Critical Relaxed Stable Matchings with Two-Sided Ties

Meghana Nasre[1], Prajakta Nimbhorkar[2], and Keshav Ranjan[1(✉)]

[1] IIT Madras, Chennai, India
meghana@cse.iitm.ac.in, ranjankeshav08@gmail.com
[2] Chennai Mathematical Institute and UMI ReLaX, Chennai, India
prajakta@cmi.ac.in

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 in our setting a critical matching which is relaxed stable always exists although computing a maximum-sized relaxed stable matching turns out to be NP-hard. Our main contribution is a $\frac{3}{2}$-approximation to the maximum-sized critical relaxed stable matching for the stable marriage problem where ties as well as critical vertices are present on both the sides of the bipartition.

Keywords: Stable Matching · Ties in Preferences · 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 the order of its preference possibly

D. Paulusma and B. Ries (Eds.): WG 2023, LNCS 14093, pp. 447–461, 2023.
https://doi.org/10.1007/978-3-031-43380-1_32

involving ties – this ordering is denoted as $\mathsf{Pref}(u)$. For a vertex u let v_1 and v_2 be two of 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 $\mathsf{Pref}(u)$. In addition, the input consists of a set $\mathcal{C} \subseteq (\mathcal{A} \cup \mathcal{B})$ of *critical vertices*. Our goal is to compute an assignment which minimizes the number of unassigned critical vertices.

Formally, an assignment or 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, we consider that the most important attribute of a 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.

Critical vertices or lower-quota positions naturally arise in applications like the Hospitals/Residents problem [7], where rural hospitals must be prioritized to ensure sufficient staffing. Another example is the problem of assigning sailors to billets [28] in the US Navy, where some critical billets cannot be left vacant [25,29]. Ties in preferences is yet another important practical consideration in matching problems and has been extensively investigated in the literature [2,8,9,13,18,19,24]. However, there is a limited investigation (see for example [5]) of matching problems with ties as well critical vertices 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 a 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 $a \succ_b M(b)$. A 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 well-known Gale and Shapley algorithm [3]. In addition, it is also known [26,27] that all stable matchings have the same size.

Stable Matchings in the Presence of Ties: When preferences are allowed to have ties, the notion of stability defined above is called 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 the stable matchings need not have the same size, and the problem of computing a maximum or minimum size stable matching is NP-hard [18] under

severe restrictions – e.g. when ties occur at the end of preference lists and only on one side of the bipartition, 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 we have critical vertices as a part of the input, a stable matching which is also critical, may not exist – for example, consider an instance of the stable matching problem with strict lists obtained by arbitrarily breaking ties in the preference lists of all agents in the example shown in Fig. 1. Any critical matching in the instance must match b_2 with a_1, resulting in the blocking edge (a_1, b_1). Since stability and criticality are not simultaneously guaranteed, an alternate notion of optimality, namely *popularity* [4], is extensively investigated in the literature [11,20,22] for the case of strict lists. The goal is to compute a matching which is *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 [11,22] 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, popular matchings are not guaranteed to exist even when ties are present in the preferences only on one side of the bipartition, without any critical vertices. Moreover, in the presence of ties, deciding whether a popular matching exists is NP-hard [1]. 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 Hospitals/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, and 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 an RSM if every blocking pair (a, b) w.r.t. M is *justified* by either a or b or both. 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 at least 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.*

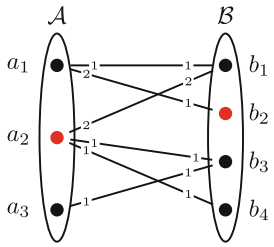

Fig. 1. Red vertices are critical, black vertices are non-critical. The numbers on the edges denote the ranks of the respective end-points. The instance does not admit any critical stable matching because b_2 remains unmatched in every stable matching. $M_1 = \{(a_1, b_2), (a_2, b_1), (a_3, b_3)\}$ is 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. (Color figure online)

A matching M is called *a critical relaxed stable matching (*CRITICAL-RSM*)* if it is *critical* as well as *relaxed stable*. In the instance shown in Fig. 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 as the 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 [18] and hard to approximate within any factor smaller than $\frac{21}{19}$ [6]. For the problem of computing a maximum-sized stable matching when ties appear on both sides of the bipartition, the current best approximation factor [13,19,24] 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 where ties and critical vertices can appear in both the bipartitions of G. Then G always admits a* CRITICAL-RSM *M such that $|M| \geq \frac{2}{3}|M'|$, where M' is a maximum size* CRITICAL-RSM *in G. Moreover, M can be computed in polynomial-time.*

Related Work: As mentioned earlier, the generalizations of the stable marriage problem to allow either ties in preferences or critical vertices/lower-quota positions has been extensively investigated. Very recently, Goko *et al.* [5] and Makino *et al.* [17] have considered the instances with both ties and critical vertices. They study the Hospitals/Residents problem with lower-quotas where ties appear on both sides. In their setting, only one side of the bipartition can have critical vertices. They define a matching with maximum satisfaction ratio, which for our one-to-one setting, coincides with critical matchings. However, their goal is to compute a matching that matches the maximum possible critical vertices amongst all stable matchings.

For strict preferences and lower-quotas/critical vertices, various notions like envyfreeness [15,30], popularity [11,20,22,23], and relaxed stability [14,15] have been studied. Relaxed stability and popularity do not define the same set of matchings even in the one-to-one strict-list setting and critical vertices restricted to one side only (see full version [21]) for the details. Hamada *et al.* [7] consider the problem of computing a matching with minimum number of blocking pairs or blocking residents, and give approximation algorithms for the same.

For the stable marriage problem with ties (without critical vertices) there is a long line of investigation [2,9,10,12,13,19,24] in order to improve the approximation ratio of the maximum size stable matching under various restricted settings. The best-known approximation algorithm for the case when ties are allowed only in one bipartition of the graph is by Lam and Plaxton [16] whereas the best-known for the case where ties are allowed on both sides is by [13,19,24]. 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 instances where ties appear on both sides and (ii) Multi-level algorithm for computing popular critical matching [23] 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 $\mathsf{Pref}(a)$. 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 favourite neighbour 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 $\mathsf{Pref}(b)$. 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 b's preference 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 $\mathsf{Pref}(a)$. 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 $\mathsf{Pref}(a)$ with $*$ status. The $*$ status of a vertex a can be interpreted as improving the rank of a in $\mathsf{Pref}(b)$ 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 [23]. Now, we briefly describe the algorithm in [23] 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 [23] is a multi-level algorithm which first matches as many critical vertices from $\mathcal{B}$ as possible. This is achieved by restricting unmatched vertices in $\mathcal{A}$ at levels $0, \ldots, t-1$ to propose only to critical vertices on the $\mathcal{B}$-side. 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 $\mathsf{Pref}(b)$. It is shown in [23] 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 from $\mathcal{B}$ that $a \in \mathcal{A}$ proposes to depends on the level of a. 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 $\mathsf{Pref}(b)$ which possibly contains ties. For a vertex $a \in \mathcal{A}$, let $\mathsf{PrefS}(a)$ denote a *strict* preference list obtained by breaking ties in $\mathsf{Pref}(a)$ in such a way that the vertices in ties are ordered by increasing order of their indices. Furthermore, let $\mathsf{PrefSC}(a)$ be the strict list obtained from $\mathsf{PrefS}(a)$ by omitting all the non-critical vertices from $\mathsf{PrefS}(a)$. For example, assume $\mathsf{Pref}(a) = (b_2, b_1), b_5, (b_3, b_4)$ 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 $\mathsf{PrefS}(a) = b_1, b_2, b_5, b_3, b_4$ and $\mathsf{PrefSC}(a) = b_5, b_4$ 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, each $a \in \mathcal{A}$ proposes to vertices in $\mathsf{PrefSC}(a)$ (see Lines 4–8 of Algorithm 1). Each time it remains unmatched, it proposes to its *most preferred* neighbour b. The most preferred neighbour in $\mathsf{PrefSC}(a)$ or $\mathsf{PrefS}(a)$ 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 $\mathsf{PrefSC}(a)$ at a level $\ell < t - 1$, then a raises its level to $\ell + 1$ and again proposes to vertices in $\mathsf{PrefSC}(a)$. 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 $\mathsf{Pref}(b)$ do not matter) or
(ii) $\ell = \ell'$ and $a_i \succ_b a_j$.

If vertex a remains unmatched after exhausting $\mathsf{PrefSC}(a)$ at level $t - 1$, a attains level t where it uses its original preference list $\mathsf{Pref}(a)$ which may contain ties. At level t our algorithm executes Király's algorithm [13]. This corresponds to Lines 10–13 of Algorithm 1. Király's algorithm is encoded in the procedure `TiesPropose()`. Since we have ties on both sides of the graph, at this level, we need the notion of a favourite neighbour and uncertain proposal defined in Sect. 2. If the vertex a remains unmatched after exhausting $\mathsf{Pref}(a)$ 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 Sect. 2.

If a *critical* vertex a remains unmatched after exhausting its preference list $\mathsf{Pref}(a)$ at level t^*, a raises its level to $t + 1$, and starts proposing to vertices in $\mathsf{PrefS}(a)$ (see Lines 16–20 of Algorithm 1). It continues to do so until either it is matched or it has exhausted $\mathsf{PrefS}(a)$ at level $s + t$. In contrast, if a non-critical vertex a remains unmatched after exhausting its preference list $\mathsf{Pref}(a)$ at level t^*, a does not propose any further. Recall that $\mathsf{PrefS}(a)$ 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

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

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

(ii) all unmatched critical $a \in \mathcal{A}$ have exhausted $\mathsf{PrefS}(a)$ at level $s+t$ and all unmatched non-critical $a \in \mathcal{A}$ have exhausted $\mathsf{Pref}(a)$ 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 $O(n \cdot |E|)$.

It is worth noting that in our algorithm, not all vertices in $\mathcal{A}$ propose at all levels. Similarly, not all vertices in $\mathcal{B}$ receive proposals from vertices at all levels. In other words, only critical vertices in $\mathcal{B}$ are allowed to receive proposals from vertices in $\mathcal{A}$ at levels at most $t-1$, and only critical vertices in $\mathcal{A}$ are allowed to propose at levels above t. Also, note that 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 $\mathsf{Pref}(a)$ and its strict counterpart $\mathsf{PrefS}(a)$ are both a superset of $\mathsf{PrefSC}(a)$). Therefore, we have the following useful observation.

Observation 1. *If a vertex $b \in \mathcal{B}$ receives a proposal from some $a' \in \mathcal{A}$ at a level z then b receives proposals from all its neighbours who exhausted their preference list at level z.*

4 Correctness of Our Algorithm

We prove that the matching M output by Algorithm 1 is

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

Let b be the most preferred unproposed vertex by a^ℓ in List(a)
if *b is unmatched in M* **then**
  $M = M \cup \{(a^\ell, b)\}$
else
  Let $a_j^y = M(b)$
  if *($\ell > y$) or ($\ell == y$ and $a \succ_b a_j$)* **then**
    $M = M \setminus \{(a_j^y, b)\} \cup \{(a^\ell, b)\}$ and add a_j^y to Q
  else add a^ℓ to Q

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

Let b be the favourite neighbour of a^ℓ in List(a) at rank k
if b was marked by a^ℓ **then** a^ℓ unmarks b
if *b is unmatched* **then**
  $M = M \cup \{(a^\ell, b)\}$
  if *there exists an unmatched b'' at rank k in* Pref(a) **then**
    Set (a^ℓ, b) as uncertain proposal // $\ell = t$ as b'' is unmatched
else if *b is part of an uncertain proposal* (a_j^y, b) **then**
  $M = (M \setminus \{(a_j^y, b)\}) \cup \{(a^\ell, b)\}$ // Here, $y = t$
  a_j^y marks b and add a_j^y to Q
else if *b is not part of an uncertain proposal* **then**
  Let $a_j^y = M(b)$
  if $\ell == t$ **then**
    if *($y < t$) or (($y == t$ or $y == t^*$) and $a \succ_b a_j$)* **then**
      $M = M \setminus \{(a_j^y, b)\} \cup \{(a^\ell, b)\}$ and add a_j^y to Q
    else add a^ℓ to Q
  if $\ell == t^*$ **then**
    if *($y < t$) or ($y == t$ and $a \succeq_b a_j$) or ($y == t^*$ and $a \succ_b a_j$)* **then**
      $M = M \setminus \{(a_j^y, b)\} \cup \{(a^\ell, b)\}$ and add a_j^y to Q
    else add a^ℓ to Q

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

We define a partition of the vertices in $\mathcal{A} \cup \mathcal{B}$ based on the levels of vertices in $\mathcal{A}$ and the matching M. This partition is useful to establish the correctness of our algorithm.

Partition of Vertices: The vertex set $\mathcal{A}$ is partitioned into $\mathcal{A}_0 \cup \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_t \cup \ldots \cup \mathcal{A}_{s+t}$, and the vertex set $\mathcal{B}$ is partitioned into $\mathcal{B}_0 \cup \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_t \cup \ldots \cup \mathcal{B}_{s+t}$. For every matched vertex $a \in \mathcal{A}$ there exists $x \in \{0, \ldots, s+t\}$ 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 $x \in \{0, \ldots, s+t\}$. 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 $\mathcal{B}_0$.

It is convenient to visualize the partitions as shown in Fig. 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 (see the full version [21] for justification).

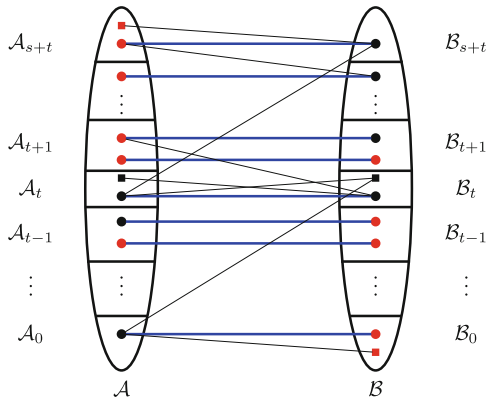

Fig. 2. The graph G_M. Red vertices are critical and black vertices are non-critical. Matched vertices are represented by circles, and unmatched vertices are represented by squares. The blue horizontal lines represent matched edges in M. Solid black lines represent edges which are not matched in M. Note that no edge in G_M is of the form $\mathcal{A}_x \times \mathcal{A}_y$ for $y \leq x - 2$. (Color figure online)

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

1. If $a \in \bigcup_{x=t+1}^{s+t} \mathcal{A}_x$ then a is critical. Thus, $|\bigcup_{x=t+1}^{s+t} \mathcal{A}_x| \leq s$.
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 such that $a \in \mathcal{A}_x$ and $b \in \mathcal{B}_y$. 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 that b is matched and $(a', b) \in M$. If $a = a'$ then by construction of G_M, $(a, b) \in \mathcal{A}_x \times \mathcal{B}_x$. If $a \neq a'$, then we use Claim 1, which is given below. It is immediate from this claim that b is in $\mathcal{B}_y$ for $y \geq x - 1$. □

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. 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 $\mathsf{Pref}(a)$, $\mathsf{PrefS}(a)$ or $\mathsf{PrefSC}(a)$ as appropriate. Since b is matched to a vertex at level $y < x - 1$, and a^{x-1} exhausted its preference list, by Observation 1, b received a proposal from a^{x-1}. At this time, b must accept this proposal by rejecting $\tilde{a}^y$ because $y < x-1$. This implies $(\tilde{a}^y, b) \notin M$ which contradicts our assumption that $(\tilde{a}^y, b) \in M$ for $y < x - 1$. □

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 the critical vertex b. Thus, b cannot remain unmatched. This implies $M(b)$ exists. We consider the following two cases:

1. The proposal by a to b results in (a, b) to be uncertain: Note that a^x is rejected by b because b receives another proposal, and hence a^x marks b. Since (a, b) is a blocking pair, $M(a)$ is ranked lower than b. However, before proposing to any vertex ranked strictly lower than b, a^x must propose to the marked vertex b. At this point, either b is matched to a better preferred partner than a which contradicts that (a, b) blocks M, otherwise, b accepts the proposal from a^x. Thus, a^x is matched to either b or to some other vertex on the same rank as b. This implies (a, b) is not a blocking edge.

2. The proposal by a to b does not result in (a, b) to be uncertain: 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.

Now, suppose that b is a non-critical vertex. Then by Property 1(2), $b \in \mathcal{B}_y$ for $y \geq t$. If $x < t$, then (a, b) is an upward edge, and we are done. Hence, assume that $x \geq t$. Since $x \geq t$, a^x is proposes 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 to b. Thus, b cannot be unmatched. Now, either b is matched to a better-preferred partner than a, which contradicts that (a, b) is a blocking pair or $M(b)$ is at a higher level than a and hence (a, b) is an upward edge. □

Lemma 3 below shows that the matching M output by Algorithm 1 is critical.

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

Proof sketch: 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 with respect to M such that the number of critical vertices matched in $M \oplus \rho$ is more than the number of critical vertices matched in M. We prove the criticality for the individual parts, that is, for $\mathcal{A}$-part and for $\mathcal{B}$-part. In other words, we show that M matches maximum possible critical nodes from $\mathcal{A}$-side, and maximum possible critical nodes from the $\mathcal{B}$-side. This immediately implies that M matches the maximum possible critical nodes that can be matched in *any* matching. Hence, M is critical. For the $\mathcal{A}$-part we show that the path $\rho = \langle u_0, v_1, u_1, \ldots \rangle$ begins at the highest level $s+t$ with an unmatched critical vertex $u_0 \in \mathcal{A}$. Using Property 1(5), we also show that at least the first two vertices on the $\mathcal{A}$-side (u_0 and u_1) on ρ are at the same level $s + t$. Then we argue that the other end of ρ must be at a level below $t + 1$. Since there are no steep downward edges (Lemma 1), the path contains at least one vertex from each level $t+1, \ldots, s+t-1$. Thus, we have at least $s+1$ many vertices in $\mathcal{A}_{t+1} \cup \ldots \cup \mathcal{A}_{s+t}$. This contradicts Property 1(1). Proof for the $\mathcal{B}$-part is analogous. See full version [21] for the complete proof. □

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: $b \in \mathcal{B}_y$ **for** $y \leq t$. Clearly, $a \in \mathcal{A}_x$ for $x \leq t-1$. Thus, by the construction of G_M, a is matched, and hence $M(a)$ exists. Clearly, $M(a)$ is at level at most $t-1$. By Property 1(2), $M(a)$ is critical. Hence, the blocking pair (a, b) is justified by Condition 1 of Definition 2.

Case 2: $b \in \mathcal{B}_y$ **for** $y > t$. 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. □

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 matching.

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 the 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 $\mathsf{Pref}(a_1)$. Thus, a_1 is at level at least t^*. Since b_1 remains unmatched in M, a did not exhaust $\mathsf{Pref}(a)$ 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 not appear as a separate level in G_M. Thus, the vertices $a, a_1 \in \mathcal{A}_t$.

The Pair (a, b) Blocks M': If $a_1 \succ_b a$, then b would have accepted the proposal of a_1^t by rejecting a^t. Thus, $a \succeq_b a_1$. 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 a 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. We claim that b must reject the proposal by a after the proposal (a, b) becomes uncertain due to a proposal by some vertex, possibly a_1^t. Such a vertex must exist because a_1^t proposed to b after (a, b) becomes uncertain. 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 is a contradiction that $a_1 \in \mathcal{A}_t$. 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. □

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

5 Conclusion

In this work, we consider the problem of computing a matching in the stable marriage problem where ties and critical vertices can appear on both sides of the bipartition. We investigate a recently introduced notion of optimality called relaxed stability for our setting. We show that every instance of our problem admits a Relaxed Stable Matching (RSM) which is also critical. It follows from the known results [6,18] that computing a maximum size critical RSM is NP-hard and hard to approximate within any factor smaller than $\frac{21}{19}$. We present a polynomial-time algorithm to compute a $\frac{3}{2}$-approximation of the maximum size critical RSM.

References

1. Biró, P., Irving, R.W., Manlove, D.F.: Popular matchings in the marriage and roommates problems. In: Calamoneri, T., Diaz, J. (eds.) CIAC 2010. LNCS, vol. 6078, pp. 97–108. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13073-1_10
2. Dean, B., Jalasutram, 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. Am. Math. Mon. **69**(1), 9–15 (1962)
4. Gärdenfors, P.: Match making: assignments based on bilateral preferences. Behav. Sci. **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 Trans. Algorithm (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)
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 (2019)
9. Huang, C.C., Kavitha, T.: Improved approximation algorithms for two variants of the stable marriage problem with ties. Math. Program. **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)

14. Krishnaa, P., Limaye, G., Nasre, M., Nimbhorkar, P.: Envy-freeness and relaxed stability: hardness and approximation algorithms. In: Harks, T., Klimm, M. (eds.) SAGT 2020. LNCS, vol. 12283, pp. 193–208. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57980-7_13
15. Krishnaa, P., Limaye, G., Nasre, M., Nimbhorkar, P.: Envy-freeness and relaxed stability: hardness and approximation algorithms. J. Comb. Optim. **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. Makino, K., Miyazaki, S., Yokoi, Y.: Incomplete list setting of the hospitals/residents problem with maximally satisfying lower quotas. In: Kanellopoulos, P., Kyropoulou, M., Voudouris, A. (eds.) SAGT 2022. Lecture Notes in Computer Science, vol. 13584, pp. 544–561. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15714-1_31
18. Manlove, D.F., Irving, R.W., Iwama, K., Miyazaki, S., Morita, Y.: Hard variants of stable marriage. Theoret. Comput. Sci. **276**(1–2), 261–279 (2002)
19. McDermid, E.: A 3/2-approximation algorithm for general stable marriage. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 689–700. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02927-1_57
20. 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)
21. Nasre, M., Nimbhorkar, P., Ranjan, K.: Critical relaxed stable matchings with two-sided ties. arXiv preprint arXiv:2303.12325 (2023)
22. 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), vol. 213, pp. 30:1–30:21 (2021)
23. Nasre, M., Nimbhorkar, P., Ranjan, K., Sarkar, A.: Popular critical matchings in the many-to-many setting. arXiv:2206.12394 (2023)
24. Paluch, K.: Faster and simpler approximation of stable matchings. Algorithms **7**(2), 189–202 (2014)
25. Robards, P.A.: Applying two-sided matching processes to the united states navy enlisted assignment process. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA (2001)
26. Roth, A.E.: Stability and polarization of interests in job matching. Econometrica: J. Econometric Soc. **52**, 47–57 (1984)
27. Roth, A.E.: On the allocation of residents to rural hospitals: a general property of two-sided matching markets. Econometrica: J. Econometric Soc. **54**, 425–427 (1986)
28. Tan, S.J., Yeong, C.M.: Designing economics experiments to demonstrate the advantages of an electronic employment market in a large military organization. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA (2001)
29. Yang, W., Sycara, K.: Two-sided matching for the us navy detailing process with market complication. Technical report, Technical Report CMU-RI-TR-03-49, Robotics Institute, Carnegie-Mellon University (2003)
30. Yokoi, Y.: Envy-free matchings with lower quotas. Algorithmica **82**(2), 188–211 (2020)