vertices from $\bigcup_{x=0}^{t-1} \mathcal{B}_x$ to get a contradiction. Hence, we conclude that such a path ρ cannot exist.

□

Lemma 4. *The output matching M of Algorithm 1 is RSM for G .*

Proof. If there is no blocking pair w.r.t. M then we are done. Hence, assume that (a, b) is a blocking pair w.r.t. M . By Lemma 2, (a, b) is an upward edge. We consider two cases based on the level of b .

Case 1: $level(b) \leq t$. Clearly, $level(a) \leq t - 1$. Thus, by the construction of G_M , a is matched and hence $M(a)$ exists. Clearly, $M(a)$ is at level at most $t - 1$. This implies, $M(a)$ is critical. Hence, the blocking pair (a, b) is justified by Condition 1 of Definition 2.

Case 2: $level(b) > t$. By construction of G_M , b is matched. Thus, $M(b)$ exists and $M(b) \in \mathcal{A}_x$ for $x \geq t + 1$. By Property 1(1), $M(b)$ is critical. Hence, the blocking pair (a, b) is justified by Condition 2 of Definition 2. \square

Lemma 5. *Let M' be any maximum size CRITICAL-RSM and M be the output of Algorithm 1 for an instance of our problem. Then $|M| \geq \frac{2}{3} \cdot |M'|$.*

Proof. We prove that $M \oplus M'$ does not admit any 1-length or 3-length augmenting path w.r.t. M . This immediately implies that $|M| \geq \frac{2}{3} \cdot |M'|$. If a is unmatched (critical or otherwise), we know from Property 1(3) and Property 1(4) that no neighbour b of a is unmatched in M . Thus, M is a maximal.

For contradiction assume that $M \oplus M'$ contains a 3-length augmenting path $\rho = \langle a_1, b, a, b_1 \rangle$ w.r.t. M . Here $(a, b) \in M$ and other two edges are in M' . We show that (a, b) blocks M' and the blocking pair is not justified. This will contradict relaxed stability of M' . We first establish the levels of the vertices.

Levels of vertices: The fact that a_1 remains unmatched in M implies that $a_1^{t^*}$ exhausted $\text{Pref}(a_1)$. Thus, a_1 is at level at least t^* . Since b_1 remains unmatched in M , a did not exhaust $\text{Pref}(a)$ at the level t . Thus, a is at level at most t . We claim that a_1 is not at level $t + 1$ or higher. If a_1 is at level $x \geq t + 1$ then a_1^x must have proposed to b as a_1 is unmatched in M . Since a is at level at most t , b must reject a and accept a_1 – a contradiction to $(a, b) \in M$. Thus, we conclude that a_1 is at level t^* . Now, if a is at level $y < t$ then b must reject a and accept a_1 as a_1 at level t^* proposed to it. Recall that t^* is a sub-level of t used in the algorithm, and t^* does appear as a separate level in G_M . Thus, the vertices $a, a_1 \in \mathcal{A}_t$.

The pair (a, b) blocks M' : Since $a_1^{t^*}$ was rejected by b , it implies $M(b) = a$ and a_1 cannot be in tie for b , otherwise b would not have rejected a $*$ status vertex over a non $*$ status vertex. Thus, $a \succ_b a_1$. Now, we show that $b \succ_a b_1$. Suppose not. Then, if $b_1 \succ_a b$ then a^t must have proposed to b_1 before b and got matched to it – a contradiction that b_1 is unmatched. Hence, assume that $b =_a b_1$. In this case, when a^t proposes to b , the vertex b must also be unmatched, otherwise b cannot be favourite neighbour of a^t . This implies that a_1 proposes to b only *after* a proposes to b . Since b_1 was unmatched when a proposed to b , the proposal from a to b was uncertain. Hence, when b received a proposal from a_1 it must have accepted it by rejecting a . Since a has an unmatched neighbour b_1 at the same rank, a must have proposed b_1 before proposing to b again. This implies b_1 is matched, a contradiction. Thus, $b \succ_a b_1$; hence (a, b) blocks M' .

The blocking pair (a, b) is not justified: In order to prove this, we show $b_1 = M'(a)$ and $a_1 = M'(b)$ are both non-critical. Note that b_1 is unmatched in M , hence if it is critical then $b_1 \in \mathcal{B}_0$ and the number of critical vertices on \mathcal{B} -side is at least 1 (that is $t \geq 1$). This implies that a cannot be at a level ≥ 1 since it has not yet proposed to at least one critical neighbour, namely b_1 . Thus, b_1 is not critical. We finish the proof by showing that a_1 is also not critical. Note that a_1 is unmatched in M , hence, if it is critical then $a_1 \in \mathcal{A}_{s+t}$ and $s > 0$. This implies that b cannot be in \mathcal{B}_x for $x < s + t$ (Property 1(3)). This is a contradiction that $b \in \mathcal{B}_t$ and $s > 0$. Thus, a_1 is not critical.

This finishes the proof that the claimed 3-length augmenting path w.r.t. M does not exist establishing the size guarantee. \square

Using Lemma 3, Lemma 4 and Lemma 5, we establish Theorem 1.

References

1. Biró, P., Irving, R.W., Manlove, D.F.: Popular matchings in the marriage and roommates problems. In: International Conference on Algorithms and Complexity. pp. 97–108. Springer (2010)
2. Dean, B., Jalsutram, R.: Factor revealing lps and stable matching with ties and incomplete lists. In: Proceedings of the 3rd International Workshop on Matching Under Preferences. pp. 42–53 (2015)
3. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly **69**(1), 9–15 (1962), <http://www.jstor.org/stable/2312726>
4. Gärdenfors, P.: Match making: Assignments based on bilateral preferences. Behavioral Science **20**(3), 166–173 (1975)
5. Goko, H., Makino, K., Miyazaki, S., Yokoi, Y.: Maximally satisfying lower quotas in the hospitals/residents problem with ties. In: 39th International Symposium on Theoretical Aspects of Computer Science (2022)
6. Halldórsson, M.M., Iwama, K., Miyazaki, S., Yanagisawa, H.: Improved approximation results for the stable marriage problem. ACM Transactions on Algorithms (TALG) **3**(3), 30-es (2007)
7. Hamada, K., Iwama, K., Miyazaki, S.: The hospitals/residents problem with lower quotas. Algorithmica **74**(1), 440–465 (2016), <https://doi.org/10.1007/s00453-014-9951-z>
8. Hamada, K., Miyazaki, S., Yanagisawa, H.: Strategy-proof approximation algorithms for the stable marriage problem with ties and incomplete lists. In: International Symposium on Algorithms and Computation. Schloss Dagstuhl-Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing (2019)
9. Huang, C.C., Kavitha, T.: Improved approximation algorithms for two variants of the stable marriage problem with ties. Mathematical Programming **154**, 353–380 (2015)
10. Iwama, K., Miyazaki, S., Yanagisawa, H.: A 25/17-approximation algorithm for the stable marriage problem with one-sided ties. Algorithmica **68**(3), 758–775 (2014)
11. Kavitha, T.: Matchings, critical nodes, and popular solutions. In: 41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021) (2021)

12. Király, Z.: Better and simpler approximation algorithms for the stable marriage problem. *Algorithmica* **60**(1), 3–20 (2011)
13. Király, Z.: Linear time local approximation algorithm for maximum stable marriage. *Algorithms* **6**(3), 471–484 (2013), <https://doi.org/10.3390/a6030471>
14. Krishnaa, P., Limaye, G., Nasre, M., Nimbhorkar, P.: Envy-freeness and relaxed stability: Hardness and approximation algorithms. In: *Algorithmic Game Theory: 13th International Symposium, SAGT 2020, Augsburg, Germany, September 16–18, 2020, Proceedings 13*. pp. 193–208. Springer (2020)
15. Krishnaa, P., Limaye, G., Nasre, M., Nimbhorkar, P., et al.: Envy-freeness and relaxed stability: hardness and approximation algorithms. *Journal of Combinatorial Optimization* **45**(1), 1–30 (2023)
16. Lam, C.K., Plaxton, C.G.: A $(1 + 1/e)$ -approximation algorithm for maximum stable matching with one-sided ties and incomplete lists. In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. pp. 2823–2840. SIAM (2019)
17. 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)
18. McDermid, E.: A $3/2$ -approximation algorithm for general stable marriage. In: *International Colloquium on Automata, Languages, and Programming*. pp. 689–700. Springer (2009)
19. Nasre, M., Nimbhorkar, P.: Popular matchings with lower quotas. In: *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017)*. pp. 44:1–44:15 (2017), <https://doi.org/10.4230/LIPIcs.FSTTCS.2017.44>
20. Nasre, M., Nimbhorkar, P., Ranjan, K., Sarkar, A.: Popular Matchings in the Hospital-Residents Problem with Two-Sided Lower Quotas. In: *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*. *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 213, pp. 30:1–30:21 (2021)
21. Nasre, M., Nimbhorkar, P., Ranjan, K., Sarkar, A.: Popular critical matchings in the many-to-many setting. *arXiv* (2023)
22. Paluch, K.: Faster and simpler approximation of stable matchings. *Algorithms* **7**(2), 189–202 (2014)
23. Robards, P.A.: Applying two-sided matching processes to the united states navy enlisted assignment process. Tech. rep., *NAVAL POSTGRADUATE SCHOOL MONTEREY CA* (2001)
24. Roth, A.E.: Stability and polarization of interests in job matching. *Econometrica: Journal of the Econometric Society* pp. 47–57 (1984)
25. Roth, A.E.: On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica: Journal of the Econometric Society* pp. 425–427 (1986)
26. Tan, S.J., Yeong, C.M.: Designing economics experiments to demonstrate the advantages of an electronic employment market in a large military organization. Tech. rep., *NAVAL POSTGRADUATE SCHOOL MONTEREY CA* (2001)
27. Yang, W., Sycara, K.: Two-sided matching for the us navy detailing process with market complication. Tech. rep., *Technical Report CMU-RI-TR-03-49, Robotics Institute, Carnegie-Mellon University* (2003)
28. Yokoi, Y.: Envy-free matchings with lower quotas. *Algorithmica* **82**(2), 188–211 (2020), <https://doi.org/10.1007/s00453-018-0493-7>

A Relaxed stability versus popularity

Relaxed stability and popularity do not define the same set of matchings even when preferences are strict and critical vertices are restricted to only one side of the bipartition. That is, neither one implies the other. We give simple examples (from [15]).

Consider the example shown in Figure 3(i). Notice that the popular matching $M_1 = \{(a_1, b_2), (a_2, b_1)\}$ is not relaxed stable because the blocking pair (a_1, b_1) is not justified (there are no critical vertices). Observe that in the absence of critical vertices, relaxed stability is the same as stability.

Now, consider the example shown in Figure 3(ii). Notice that the matching $M_2 = \{(a_1, b_2)\}$ is relaxed stable as the only blocking pair (a_1, b_1) is justified by a_1 . But $M_3 = \{(a_1, b_1)\}$ is more popular than M_2 . Thus a relaxed stable matching M_2 is not popular.

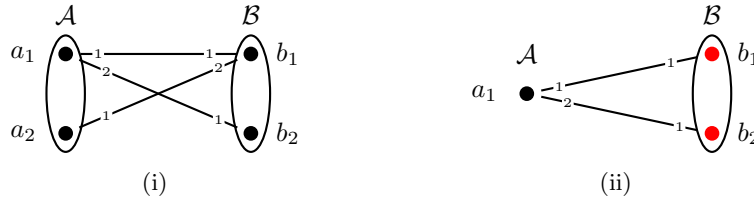


Fig. 3: The red vertices are critical and black vertices are non-critical. The numbers on the edges denote the ranking of the other endpoint in the preference list of that vertex.