Manipulation Strategies for the Rank-Maximal Matching Problem*

Pratik Ghosal and Katarzyna Paluch

University of Wroclaw

Abstract. We consider manipulation strategies for the rank-maximal matching problem. Let $G = (A \cup P, \mathcal{E})$ be a bipartite graph such that A denotes a set of applicants and P a set of posts. Each applicant $a \in A$ has a preference list over the set of his neighbours in G, possibly involving ties. A matching M is any subset of edges from $\mathcal{E}$ such that no two edges of M share an endpoint. A *rank-maximal* matching is one in which the maximum number of applicants is matched to their rank one posts, subject to this condition, the maximum number of applicants is matched to their rank two posts and so on. A central authority matches applicants to posts in G using one of rank-maximal matchings. Let a_1 be the sole manipulative applicant, who knows the preference lists of all the other applicants and wants to falsify his preference list, so that, he has a chance of getting better posts than if he were truthful, i.e., than if he gave a true preference list.

We give three manipulation strategies for a_1 in this paper. In the first problem 'best nonfirst', the manipulative applicant a_1 wants to ensure that he is never matched to any post worse than the most preferred post among those of rank greater than one and obtainable, when he is truthful. In the second strategy 'min max' the manipulator wants to construct a preference list for a_1 such that the worst post he can become matched to by the central authority is best possible or in other words, a_1 wants to minimize the maximal rank of a post he can become matched to. To be able to carry out strategy 'best nonfirst', a_1 only needs to know the most preferred post of each applicant, whereas putting into effect 'min max' requires the knowledge of whole preference lists of all applicants. The last manipulation strategy 'improve best' guarantees that a_1 is matched to his most preferred post at least in some rank-maximal matchings.

1 Introduction

We consider manipulation strategies for the rank-maximal matching problem. In the rank-maximal matching problem, we are given a bipartite graph $G = (A \cup P, \mathcal{E})$ where A denotes a set of applicants and P a set of posts. Each applicant $a \in A$ has a preference list over the set of his neighbours in G, possibly involving ties. Preference lists are represented by ranks on the edges - an edge (a, p) has

* Partly supported by Polish National Science Center grant UMO-2013/11/B/ST6/01748

rank i, denoted as $rank(a,p) = i$, if post p belongs to one of a's i-th choices. An applicant a prefers a post p to a post p' if $rank(a,p) < rank(a,p')$. In this case, we say that (a,p) has higher rank than (a,p'). If a is indifferent between p and p', then $rank(a,p) = rank(a,p')$. Posts most preferred by an applicant a have rank one in his preference list. A *matching* M is any subset of edges $\mathcal{E}$ such that no two edges of M share an endpoint. A matching is called a *rank-maximal* matching if it matches the maximum number of applicants to their rank one posts and subject to this condition, the maximum number of applicants to their rank two posts, and so on. A rank-maximal matching can be computed in $O(\min(c\sqrt{n}, n)m)$ time, where n is the number of applicants, m the number of edges and c the maximum rank of an edge in an optimal solution [20].

A central authority matches applicants to posts by using the rank-maximal matching algorithm. Since there may be more than one rank- maximal matching of G, we assume that the central authority may choose any one of them arbitrarily. Let a_1 be a manipulative applicant, who knows the preference lists of all the other applicants and wants to falsify his preference list, so that, he has a chance of getting better posts than if he were truthful, i.e., than if he gave a true preference list. We can always assume that a_1 does not get his most preferred post in every rank-maximal matching when he is truthful, otherwise, a_1 does not have any incentive to cheat. Also, we can notice that it is usually advantageous for a_1 to truncate his preference list. Let H_p denote the graph, in which a_1's preference list consists of only one post p. Then as long as no rank-maximal matching of H_p leaves a_1 unmatched, he is guaranteed to always get the post p. To cover the worst case situation for a_1, our strategies require a_1 to provide a full preference list that includes every post from P. Also, a_1 could make the posts, he does not want to be matched to, appear very far in his preference list. Thus, we assume that a_1 does not have any gap in his preference list, i.e., it cannot happen that in a_1's preference list there are a rank i and rank $(i+2)$ posts but none of rank $(i+1)$.

Our Contribution: Our contribution consists in developing manipulation strategies for the rank-maximal matching problem. Given a graph instance with the true preference list of every applicant, we introduce three manipulation strategies for a_1. We consider the case where a_1 is the sole manipulator in G.

Our first manipulation strategy named 'best nonfirst' is described in Section 3. The strategy may not provide an optimal improvement for a_1, but it is simple and fast. This strategy guarantees that a_1 is never matched to any post worse than the second best post he can be matched to in a rank-maximal matching, when he is truthful. In other words, if a_1 is matched to a post p when he is truthful and p is not his most preferred post, then the strategy 'best nonfirst' ensures that he is never matched to any post ranked worse than p in any rank-maximal matching. The advantage of this strategy is that a_1 does not need to know full preference lists of the other applicants. He only needs to know the most preferred post of each applicant to be able to successfully execute the strategy.

Next, in Section 5.2 we propose the strategy 'min max'. The strategy minimizes the maximal rank of a post a_1 can become matched to. Thus it optimally improves the worst post of a_1 that is obtainable from the central authority. What is more, the strategy has the property that by using it, a_1 always gets matched to p_1, which is the best among worst posts he can be matched to. Moreover, we prove that there does not exist a strategy that simultaneously guarantees that a_1 never gets a post worse than p_1 and sometimes gets a post better than p_1.

Last but not least, we have studied the manipulation strategy 'improve best' in Section 6. The previous two manipulation strategies improve the worst post a_1 can be matched to in a rank-maximal matching. Hence, these strategies may not match a_1 to his most preferred post in any rank maximal matching. In this manipulation strategy, a_1 has a different goal - he wants to be matched to his most preferred post in some rank-maximal matchings. Note that it is not possible for him to ensure that he always gets his most preferred post.

Previous and related work. The rank-maximal matching problem belongs to the class of matching problems with preferences. In the problems with one-sided preferences, the considered graph is bipartite and each vertex of only one set of the bipartition expresses preferences over the set of its neighbours. Apart from rank-maximal matchings, other types of matchings from this class include pareto-optimal [1] [28] [5], popular [3] and fair[15] matchings among others. In the problems with two-sided preferences, the underlying graph is also bipartite but vertices from both sides of the bipartition express preferences over their neighbours. The most famous example of a matching problem with two-sided preferences is that of a stable matching known also as the stable marriage problem. Since the seminal paper by Gale and Shapley [8], it has been studied very intensively, among others in [12],[18],[27]. In the non-bipartite matching problems with preferences each vertex from the graph ranks all of its neighbours. The stable roommate problem [17] is a counterpart of the stable marriage problem in the non-bipartite setting.

The rank-maximal matching problem was first introduced by Irving[19]. A rank-maximal matching can be found via a relatively straightforward reduction to the maximum weight matching problem. The already mentioned [20] gives a combinatorial algorithm that runs in $O(\min(n, c\sqrt{n})m)$ time. The capacitated and weighted versions were considered, respectively, in [25] and [21]. A switching graph characterization of the set of all rank-maximal matchings is described in [11]. Finally, the dynamic version of the rank-maximal matching problem was considered in [24] and [10].

A matching problem with preferences is called strategy-proof if it is in the best interest of each applicant to provide their true preference list. An example of a strategy-proof mechanism among matching problems with one-sided preferences is that of a pareto optimal matching. The strategyproofness of a pareto optimal matching has applications in house allocation [16] [2] [30] [22] and kidney exchange [29] [4]. Regarding the stable matching problem, if a stable matching algorithm produces a men-optimal stable matching, then it is not possible for men to gain any advantage by changing or contracting their preference lists and then the best strategy for them is to keep their true preference lists [7][26].

In the context of matching with preferences cheating strategies were mainly studied for the stable matching problem. Gale and Sotomayor [9] showed that women can shorten their preference lists to force an algorithm, that computes the men-optimal stable matching, to produce the women-optimal stable matching. Teo et al. [31] considered a cheating strategy, where women are required to give a full preference list and one of the women is a manipulator. Huang [13] explored the versions, in which, men can make coalitions. Manipulation strategy in the stable roommate problem was also considered by Huang[14]. For a matching problem with one-sided preferences, Nasre[23] studied manipulation strategies for the popular matching problem.

2 Background

A matching M is said to be *maximum (in a graph G)* if, among all matchings of G, it has the maximum number of edges. A path P is said to be *alternating with respect to matching M* or *M-alternating* if its edges belong alternately to M and $\mathcal{E} \setminus M$. A vertex v is *unmatched* or *free* in M if it is not incident to any edge of M. An M-alternating path P such that both its endpoints are unmatched in M, is said to be *M-augmenting* (or augmenting with respect to M). It was proved by Berge [6] that a matching M is maximum if and only if there exists no M-augmenting path.

We state the following well-known properties of maximum matchings in bipartite graphs. Let $G = (A \cup P, \mathcal{E})$ be a bipartite graph and let M be a maximum matching in G. The matching M defines a partition of the vertex set $A \cup P$ into three disjoint sets. A vertex $v \in A \cup P$ is even (resp. odd) if there is an even (resp. odd) length alternating path with respect to M from an unmatched vertex to v. A vertex v is unreachable if there is no alternating path from an unmatched vertex to v. The even, odd and unreachable vertices are denoted by E, O and U respectively. The following lemma is well known in matching theory. The proofs can be found in [20].

Lemma 1. *Let E, O and U be the sets of vertices defined as above by a maximum matching M in G. Then,*

1. *E, O and U are pairwise disjoint, and independent of the maximum matching M in G.*
2. *In any maximum matching of G, every vertex in O is matched with a vertex in E, and every vertex in U is matched with another vertex in U. The size of a maximum matching is $|O| + |U|/2$.*
3. *G contains no edge between a vertex in E and a vertex in $E \cup U$.*

2.1 Rank-Maximal Matchings

Next we review an algorithm by Irving et al. [20] for computing a rank-maximal matching. Let $G = (A \cup P, \mathcal{E})$ be an instance of the rank-maximal matching problem. Every edge $e = (a, p)$ has a rank reflecting its position in the preference list of applicant a. $\mathcal{E}$ is the union of disjoint sets $\mathcal{E}_i$, i.e., $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3 ... \cup \mathcal{E}_r$, where $\mathcal{E}_i$ denotes the set of edges of rank i and r denotes the lowest rank of an edge in G.

Definition 1. *The signature of a matching M is defined as an r-tuple $\rho(M) = (x_1, ..., x_r)$ where, for each $1 \leq i \leq r$, x_i is the number of applicants who are matched to their i-th rank post in M.*

Let M and M' be two matchings of G, with the signatures $sig(M) = (x_1, ..., x_r)$ and $sig(M') = (y_1, ..., y_r)$. We say $M \succ M'$ if there exists k such that $x_i = y_i$ for each $1 \leq i < k \leq r$ and $x_k > y_k$.

Definition 2. *A matching M of a graph G is called rank-maximal if and only if M has the best signature under the ordering $\succ$ defined above.*

We give a brief description of the algorithm of Irving et al. [20] for computing a rank-maximal matching, whose pseudocode (Algorithm 1) is given below. Let us denote $G_i = (A \cup P, \mathcal{E}_1 \cup \mathcal{E}_2 \cup ... \cup \mathcal{E}_i)$ as a subgraph of G that only contains edges of rank smaller or equal to i. The algorithm runs in phases. The algorithm starts with $G'_1 = G_1$ and a maximum matching M_1 of G_1. In the first phase, the set of vertices is partitioned into E_1, O_1 and U_1. The edges between O_1 and $O_1 \cup U_1$ are deleted. Since the vertices incident to $O_1 \cup U_1$ have to be matched in G_1 in every rank-maximal matching, the edges of rank greater than 1 incident to such vertices are deleted from the graph G. Next we add the edges of rank 2 and call the resulting graph G'_2. The graph G'_2 may contain some M_1-augmenting paths. We determine the maximum matching M_2 in G'_2 by augmenting M_1. In the i-th phase, the vertices are partitioned into three disjoint sets $E(G'_i)$, $O(G'_i)$ and $U(G'_i)$. We delete every edge between O_i and $O_i \cup U_i$. Also, we delete every edge of rank greater than i incident to vertices in $O_i \cup U_i$. Next we add the edges of rank $(i + 1)$ and call the resulting graph G'_{i+1}. We determine the maximum matching M_{i+1} in G'_{i+1} by augmenting M_i. G' is also called the reduced graph of G.

Algorithm 1 for computing a rank-maximal matching

procedure RankMaximalMatching(G)
  $G'_1 \leftarrow G_1$
  Let M_1 be any maximum matching of G'_1
  for $i = 1, 2, \ldots, r$ **do**
    Partition the vertices of G'_i into the sets $E(G'_i)$, $O(G'_i)$ and $U(G'_i)$
    Delete all edges in $\mathcal{E}_j$ (for $j > i$) which are incident on vertices in $O(G'_i) \cup U(G'_i)$
    Delete all $O(G'_i)O(G'_i)$ and $O(G'_i)U(G'_i)$ edges from G'_i.
    Add the edges in $\mathcal{E}_{i+1}$ and denote the graph as G'_{i+1}.
    Determine a maximum matching M_{i+1} in G'_{i+1} by augmenting M_i.
  return M_r

The following invariants are proved in [20].

1. For every $1 \leq i \leq r$, every rank-maximal matching in G_i is contained in G'_i.
2. The matching M_i is rank-maximal in G_i, and is a maximum matching in G'_i.
3. If a rank-maximal matching in G has signature $(s_1, ..., s_i, ...s_r)$, then M_i has signature $(s_1, ..., s_i)$.
4. The graphs G'_i, $(1 \leq i \leq r)$ are independent of the rank-maximal matching computed by the algorithm.

Lemma 2. *Let $G = (A \cup P, \mathcal{E})$ and $G' = (A \cup P, \mathcal{E}')$ be two bipartite graphs with ranks on the edges. Suppose that $\mathcal{E}' \subseteq \mathcal{E}$. Also, every edge $e \in \mathcal{E}'$ has the same rank in G and G'. Then any rank-maximal matching M of G such that $M \subseteq \mathcal{E}'$ is also a rank-maximal matching of G'.*

Proof. Let M be a rank-maximal matching of G. Moreover, we assume that M is a matching of G'. Since each edge of $\mathcal{E}'$ has the same rank in both G and G', the signature of M is the same in both graphs. Therefore the signature of a rank-maximal matching of G' is not worse than the signature of a rank-maximal matching of G. Suppose the signature of a rank-maximal matching of G' is strictly better than the signature of a rank-maximal matching of G. If M' is a rank-maximal matching of G', by the construction of G and G', M' is a matching of G and M' has the same signature in both G and G'. But $M \succ M'$. Thus M is not a rank-maximal matching of G which is a contradiction. □

3 Properties of a preference list and strategy 'best nonfirst'

Here we note down some properties of the preference list of any applicant. Let us assume that the preference list of a_1 in G has the form $(P_1, \ \ P_2, \ \ P_3, \ldots, P_i, \ldots, P_t)$, where P_i denotes the set of posts of rank i in the preference list of a_1. $G \setminus \{a_1\}$ denotes the graph obtained from G after the removal of the vertex a_1 from G. We define an f-post of G in a similar way as in the popular matching problem [3]

Definition 3. *A post is called an f-post of G if and only if it belongs to $O(G_1 \setminus \{a_1\})$ or $U(G_1 \setminus \{a_1\})$, where $G_1 = (A \cup P, \mathcal{E}_1)$. The remaining posts of G are called non-f-posts.*

Lemma 3. *If P_1 contains a post that is a non-f-post, then a_1 is always matched to one of such posts in a rank-maximal matching of G and thus to one of his first choices.*

Proof. Let M' be a maximum matching of $G_1 \setminus \{a_1\}$. M' is a matching of G_1 but not necessarily of maximum size. a_1 is unmatched in M' and P_1 contains a post p that is not an f post. Hence, p belongs to $E(G_1 \setminus \{a_1\})$ and there exists an even length M'-alternating path S starting at p and ending at some unmatched vertex p' in $G_1 \setminus \{a_1\}$. Therefore, S together with the edge (a_1, p)

forms an augmenting path in the graph G_1. If we apply any such augmenting path we obtain a maximum matching of G_1 and a_1 is matched in every maximum matching of G_1. Let us also notice that no edge (a_1, p_1) such that p_1 is an f-post from P_1 belongs to an M'-augmenting path. This shows that a_1 is matched in every maximum matching of G_1 and to a non-f-post from P_1. This completes our proof. $\square$

Next lemma shows that if a_1 is not matched to a rank one post in some rank-maximal matching of G, then an f-post may be defined in an alternative way that takes into account the whole graph G_1. This property is needed during the construction of strategy 'min max'.

Lemma 4. *Let us assume that a_1 is not matched to a rank one post in some rank-maximal matching of G. Then a post is an f-post if and only if it belongs to $O(G_1)$ or $U(G_1)$.*

Proof. We say that a vertex v has the same type in graphs G and H if $v \in (E(G) \cap E(H)) \cup (O(G) \cap O(H)) \cup (U(G) \cap U(H))$.

We have assumed that a_1 is not matched to his rank one post in every rank-maximal matching. By the properties of the rank-maximal matchings, $a_1 \in E(G_1)$. Hence, we can find a maximum matching M of G_1 in which a_1 is an unmatched vertex. Notice that M is a maximum matching of both G_1 and $G_1 \setminus \{a_1\}$. Hence, a vertex that is reachable from a free vertex other than a_1 in G_1, has the same type in both G_1 and $G_1 \setminus \{a_1\}$. The vertices from P that are reachable only from a_1 by an alternating path in G_1, belong to $O(G_1)$. These vertices become unreachable in $G_1 \setminus \{a_1\}$. Finally an unreachable vertex in G_1 is also an unreachable vertex in $G_1 \setminus \{a_1\}$. Therefore, a post that belongs to $O(G_1)$ or $U(G_1)$, is an f-post.

Conversely, let us consider an f-post p. p is either an odd or an unreachable vertex in $G_1 \setminus \{a_1\}$. Suppose p becomes an even vertex in G_1. We have proved in the previous part that p must be reachable from vertices other than a_1. Also we know that a vertex that is reachable from a free vertex other than a_1 in G_1, has the same type in both G_1 and $G_1 \setminus \{a_1\}$. Hence, p is an even vertex in $G_1 \setminus \{a_1\}$, which is a contradiction. Therefore, p is either an odd or an unreachable vertex in G_1. $\square$

The lemma below characterises the set of potential posts a_1 can be matched to, if he provides his true preference list.

Lemma 5. *Let G be a bipartite graph and i be the rank of the highest ranked non-f-post in the preference list of a_1. If a_1 is not matched to a rank one post, then a_1 can only be matched to a post of rank i or greater than i in any rank-maximal matching of G.*

Proof. Let M be a rank-maximal matching of G. While computing a rank-maximal matching of G we start by finding a maximum matching of G_1. Since a_1 is not matched to a rank one post in every rank-maximal matching of G, by Lemma 4, the set of f-posts contains every vertex from $O(G_1)$ and $U(G_1)$.

Hence, we delete every edge, that has rank bigger than 1, incident to an f-post. Thus, every edge $e = (a_1, p)$ such that p is an f-post and $rank(a, p) > 1$ gets deleted after the first iteration of the algorithm. Therefore, no such edge can belong to a rank-maximal matching and a_1 can only be matched to a post of rank i or worse. □

The above lemmas provide us with an easy method of manipulation that guarantee that a_1 can always be matched to the best non-f-post in his true preference list. Lemma 5 shows that the most preferred non-f-post of a_1 is ranked not worse than the second most preferred post he can be matched to, when he is truthful. We assume that a_1 is not matched to a rank one post in every rank-maximal matching of G. Otherwise, the manipulator has no incentive to cheat. Let $p_i \in P_i$ be a highest ranked non-f post in the true preference list of a_1. We put p_i as a rank 1 post in the falsified preference list of a_1. Next, we fill the falsified preference list of a_1 arbitrarily. This completes the description of strategy ‘best nonfirst’.

Algorithm 2 Strategy ‘best nonfirst’

```
p_i ← a highest ranked non-f-post in the true preference list of a_1.
p_i ← the rank one post in the falsified preference list of a_1 in H
Fill the rest of the preference list of a_1 in an arbitrary order
Output H
```

Theorem 1. *The graph H computed by Algorithm 2 is a strategy ‘best nonfirst’.*

The correctness of Algorithm 2 follows from Lemma 3.

4 Example of Strategy ‘best non-first’ Not Being Optimal

We have given a strategy ‘best nonfirst’ in the previous part of the paper. But that strategy may not be optimal in the sense that the manipulator may arrange to get a post of even better rank. Let us consider an example from Figure 1. Let us assume that a_1 is a manipulator. The first preference table contains true preference list of all applicants $a_1, \ldots, a_6$. a_1 is matched to p_5 in every rank-maximal matching of this instance. p_3 is the best non-f-post in the preference list of a_1. The second preference table contains a falsified preference list of a_1. Here a_1 adopted the strategy ‘best nonfirst’ and as a result he is matched to p_3 in every rank-maximal matching. We can fill the rest of the preference list of a_1 arbitrarily. Consider now the third preference table, in which a_1 falsifies his preference list in yet another way. By presenting this falsified preference list a_1 contrives to get matched to p_2 in every rank-maximal matching. Post p_2 is also his true first choice. Hence getting matched to p_3 is not an optimal strategy for a_1.

a_1	p_2	p_1	p_3	$\underline{p_5}$	p_4	
a_2	$\underline{p_1}$	p_2	p_3	p_4	p_5	
a_3	p_1	p_2	p_3	$\underline{p_4}$	p_5	
a_4	p_1	p_2	$\underline{p_3}$	p_4	p_5	
a_5	$\underline{p_2}$	p_1	p_3	p_6	p_4	p_5
a_6	$\underline{p_6}$					

a_1	$\underline{p_3}$					
a_2	$\underline{p_1}$	p_2	p_3	p_4	p_5	
a_3	p_1	p_2	p_3	$\underline{p_4}$	p_5	
a_4	p_1	p_2	p_3	p_4	$\underline{p_5}$	
a_5	$\underline{p_2}$	p_1	p_3	p_6	p_4	p_5
a_6	$\underline{p_6}$					

a_1	$\underline{p_2}$	p_1	p_6	p_3	p_4	p_5
a_2	$\underline{p_1}$	p_2	p_3	p_4	p_5	
a_3	p_1	p_2	p_3	p_4	$\underline{p_5}$	
a_4	p_1	p_2	p_3	$\underline{p_4}$	p_5	
a_5	p_2	p_1	$\underline{p_3}$	p_6	p_4	p_5
a_6	$\underline{p_6}$					

Fig. 1. Example that shows that strategy 'best non-first' may not be optimal. The underlined posts are those matched to the corresponding applicant. We can fill the rest of the positions arbitrarily because the presence of those posts will not affect the form of any rank-maximal matching

5 Strategy 'min max'

The example in the previous section clearly shws that the strategy 'best nonfirst' may not provide an optimal solution. In this section we introduce the strategy 'min max' that optimizes the worst post a_1 can be matched to in a rank-maximal matching.

5.1 Critical Rank

The notion that is going to be very useful while constructing a preference list is that of a critical rank.

Definition 4. *Let $G = (A \cup P, \mathcal{E})$ be a bipartite graph with ranks on the edges belonging to $\{1, 2, \ldots, r\}$. Suppose that $a \in A, p \in P$ and (a, p) does not belong to $\mathcal{E}$. Let $H = (A \cup P, \mathcal{E} \cup \{(a, p)\})$. We define a critical rank of (a, p) in H as follows.*

If there exists a natural number $1 \leq i \leq r$ such that $a \in O(G'_i) \cup U(G'_i)$ or $p \in O(G'_i) \cup U(G'_i)$, then the critical rank of (a, p) in H is equal to $\min\{i : (O(G'_i) \cup U(G'_i)) \cap \{a, p\} \neq \emptyset\}$. Otherwise, the critical rank of (a, p) is defined as $r + 1$.

The next lemma reveals an interesting property of the critical rank of an edge (a, p).

Lemma 6. *Let G, H and (a,p) be as in Definition 4. Then the critical rank of (a,p) is c if and only if*

1. *for every $1 \le i < c$, the edge (a,p) belongs to every rank-maximal matching of H_i, in which (a,p) has rank i, and*
2. *for every $c < i \le r$, the edge (a,p) does not belong to any rank-maximal matching of H_i, in which (a,p) has rank i, and*
3. *there exists a rank-maximal matching M of H_c, in which (a,p) has rank c such that (a,p) is not contained in M.*

Proof. We start by proving the following claim.

Claim. Suppose that $e = (a,p)$ has rank i in the graph H. Then (a,p) belongs to every rank-maximal matching of H_i if and only if for every $j \le i$ both $a \in E(G'_j)$ and $p \in E(G'_j)$.

Since (a,p) has rank i in H and G differs from H only by the existence of the edge (a,p), we know that the reduced graphs G'_j and H'_j are the same for each $j < i$. Also, the reduced graphs H'_i and G'_i may be the same or differ by the existence of the edge e.

If for some $j < i$ it holds that $a \in O(G'_j) \cup U(G'_j)$ or $p \in O(G'_j) \cup U(G'_j)$, then the edge (a,p) does not belong to H'_i, because it is removed at the beginning of phase $j+1$ during the computation of a rank-maximal matching of H. Thus in this case e does not belong to any rank-maximal matching of H_i. If for every $j < i$, $a \in E(G'_j)$ and $p \in E(G'_j)$ and $a \in O(G'_i) \cup U(G'_i)$ or $p \in O(G'_i) \cup U(G'_i)$, then e belongs to H'_i, but there exists a rank-maximal matching of H_i that does not contain e. This is so because the addition of e to G'_i does not create any augmenting path in H'_i, therefore a maximum matching of G'_i is also a maximum matching of H'_i and thus a rank-maximal matching of H_i. This shows that in this case every rank-maximal matching of G_i is also rank-maximal in H_i. Hence, there exists a rank-maximal matching of H_i that does not contain (a,p).

Assume now that for every $j \le i$ both $a \in E(G'_j)$ and $p \in E(G'_j)$. This means that the reduced graph H'_i does contain (a,p). Any rank-maximal matching of H_i is a maximum matching of H'_i. Let M_i denote a maximum matching of G'_i. Since $a \in E(G'_i)$ and $p \in E(G'_i)$, (a,p) belongs to every M_i-augmenting path - because a is the endpoint of some even length M_i-alternating path ending at a free vertex $a' \in A$ and similarly, p is the endpoint of some even length M_i-alternating path ending at a free vertex $p' \in P$. Together with (a,p) these paths form an M_i-augmenting path in H'_i. Additionally, we notice that (a,p) belongs to every M_i-augmenting path and thus to every maximum matching of H'_i and hence to every rank-maximal matching of H_i.

Let us now prove the other direction of the claim and suppose that (a,p) belongs to every rank-maximal matching of H_i. This means that (a,p) belongs to every maximum matching of H'_i. Then (a,p) must be contained in H'_i and by the above arguments, we know that for every $j < i$ both $a \in E(G'_j)$ and $p \in E(G'_j)$. No maximum matching of G'_i contains (a,p) - therefore a maximum matching of H'_i must be bigger by one than a maximum matching of G'_i. This means that (a,p) must belong to a path augmenting with respect to a maximum matching of G'_i, which means that the endpoints of (a,p) belong to $E(G'_i)$. This way we have proved the claim.

The claim, which we have just proved, shows that if (i) for every $1 \leq i < c$ the edge (a,p) belongs to every rank-maximal matching of H_i, in which (a,p) has rank i and (ii) there exists a rank-maximal matching M of H_c, in which (a,p) has rank c such that (a,p) is not contained in M, then $c = \min\{i : (O(G'_i) \cup U(G'_i)) \cap \{a,p\} \neq \emptyset\}$ and hence, c is the critical rank of (a,p).

The claim also shows that if c is the critical rank of (a,p), then for every $i < c$ the edge (a,p) belongs to every rank-maximal matching of H_i, in which (a,p) has rank i and there exists a rank-maximal matching of H_c, in which (a,p) has rank c that does not contain (a,p). It remains to prove that if c is the critical rank of (a,p), then for every $c < i \leq r$ the edge (a,p) belongs to no rank-maximal matching of H_i, in which (a,p) has rank i. This follows from the fact that any edge of rank $i > c$ incident to a vertex belonging to $O(H'_c) \cup U(H'_c)$ is removed from H'_i and therefore cannot belong to a maximum matching of H'_i and thus cannot be present in any rank-maximal matching of H_i. This ends the proof of the lemma. □

Corollary 1. *The critical rank of an edge incident to an f-post of G is* 1 *in G.*

The next two lemmas explain the change of the critical rank of (a,p) when we add an f-post p' as a rank 1 post to the preference list of a.

Lemma 7. *Let G be a bipartite graph and a be an applicant. Let p be the only post in the preference list of a. Suppose that the critical rank of (a,p) is c in G. Let $\hat{G} = G \cup \{(a,p')\}$ where p' is a rank* 1*, f-post in the preference list of a. Then the critical rank of (a,p) is at most c in $\hat{G}$.*

Proof. It suffices to show that (a,p) is not matched in every rank-maximal matching of $\hat{G}_c$ with (a,p) having rank c and for every $c < i \leq r$ no rank-maximal matching of $\hat{G}$, in which (a,p) has rank i contains (a,p).

Suppose, (a,p) is matched in every rank-maximal matching of $\hat{G}_c$. If M is a matching of $\hat{G}_c$, by Lemma 2, M is also a rank-maximal matching of G. Since the critical rank of (a,p) is c in G, there exists a rank-maximal matching M' of G_c that does not contain the edge (a,p). The signature of M and M' is the same. Hence M' is a rank-maximal matching of $\hat{G}_c$, which is a contradiction.

To prove the second part, suppose to the contrary that there exists rank-maximal matching M of $\hat{G}$, in which (a,p) has rank i that contains (a,p). Then M is also a matching of G and by Lemma 2 it is also a rank-maximal matching of G. However, by Lemma 6 no rank-maximal matching of G, in which (a,p) has rank $i > c$ can contain (a,p) - a contradiction. □

Lemma 8. *Let $G = (A \cup P, \mathcal{E})$ be a bipartite graph, in which a has two neighbors p and p' such that apart from (a,p), each edge has a rank belonging to $\{1, 2, \ldots, r\}$. Additionally, p' is a rank one f-post in the preference list of a. Let $G' = (A \cup P, \mathcal{E} \setminus \{(a,p)\})$ and $G'' = (A \cup P, \mathcal{E} \setminus \{(a,p')\})$. Suppose that a becomes* *unreachable in* G'_i *and the critical rank of* (a, p) *is* c *in* G''. *Then the critical rank of* (a, p) *in the graph* G *is equal to, correspondingly:*

1. c *if* $c \leq i$,
2. i *if* $c > i$.

Proof. We can prove that for every $j < i$ there exists a rank-maximal matching of G'_j that contains (a, p') and there exists a rank-maximal matching of G'_j that does not contain (a, p').

Claim. Let us suppose that (a, p) has rank $c' < i$ in G. Then, the existence of a rank-maximal matching M of G that does not contain (a, p) implies that the critical rank of (a, p) in G'' is at most c'.

Proof. By Lemma 2, M is a rank-maximal matching of G' and hence, every rank-maximal matching of G' is also rank-maximal in G. We know that there exists a rank-maximal matching M' of G' that does not contain (a, p'). Thus a is unmatched in M'. By Fact 2, M' is also a rank-maximal matching of G''. M' does not contain (a, p), which shows that the critical rank of (a, p) is at most c' in G''. □

First we assume that $c \leq i$. Since G'' is a subgraph of G, by Lemma 7, the critical rank of $(a, p) \leq c$ in G. Suppose the critical rank of $(a, p) = c' < c$ in G. Let us consider a graph G, in which (a, p) has rank c'. Since the critical rank of (a, p) is equal to c', there exists a rank-maximal matching M of G that does not contain (a, p). By the above claim, the critical rank of (a, p) in G'' is at most $c' < c$ - a contradiction. We conclude that the critical rank of (a, p) remains c in G if $c \leq i$.

Let us consider now the case when $c > i$. First we show that the critical rank of $(a, p) \leq i$ in G. Let us consider the graph G'. The vertex a becomes unreachable in G' after iteration i. We know that $G = G' \cup \{(a, p)\}$. Hence, the edge (a, p) is deleted in the graph G if the rank of $(a, p) > i$ in G. Therefore, the critical rank of $(a, p) \leq i$ in G.

Next we show that the critical rank of $(a, p) = i$ in G. Suppose the critical rank of (a, p) equals $i' < i$ in G. Since the critical rank of (a, p) is equal to $i' < i$, there exists a rank-maximal matching M of G, in which (a, p) has rank i' that does not contain (a, p). Again by the claim, the critical rank of (a, p) in G'' is at most $i' < c$ - a contradiction. Therefore we have proved that the critical rank of (a, p) is equal to i in G. □

Corollary 2. *Let* G *be a bipartite graph such that* p' *is a rank one,* f*-post in the preference list of* a. *Suppose that* a *becomes unreachable after iteration* i. *Let* $\hat{G} = G \cup \{(a, p)\}$ *with the rank of the edge* (a, p) *being* c. *Then* (a, p) *is never matched in a rank- maximal matching of* $\hat{G}$, *if* $c > i$.

The next lemma is useful while building a falsified preference list of a using the strategy 'min max'. This lemma basically combines two short preference lists of a into a longer preference list.

Lemma 9. *Let us consider two bipartite graphs G_1 and G_2 such that p is a rank one, f-post in the preference list of a in both graphs. Also a has only two neighbors in each of the graphs. In G_1, a has p_1 as a rank i post. In G_2, a has p_2 as a rank j post. We assume that G_3 is the union of graphs G_1 and G_2. Then a is matched to p in every rank-maximal matching of both G_1 and G_2 if and only if a is matched to p in every rank-maximal matching of G_3.*

Proof. Assume that a is matched to p_1 in a rank-maximal matching M of G_3. M is also a matching in the graph G_1. Since G_1 is a subgraph of G_3, from the fact 2, M is a rank-maximal matching of G_1. This means that a is matched to p_1 in some rank-maximal matchings of G_1, which is a contradiction.

Conversely, let a be matched to p in every rank-maximal matching of G_3. Let M_3 be a rank-maximal matching of G_3. Without loss of generality, suppose that a is matched to p_1 in a rank-maximal matching M_1 of G_1. Since G_1 is a subgraph of G_3, from fact 2, M_3 is a rank-maximal matching of G_1. Thus, M_1 and M_3 have the same signature. Therefore, M_1 is a rank-maximal matching of G_3, which is a contradiction. □

5.2 Algorithm for Strategy 'min max'

In this section, we give an algorithm that computes a graph H by using the strategy 'min max' for the applicant a_1. We recall that strategy 'min max' consists in finding a full preference list for a_1 such that the maximal rank of a post he can obtain is minimized. Since we have assumed that a_1 is not always matched to his first choice when he is truthful and since strategy 'best nonfirst' ensures that a_1 always gets the highest ranked non-f-post, it remains to check if it is possible for a_1 to get one of the f-posts in every rank-maximal matching. For a given f-post p we want to verify if a_1 can construct a full preference list that guarantees that a becomes matched to p in every rank-maximal matching of the resultant graph. From all such f-posts, we want to choose that of the highest rank in the true preference list of a_1. Below we show that this way we indeed compute the strategy 'min max'.

Let p be an f-post that a_1 wants to be matched to in every rank-maximal matching of H_p, where H_p contains a full falsified preference list of a_1. How do we construct such H_p? Let $\hat{H}_p$ denote the graph, in which a_1 is incident only to p and (a_1, p) has rank one. By Lemma 9, we know that in order to obtain H_p, it suffices to find a certain number of graphs H_{p,p_j} such that p and p_j are the only posts in the preference list of a_1, p has rank 1, p_j has rank $j > 1$ and every rank-maximal matching of H_{p,p_j} matches a_1 to p. Then we can combine those graphs into one graph H_p. In fact, it suffices to fill the preference list of a_1 only till rank k, where k is the rank, when a_1 becomes an unreachable vertex in $\hat{H}'_{p,i}$. This follows from Corollary 2, which says that no rank-maximal matching of $H_{p,p'}$ such that (a_1, p') has rank $i > k$ contains (a_1, p'). Therefore, the ranks greater than k in the preference list of a_1 may be filled with arbitrary posts not occurring previously.

Suppose that we want to find a "good" post for rank $i < k$ in the preference list of a_1. First, we check if there is any available post p' such that the critical rank of (a_1, p') is smaller than i in $H_{p,p'}$. If we find such a post, then by Lemma 6, the edge (a_1, p') never occurs in a rank-maximal matching of $H_{p,p'}$ in which (a_1, p') has rank i. Therefore, we may add p' to the preference list of a_1 as a rank i post. Otherwise, we consider a post p'' with critical rank i in the graph $H_{p,p''}$. We verify if p is matched to a_1 in every rank-maximal matching, when we add (a_1, p'') as a rank i edge to the graph $\hat{H}_p$. If yes, then we put p'' as an ith choice in H_p. If not, we check another post with critical rank i. If we are unable to find any post for rank i, Algorithm 3 outputs that there does not exist any preference list that matches a_1 to p in every rank-maximal matching of H_p.

The algorithm that computes a graph H_p, if it exists, is given below as Algorithm 3. The thing that still requires explanation is how we verify if (a, p) belongs to every rank-maximal matching of $H_{p,p'}$. For this, we need the reduced graph of $H_{p,p'}$ from phase r, which we can obtain by either applying the standard rank-maximal matching algorithm [20] or we can use one of the dynamic algorithms[10][24] if we want to have a faster algorithm. Once we have access to this reduced graph of $H_{p,p'}$ we can use the following lemma.

Lemma 10. *Let G be an instance of the rank-maximal matching problem, in which the maximal rank of an edge is r. Also, we assume that M is a fixed rank-maximal matching of G that matches an edge (a, p). Let us consider the switching graph of the matching M in G. Then the edge (a, p) belongs to every rank-maximal matching of G if there does not exist any switching path or switching cycle in the switching graph of M that contains the vertex p.*

Proof. Let us fix a rank-maximal matching M of G that matches the edge (a, p). Theorem 1 from [11] states that every rank-maximal matching G can be obtained from M by applying some vertex-disjoint switching paths and switching cycles in the switching graph of M. If there does not exist any switching path or switching cycle containing the vertex p, p has the same partner in every rank-maximal matching of G. Therefore, (a, p) is matched in every rank-maximal matching of G. $\square$

Definition 5. *We say that an f-post p is* feasible *if there exists a graph H_p such that every rank-maximal matching of H_p matches a_1 to p.*

In the lemma below we prove the correctness of Algorithm 3.

Lemma 11. *If Algorithm 3 outputs a graph H_p, then every rank-maximal matching of H_p matches a_1 to p. Otherwise, there does not exist a graph H_p, in which a_1 gives a full preference list such that every rank-maximal matching of H_p matches a_1 to p.*

Proof. If Algorithm 3 outputs a graph H_p, then by Corollary 2 and Lemma 9 every rank-maximal matching of H_p matches a_1 to p. If the algorithm does not output any graph, then it means that there was a problem for some i with finding

Algorithm 3 Construction of H_p

$C_i \leftarrow \{p' \in P$: the critical rank of (a_1, p') in $H_{p,p'}$ equals $i\}$
$L \leftarrow$ an empty list - L is the falsified preference list of a_1 that is going to have the form $(p_1, p_2, ..., p_n)$, where p_i denotes the rank i post in L.
add p to L – this is the rank 1 post in the preference list L of a_1
$k \leftarrow$ the number of phase when a_1 becomes unreachable in $\hat{H}_p$, i.e., $a_1 \in U(\hat{H}'_{p,k})$ and $a_1 \in E(\hat{H}'_{p,i})$ for every $i < k$, where $\hat{H}'_{p,i}$ is the i-th reduced graph of $\hat{H}_p$.
$C \leftarrow C_1$
for $i = 2, \ldots, k$ **do**
  if $C \neq \emptyset$ **then** (there exists a post p' in C)
    add p' as a rank i post to the falsified preference list L of a_1
    $C \leftarrow C \setminus \{p'\}$
  else $(C = \emptyset)$
    $SEARCH \leftarrow TRUE$
    while $\exists p'$ with critical rank of (a, p') equal to i in $\hat{H}_p$ and $SEARCH$ **do**
      if (a, p) belongs to every rank-maximal matching of $H_{p,p'}$ (Lemma 10) **then**
        add p' as a rank i post to the falsified preference list L of a_1
        $SEARCH \leftarrow FALSE$
  if SEARCH **then** Break
  $C \leftarrow C \cup C_i$
if L is a full preference list **then**
return H_p
else
return p is not a feasible f-post

a rank i post for the preference list of a_1. However, the algorithm considers every available post p' of critical rank at most i in $H_{p,p'}$ for that position. Therefore, if none of them has the required property that every rank-maximal matching of $H_{p,p'}$ matches a_1 to p, then no post of critical rank greater than i in $H_{p,p'}$ satisfies it either. This finishes the proof of correctness of Algorithm 3. □

Theorem 2. *Let p be the highest ranked feasible f-post in the true preference list of a_1. Then H_p output by Algorithm 3 is a strategy 'min max'. Moreover, each graph H that is a strategy 'min max' has the property that each rank-maximal matching of H matches a_1 to p.*

Proof. Let H_{opt} denote a graph that is a strategy 'min max'. Suppose that H_p is not a strategy 'min max'. There exists then a post p' such that $rank(a_1, p') < rank(a_1, p)$ in the true preference list of a_1 and (a_1, p') belongs to some rank-maximal matching of H_{opt}. Since $rank(a_1, p') < rank(a_1, p)$ in the true preference list of a_1, p' is an f-post. Also, a_1 can only be matched to a post of rank not worse than $rank(a_1, p)$ in the true preference list of a_1 in a rank-maximal matching of H_{opt}, otherwise H_{opt} would not be a strategy 'min max' because it would fare worse than the strategy H_p in terms of the worst post a_1 can become matched to. This means that a_1 can only be matched to f-posts under strategy H_{opt}, because every non-f-post has a worse rank than the rank of post p in the preference list of a_1. Corollary 1 shows that the critical rank of an f-post is 1. Hence, a_1 can only be matched to a post that has rank 1 in H_{opt}. Therefore, we can only put p' as a rank 1 post in the preference list of a_1 in H_{opt}.

Now we will show how to build a preference list of a_1 without any ties that matches a_1 to p' in every rank-maximal matching. Let us denote this graph as H_{mod}. We put p' as a rank 1 post. We create the preference list of a_1 in H_{mod} using the preference list of a_1 in H_{opt}. Suppose we want to add a rank i post to the preference list of a_1 in H_{mod}. If there is only one rank i post in the preference list of a_1 in H_{opt}, then that post will have the same rank in the preference list of a_1 in H_{mod}. If there is more than one rank i posts in the preference list of a_1 in H_{opt}, we choose only one of them. Finally, we add the remaining posts to a_1's preference list in H_{mod} in an arbitrary order as his least preferred posts. This way we get a preference list without any ties.

First, we show that H_{mod} and H_{opt} have the same signature. A rank-maximal matching in H_{opt} that matches a to p' is a matching in H_{mod} and has the same signature. Thus a rank-maximal matching of H_{opt} does not have a better signature than a rank- maximal matching of H_{mod}. Also, any post in the preference list of a_1 has rank in H_{opt} that is not worse than in H_{mod}. If we consider a rank-maximal matching of H_{mod}, then the signature of that matching is not worse in H_{opt}. Therefore rank-maximal matchings of H_{opt} and H_{mod} have the same signature.

Next we want to prove that a_1 is matched to p' in every rank-maximal matching of H_{mod}. Suppose that a_1 is matched to some p'' in a rank-maximal matching M of H_{mod}. We can see that M is a matching in H_{opt}. The rank of p'' in the preference list of a_1 in H_{opt} is not worse than the rank of p'' in the preference list of a_1 in H_{mod}. This shows that the signature of the matching M in H_{opt} is not worse than the signature of M in H_{mod}. But we know that the signatures of rank-maximal matching of H_{opt} and H_{mod} are the same. Hence, M is a rank-maximal matching of H_{opt}. This implies that a_1 is matched to p'' in a rank-maximal matching of H_{opt} too.

The preference list of a_1 in H_{mod} does not contain any ties, which means that p'' is not a rank 1 post in the preference list of a_1 in H_{mod}. We know that the critical rank of an f-post is 1. Therefore p'' is not an f-post. But a_1 cannot be matched to a non-f-post in H_{opt} - a contradiction. This proves that a_1 is matched to p' in every rank-maximal matching of H_{mod}. Lemma 9 shows that Algorithm 3 could also create the same preference list because it would consider each graph H_{p,p_i} such that p_i has rank i in H_{mod}. Since the rank of (a_1, p') is strictly better than the rank of (a_1, p) in the true preference list of a_1, it means that p is not a highest rank feasible f-post - a contradiction. Therefore H_p is indeed a strategy 'min max'. Also, by the construction of H_{mod} we have shown that each graph H that is a strategy 'min max' has the property that each rank-maximal matching of H matches a_1 to the same post. $\square$

6 'Improve Best' Strategy

In the previous section, we present an optimal algorithm for the manipulation strategy 'min max'. Also, we have shown that 'min max' strategy matches a_1 to the same post in every rank-maximal matching. There is a possibility that a_1 may not be matched to his most preferred post in any rank-maximal matching by using 'min max' strategy. Therefore, a natural question is whether there is a manipulation strategy that matches a_1 to his most preferred post in at least one rank-maximal matching. Here we present a strategy that matches a_1 to his most preferred post in some rank-maximal matchings.

Lemma 12. *Let G be a bipartite graph and a, a' be two applicants with identical preference list. If (a, p) is matched in a rank-maximal matching of G then the edge (a', p) is present in the reduced graph of G.*

Proof. Let (a, p) is matched in G. If we swap the partners of a and a', we get another matching in G. Since the preference list of a and a' are identical, the signature of both matchings is the same. Hence, there exists a rank-maximal matching that matches the edge (a', p). Thus (a', p) is present in the reduced graph of G. $\square$

6.1 Strategy

Here we give a brief description of the strategy 'improve best' for a_1. We assume that p_1 is the most preferred post in the preference list of a_1. Given a bipartite graph containing the true preference list of every applicant and a rank-maximal matching, we apply the decremental dynamic rank-maximal matching algorithm[10] to delete a_1 from the graph. Let us denote the updated graph as

G. First, we check which applicant is matched to p_1 in G. Suppose p_1 is matched to a'_1 in G. a_1 copies the preference list of a'_1 and present it as his falsified preference list. Let H be the graph with the falsified preference list of a_1. We claim that (a_1, p_1) is a rank-maximal pair in H when he uses this strategy. In other words, a_1 is matched to p_1 in some rank-maximal matchings of H.

6.2 Correctness

We assume that there is a rank-maximal matching of G that matches the edge (a'_1, p_1). We apply the incremental rank-maximal matching algorithm to get a rank-maximal matching of H. If we consider the vertex a'_1 in H, there are two possibilities. In the first case, let a'_1 be matched to p_1 in H. Without loss of generality, we assume that a_1 is matched to p'_1 in H. Now consider another matching of H by swapping the partners of a_1 and a'_1. Since a_1 and a'_1 has the identical preference list, both matchings have the same signature. Hence (a_1, p_1) is a rank-maximal pair of H. In the other case, we assume that (a'_1, p_1) is not a

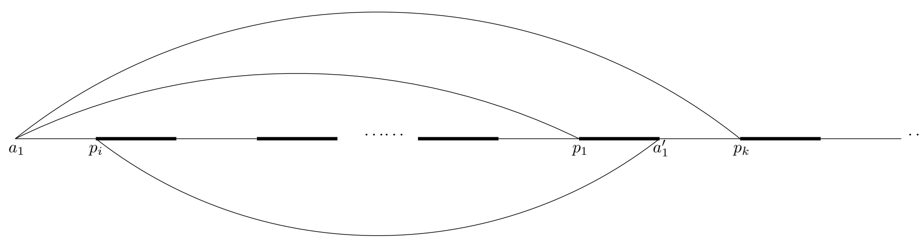

Fig. 2. a_1 is the manipulator who is copying the preference list of a'_1 and the thick edges belong to the matching of G

matched edge in H. By Theorem 10 of [10], we get a rank-maximal matching of H from G by applying an alternating path starting from a_1 (Firgure 2). Since (a'_1, p_1) is not matched in H, the alternating path contains the edge (a'_1, p_1). Suppose, a rank i post p_i (resp. rank k post p_k) is matched to a_1 (resp. a'_1) in H. By Lemma 12, the edges (a_1, p_k) and (a'_1, p_i) are also present in the reduced graph of H. Therefore, the path segment $a_1 \to p_i \to \cdots \to p_1 \to a'_1 \to p_k$ together with the edge (a_1, p_k) creates an alternating cycle in the reduced graph of H. Any alternating cycle in a reduced graph is a switching cycle [11]. If we switch along the alternating cycle in H, (a'_1, p_1) becomes a matched edge in H. Now we have arrived at the first case. Therefore we have proved that (a_1, p_1) is a rank-maximal pair in H.

References

1. Abdulkadiroğlu, A., Sönmez, T.: Random serial dictatorship and the core from random endowments in house allocation problems. Econometrica 66(3), 689–701 (1998)

2. Abraham, D.J., Cechlárová, K., Manlove, D., Mehlhorn, K.: Pareto optimality in house allocation problems. In: In proceedings of 16th International Symposium on Algorithms and Computation , ISAAC. pp. 1163–1175 (2005)
3. Abraham, D.J., Irving, R.W., Kavitha, T., Mehlhorn, K.: Popular matchings. SIAM Journal on Computing 37(4), 1030–1045 (2007)
4. Ashlagi, I., Fischer, F.A., Kash, I.A., Procaccia, A.D.: Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. Games and Economic Behavior 91, 284–296 (2015)
5. Aziz, H., Brandt, F., Harrenstein, P.: Pareto optimality in coalition formation. Games and Economic Behavior 82, 562–581 (2013)
6. Berge, C.: Two theorems in graph theory. Proceedings of the National Academy of Sciences 43(9), 842–844 (1957)
7. Dubins, L.E., Freedman, D.A.: Machiavelli and the gale-shapley algorithm. The American Mathematical Monthly 88(7), 485–494 (1981)
8. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. American Mathematical Monthly 69(1), 9–15 (1962)
9. Gale, D., Sotomayor, M.: Ms. machiavelli and the stable matching problem. The American Mathematical Monthly 92(4), 261–268 (1985)
10. Ghosal, P., Kunysz, A., Paluch, K.E.: The dynamics of rank-maximal and popular matchings. CoRR, abs/1703.10594 (2017)
11. Ghosal, P., Nasre, M., Nimbhorkar, P.: Rank-maximal matchings–structure and algorithms. In: In proceedings of 25th International Symposium on Algorithms and Computation , ISAAC. pp. 593–605 (2014)
12. Gusfield, D., Irving, R.W.: The stable marriage problem: structure and algorithms. Foundations of computing, MIT Press (1989)
13. Huang, C.C.: Cheating by men in the gale-shapley stable matching algorithm. In: In proceedings of 14th Annual European Symposium on Algorithms ESA. pp. 418–431 (2006)
14. Huang, C.C.: Cheating to get better roommates in a random stable matching. In: In proceedings of 24th Annual Symposium on Theoretical Aspects of Computer Science STACS. pp. 453–464 (2007)
15. Huang, C.C., Kavitha, T., Mehlhorn, K., Michail, D.: Fair matchings and related problems. Algorithmica 74(3), 1184–1203 (2016)
16. Hylland, A., Zeckhauser, R.: The efficient allocation of individuals to positions. Journal of Political economy 87(2), 293–314 (1979)
17. Irving, R.W.: An efficient algorithm for the stable roommates problem. Journal of Algorithms 6(4), 577–595 (1985)
18. Irving, R.W.: Matching medical students to pairs of hospitals: A new variation on a well-known theme. In: In proceedings of 6th Annual European Symposium on Algorithms, ESA. pp. 381–392 (1998)
19. Irving, R.W.: Greedy matchings. Technical Report, University of Glasgow, Tr-2003-136 (2003)
20. Irving, R.W., Kavitha, T., Mehlhorn, K., Michail, D., Paluch, K.E.: Rank-maximal matchings. ACM Transactions on Algorithms (TALG) 2(4), 602–610 (2006)
21. Kavitha, T., Shah, C.D.: Efficient algorithms for weighted rank-maximal matchings and related problems. In: In proceedings of 17th International Symposium on Algorithms and Computation, ISAAC. pp. 153–162 (2006)
22. Krysta, P., Manlove, D., Rastegari, B., Zhang, J.: Size versus truthfulness in the house allocation problem. In: In proceedings of ACM Conference on Economics and Computation, EC. pp. 453–470 (2014)

23. Nasre, M.: Popular matchings: Structure and strategic issues. SIAM J. Discrete Math. 28(3), 1423–1448 (2014)
24. Nimbhorkar, P., Rameshwar, V.: Dynamic rank-maximal matchings. In: In proceedings of 23rd International Computing and Combinatorics Conference, COCOON. pp. 433–444 (2017)
25. Paluch, K.: Capacitated rank-maximal matchings. In: In proceedings of International Conference on Algorithms and Complexity, ICAC. pp. 324–335 (2013)
26. Roth, A.E.: The economics of matching: Stability and incentives. Mathematics of operations research 7(4), 617–628 (1982)
27. Roth, A.E.: The evolution of the labor market for medical interns and residents: A case study in game theory. Journal of Political Economy 92(6), 991 (1984)
28. Roth, A.E., Postlewaite, A.: Weak versus strong domination in a market with indivisible goods. Journal of Mathematical Economics 4(2), 131–137 (1977)
29. Roth, A.E., Sönmez, T., Ünver, M.U.: Pairwise kidney exchange. J. Economic Theory 125(2), 151–188 (2005)
30. Shapley, L., Scarf, H.: On cores and indivisibility. Journal of mathematical economics 1(1), 23–37 (1974)
31. Teo, C.P., Sethuraman, J., Tan, W.P.: Gale-shapley stable marriage problem revisited: Strategic issues and applications. Management Science 47(9), 1252–1267 (2001)