

Rank-Maximal Matchings – Structure and Algorithms

Pratik Ghosal¹, Meghana Nasre ^{*2}, and Prajakta Nimbhorkar^{**3}

¹ University of Wrocław, Poland (pratikghosal20082@gmail.com)

² Indian Institute of Technology, Madras, India (meghana@cse.iitm.ac.in)

³ Chennai Mathematical Institute, India (prajakta@cmi.ac.in)

Abstract. Let $G = (\mathcal{A} \cup \mathcal{P}, E)$ be a bipartite graph where \mathcal{A} denotes a set of agents, \mathcal{P} denotes a set of posts and ranks on the edges denote preferences of the agents over posts. A matching M in G is rank-maximal if it matches the maximum number of applicants to their top-rank post, subject to this, the maximum number of applicants to their second rank post and so on.

In this paper, we develop a *switching graph* characterization of rank-maximal matchings, which is a useful tool that encodes all rank-maximal matchings in an instance. The characterization leads to simple and efficient algorithms for several interesting problems. In particular, we give an efficient algorithm to compute the set of *rank-maximal pairs* in an instance. We show that the problem of counting the number of rank-maximal matchings is $\#P$ -Complete and also give an FPRAS for the problem. Finally, we consider the problem of deciding whether a rank-maximal matching is *popular* among all the rank-maximal matchings in a given instance, and give an efficient algorithm for the problem.

1 Introduction

We consider the problem of matching applicants to posts where applicants have preferences over posts. This problem is motivated by several important real-world applications like allocation of graduates to training positions [5] and families to government housing [16]. The input to the problem is a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$, where \mathcal{A} is a set of applicants, \mathcal{P} is a set of posts, and the set E can be partitioned as $E = E_1 \cup \dots \cup E_r$, where E_i contains the edges of rank i . An edge $(a, p) \in E_i$ if p is an i th choice of a . An applicant a prefers a post p to p' if, for some $i < j$, $(a, p) \in E_i$ and $(a, p') \in E_j$. Applicant a is indifferent between p and p' if $i = j$. This ranking of posts by an applicant is called *the preference list* of the applicant. When applicants can be indifferent between posts, preference lists are said to contain ties, else preference lists are strict.

* Part of the work done by the author was supported by IIT-M initiation grant CSE/14-15/824/NFIG/MEGA.

** Part of the work has been done while the author was on a sabbatical to the Institute of Mathematics of the Czech Academy of Sciences, Prague.

The problem of matching under one-sided preferences has received lot of attention and there exist several notions of optimality like pareto-optimality [1], rank-maximality [7], popularity [2], and fairness. We focus on the well-studied notion of *rank-maximal* matchings which are guaranteed to exist in any instance. Rank-maximality was first studied under the name of *greedy matchings* by Irving [6], who also gave an algorithm for computing such matchings in case of strict lists. A rank-maximal matching matches maximum number of applicants to their rank 1 posts, subject to that, maximum number of applicants to their rank 2 posts and so on. Given an instance of the rank-maximal matchings problem possibly involving ties, Irving et al.[7] gave an $O(\min(n + r, r\sqrt{n})m)$ time algorithm to compute a rank-maximal matching. Here $n = |\mathcal{A}| + |\mathcal{P}|$, $m = |E|$, and r denotes the maximal rank in the instance. The weighted and capacitated versions of this problem have been studied in [11] and [14] respectively.

In this paper, we study the structure of the rank-maximal matchings using the notion of a *switching graph*. This notion was introduced in the context of *popularity* which is an alternative criterion of optimality in the one-sided preferences model. See [2] for a definition of popular matchings. McDermid and Irving [12] studied the switching graph of popular matchings for strict instances, and Nasre [13] extended it to the case of ties. This characterization has turned out to be useful for several problems like counting the number of popular matchings in strict instances, computing an *optimal* popular matching, developing an optimal manipulation strategy for an agent etc.

It is natural to extend the switching graph characterization to analyze rank-maximal matchings. Besides being interesting in its own right, it turns out to be useful in answering several natural questions. For instance, given instance $G = (\mathcal{A} \cup \mathcal{P}, E)$, is there a rank-maximal matching in G which matches an applicant a to a particular post p ? Is a rank-maximal matching preferred by a majority of applicants over other rank-maximal matchings in the instance? We show the following new results in this paper:

- A switching graph characterization of the rank-maximal matchings problem, and its properties, using which, we answer the questions mentioned above.
- An efficient algorithm for computing the set of *rank-maximal pairs*. An edge $(a, p) \in E$ is a rank-maximal pair if there exists a rank-maximal matching in G that matches a to p .
- We show that the problem of counting the number of rank-maximal matchings is #P-complete even for strict preference lists. We then give an FPRAS for the problem by reducing it to the problem of counting the number of perfect matchings in a bipartite graph.
- In order to choose one among possibly several rank-maximal matchings in a given instance G , we consider the question of finding a rank-maximal matching that is popular among all the rank-maximal matchings in G . We call such a matching a *popular rank-maximal matching*. We show that, given a rank-maximal matching, it can be efficiently checked whether it is a popular rank-maximal matching. If not, we output a rank-maximal matching which is more popular than the given one.

We remark that the switching graph is a weighted directed graph constructed with respect to a particular matching. In case of popular matchings, it is known from [2] that, there are at most two distinct ranked posts in an applicant’s preference list, to which he can be matched in any popular matching. This results in a switching graph with edge-weights $\{+1, -1, 0\}$. In case of rank-maximal matchings, the situation becomes more interesting since an applicant can be matched to one among several distinct ranked posts, and the edge-weights in the switching graph could be arbitrary. Surprisingly, the characterization still turns out to be similar to that of popular matchings, although the proofs are significantly different. We expect that the switching graph will find several applications apart from those shown in this paper.

2 Preliminaries

A matching M of G is a subset of edges, no two of which share an end-point. For a matched vertex u , we denote by $M(u)$ its partner in M .

Properties of maximum matchings in bipartite graphs: Let $G = (\mathcal{A} \cup \mathcal{P}, E)$ be a bipartite graph and let M be a maximum matching in G . The matching M defines a partition of the vertex set $\mathcal{A} \cup \mathcal{P}$ into three disjoint sets, defined below:

Definition 1 (Even, odd, unreachable vertices). *A vertex $v \in \mathcal{A} \cup \mathcal{P}$ 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 following lemma is well-known in matching theory; see [15] or [7] for a proof.

Lemma 1 ([15]). *Let \mathcal{E} , \mathcal{O} , and \mathcal{U} be the sets of even, odd, and unreachable vertices defined by a maximum matching M in G . Then,*

- (a) \mathcal{E} , \mathcal{O} , and \mathcal{U} are disjoint, and are the same for all the maximum matchings in G .
- (b) In any maximum matching of G , every vertex in \mathcal{O} is matched with a vertex in \mathcal{E} , and every vertex in \mathcal{U} is matched with another vertex in \mathcal{U} . The size of a maximum matching is $|\mathcal{O}| + |\mathcal{U}|/2$.
- (c) No maximum matching of G contains an edge with one end-point in \mathcal{O} and the other in $\mathcal{O} \cup \mathcal{U}$. Also, G contains no edge with one end-point in \mathcal{E} and the other in $\mathcal{E} \cup \mathcal{U}$.

Rank-maximal matchings: An instance of the rank-maximal matchings problem consists of a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$, where \mathcal{A} is a set of applicants, \mathcal{P} is a set of posts, and E can be partitioned as $E_1 \cup E_2 \cup \dots \cup E_r$. Here E_i denotes the edges of rank i , and r denotes the maximum rank any applicant assigns to a post. An edge (a, p) has rank i if p is an i th choice of a .

Definition 2 (Signature). *The signature of a matching M is defined as an r -tuple $\rho(M) = (x_1, \dots, x_r)$ 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, M' be two matchings in G , with signatures $\rho(M) = (x_1, \dots, x_r)$ and $\rho(M') = (y_1, \dots, y_r)$. Define $M \succ M'$ if $x_i = y_i$ for $1 \leq i < k \leq r$ and $x_k > y_k$.

Definition 3 (Rank-maximal matching). A matching M in G is rank-maximal if M has the maximum signature under the above ordering \succ .

Observe that all the rank-maximal matchings in an instance have the same cardinality and the same signature.

Computing Rank-maximal Matchings: Now we recall Irving et al.'s algorithm [7] for computing a rank-maximal matching in a given instance $G = (\mathcal{A} \cup \mathcal{P}, E_1 \cup \dots \cup E_r)$. Recall that E_i is the set of edges of rank i . For the sake of convenience, for each applicant a , we add a dummy last-resort post $\ell(a)$ at rank $r + 1$ in a 's preference list, and refer to the modified instance as G . This ensures that every rank-maximal matching is \mathcal{A} -complete i.e. matches all the applicants.

Let $G_i = (\mathcal{A} \cup \mathcal{P}, E_1 \cup \dots \cup E_i)$. The algorithm starts with $G'_1 = G_1$ and any maximum matching M_1 in G'_1 .

For $i = 1$ to r do the following and output M_{r+1} :

1. Partition the vertices in $\mathcal{A} \cup \mathcal{P}$ into even, odd, and unreachable as in Definition 1 and call these sets $\mathcal{E}_i, \mathcal{O}_i, \mathcal{U}_i$ respectively.
2. Delete those edges in $E_j, j > i$, which are incident on nodes in $\mathcal{O}_i \cup \mathcal{U}_i$. These are the nodes that are matched by every maximum matching in G'_i .
3. Delete all the edges from G'_i between a node in \mathcal{O}_i and a node in $\mathcal{O}_i \cup \mathcal{U}_i$. We refer to these edges as $\mathcal{O}_i \mathcal{O}_i$ and $\mathcal{O}_i \mathcal{U}_i$ edges respectively. These are the edges which do not belong to any maximum matching in G'_i .
4. Add the edges in E_{i+1} to G'_i and call the resulting graph G'_{i+1} .
5. Determine a maximum matching M_{i+1} in G'_{i+1} by augmenting M_i .

The algorithm constructs a graph G'_{r+1} . We construct a *reduced graph* G' by deleting all the edges from G'_{r+1} between a node in \mathcal{O}_{r+1} and a node in $\mathcal{O}_{r+1} \cup \mathcal{U}_{r+1}$. The graph G' will be used in subsequent sections.

We note the following invariants of Irving et al.'s algorithm:

- (I1) For every $1 \leq i \leq r$, every rank-maximal matching in G_i is contained in G'_i .
- (I2) The matching M_i is rank-maximal in G_i , and is a maximum matching in G'_i .
- (I3) If a rank-maximal matching in G has signature $(s_1, \dots, s_i, \dots, s_r)$ then M_i has signature (s_1, \dots, s_i) .
- (I4) The graphs $G'_i, 1 \leq i \leq r + 1$ constructed at the end of iteration i of Irving et al.'s algorithm, and G' are independent of the rank-maximal matching computed by the algorithm. This follows from Lemma 1 and invariant I2.

3 Switching Graph Characterization

In this section, we describe the switching graph characterization of rank-maximal matchings, and show its application in computing *rank-maximal pairs*.

Let M be a rank-maximal matching in G and let $G' = (\mathcal{A} \cup \mathcal{P}, E')$ be the reduced graph as described in Section 2.

Definition 4 (Switching Graph). The switching graph $G_M = (V_M, E_M)$ with respect to a rank-maximal matching M is a directed weighted graph with $V_M = \mathcal{P}$ and $E_M = \{(p_i, p_j) \mid \exists a \in \mathcal{A}, (a, p_i) \in M, (a, p_j) \in E'\}$. Further, weight of an edge (p_i, p_j) is $w(p_i, p_j) = \text{rank}(a, p_j) - \text{rank}(a, p_i)$, where $\text{rank}(a, p)$ is the rank of a post p in the preference list of an applicant a .

Thus an edge $(p_i, p_j) \in E_M$ iff there exists an applicant a such that $(a, p_i) \in M$ and (a, p_j) is an edge in the graph G' . We define the following notation:

1. *Sink vertex:* A vertex p of G_M is called a *sink* vertex, if p has no outgoing edge in G_M and $p \in \mathcal{E}_1 \cap \mathcal{E}_2 \cap \dots \cap \mathcal{E}_{r+1}$. Recall that \mathcal{E}_i is the set of vertices which were even in the graph G'_i constructed in the i th iteration of Irving et al.'s algorithm.
2. *Sink and non-sink components of G_M :* A connected component \mathcal{X} in the underlying undirected graph of G_M is called a *sink component* if \mathcal{X} contains one or more sink vertices, and a *non-sink component* otherwise.
3. *Switching paths and switching cycles:* A path $T = \langle p_0, p_1, \dots, p_{k-1} \rangle$ in G_M is called a *switching path* if T ends in a sink vertex and $w(T) = 0$. Here, $w(T)$ is the sum of the weights of the edges in T . A cycle $C = \langle p_0, \dots, p_{k-1}, p_0 \rangle$ in G_M is called a *switching cycle* if $w(C) = 0$.
4. *Switching operation:* Let $T = \langle p_0, p_1, \dots, p_{k-1} \rangle$ be a switching path in G_M . Let $\mathcal{A}_T = \{a \in \mathcal{A} \mid M(a) \in T\}$. Further, let $M(a_i) = p_i$ for $0 \leq i \leq k-2$. We denote by $M' = M \cdot T$, the matching obtained by *applying* T to M . Thus, for $a_i \in \mathcal{A}_T$, $M'(a_i) = p_{i+1}$, and for $a \notin \mathcal{A}_T$, $M'(a) = M(a)$. The matching $M \cdot C$, obtained by *applying* a switching cycle C to M is defined analogously. We also refer to $M \cdot C$ or $M \cdot T$ as a *switching operation*.

Figure 1 illustrates an example instance along with its switching graph.

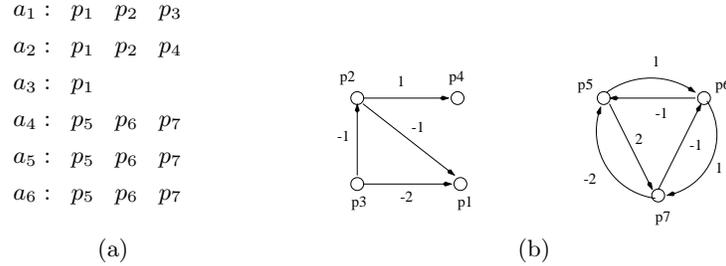


Fig. 1. (a) Preference lists of agents $\{a_1, \dots, a_6\}$ in increasing order of ranks. (b) Switching graph G_M with respect to rank-maximal matching $M = \{(a_1, p_3), (a_2, p_2), (a_3, p_1), (a_4, p_7), (a_5, p_5), (a_6, p_6)\}$. The vertex p_4 is the only sink-vertex and the path (p_3, p_2, p_4) is a switching path. Note that every directed cycle is a switching cycle.

3.1 Properties of the switching graph

In this section, we prove several useful properties of the switching graph by characterizing switching paths and switching cycles.

In the following lemma, we show that a switching operation on a rank-maximal matching M results in another rank-maximal matching in G .

Lemma 2. *Let T (resp. C) be a switching path (resp. switching cycle) in G_M . Then, $M' = M \cdot T$ (resp. $M' = M \cdot C$) is a rank-maximal matching in G .*

Proof. We prove the lemma for a switching path T . A similar argument follows for a switching cycle. To show that M' is rank-maximal, we show that M and M' have the same signature.

Let $T = \langle p_0, p_1, \dots, p_{k-1} \rangle$ be a switching path in G_M . Let $\mathcal{A}_T = \{a \mid M(a) \in T\}$. By the definition of a *switch*, we know that $|M| = |M'|$ and for each $a \notin \mathcal{A}_T$, we have $M'(a) = M(a)$. Thus, it suffices to show that the signatures of M and M' restricted to the applicants in \mathcal{A}_T are the same. We denote them by $\rho_T(M) = (x_1, x_2, \dots, x_r)$ and $\rho_T(M') = (y_1, y_2, \dots, y_r)$ respectively. Note that an edge of rank i in M contributes $-i$ to the weight of T , whereas one in M' contributes i . Further, since T is a switching path, $w(T) = 0$. Thus,

$$w(T) = (y_1 - x_1) + 2(y_2 - x_2) + \dots + r(y_r - x_r) = 0 \quad (1)$$

Since we consider only applicants in \mathcal{A}_T , we know that, $\sum_{i=1}^r x_i = \sum_{i=1}^r y_i$, i.e.,

$$\sum_{i=1}^r (x_i - y_i) = 0 \quad (2)$$

For contradiction, assume that $\rho_T(M) \succ \rho_T(M')$. That is, there exists an index j such that $x_j > y_j$ and, for $1 \leq i < j$, we have $x_i = y_i$. Then, for Eqn. 2 to be satisfied, there exists an index $\ell > j$ such that $x_\ell < y_\ell$. In fact we will show the following stronger claim:

Claim. There exists an index $\ell > j$ such that $\sum_{i=1}^{\ell} (x_i - y_i) < 0$.

Before proving the claim, we show how it suffices to complete the proof of the lemma. Assuming the claim, consider the reduced graph G'_ℓ constructed in the ℓ th iteration of Irving et al.'s algorithm.

As $\sum_{i=1}^{\ell} (x_i - y_i) < 0$, we have $\sum_{i=1}^{\ell} x_i < \sum_{i=1}^{\ell} y_i$. Thus $|M \cap G'_\ell| < |M' \cap G'_\ell|$. However, by Invariant (I2) (ref. Section 2), this contradicts the fact that every rank-maximal matching restricted to any rank ℓ is also a maximum matching in the reduced graph G'_ℓ . This completes the proof of the lemma. We prove the claim below.

Proof (of claim): Assume the contrary, i.e. $\sum_{i=1}^k (x_i - y_i) \geq 0$ for all k . Note that this is trivially true for $k \leq j$, by our choice of j . Equivalently, $\sum_{i=k+1}^r (x_i - y_i) \leq 0$ for all k . Define $T_k = \sum_{i=k}^r (x_i - y_i)$ for $1 \leq k \leq r$. Thus, to prove the

claim, it suffices to show that there exists an index ℓ such that $T_\ell > 0$. Now consider Eqn. 1. It can be rewritten as follows:

$$(x_1 - y_1) + 2(x_2 - y_2) + \dots + r(x_r - y_r) = T_1 + T_2 + \dots + T_r = 0 \quad (3)$$

We know that $T_1 = 0$, because it is the left-side of Eqn. 2. Now, consider the term $T_r = x_r - y_r$. If $T_r = 0$, we can eliminate x_r and y_r and get equations in $r - 1$ variables. If $T_r > 0$, then Eqn. 2 implies that the claim holds for $k = r - 1$. So, without loss of generality, we can assume $T_r < 0$. But then, to satisfy Eqn. 3, there exists an index i , $1 < i < r$, such that $T_i > 0$. This implies that the claim holds for $\ell = i - 1$. This completes the proof of the claim. \square

Now we address the question of recognition of switching paths and switching cycles in G_M . In Lemma 3, we show that every cycle in G_M is in fact a switching cycle, that is, a zero-weight cycle. In Lemma 4, we characterize switching paths. The proofs for both these lemmas can be found in [4].

Lemma 3. *Let M be a rmm in G , and C be a cycle in G_M . Then $w(C) = 0$.*

Proof. (Sketch) Let C' be the alternating cycle in G' , corresponding to the cycle C in G_M . To prove the Lemma, it suffices to show that, C' has an equal number of matched and unmatched edges of any rank i , and hence $w(C) = 0$. We complete the proof by induction on i . \square

Lemma 4. *Let M be a rmm in G , and G_M be the switching graph with respect to M . Recall that \mathcal{E}_i is the set of even vertices in the graph G'_i constructed in the i th iteration of Irving et al.'s algorithm. The following properties hold :*

1. *Let p be an unmatched post in M . Then $p \in \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{r+1}$ and therefore is a sink in G_M .*
2. *A post p belongs to a sink component iff $p \in \mathcal{E}_{r+1}$. A post p belongs to a non-sink component iff $p \in \mathcal{U}_{r+1}$.*
3. *Let T be a path from a post p to some sink q in G_M . Then $w(T) = 0$ iff $p \in \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{r+1}$.*

In the following theorem, we prove that every rank-maximal matching can be obtained from M by applying suitable *switches*.

Theorem 1. *Every rank-maximal matching M' in G can be obtained from M by applying to M vertex-disjoint switching paths and switching cycles in G_M .*

Proof. Consider any rank-maximal matching M' in G . We show that M' can be obtained from M by applying a set of vertex-disjoint switching paths and switching cycles of G_M . Consider $M \oplus M'$ which is a collection of vertex-disjoint paths and cycles in G . Also note that the cycles and paths contain alternating edges of M and M' . We show that the paths and cycles in $M \oplus M'$ are switching paths and switching cycles in G_M .

From the invariants of Irving et al.'s algorithm mentioned in Section 2, all the edges of M and M' are also present in G' . A cycle in $M \oplus M'$ has alternating

edges of M and M' , and hence has a corresponding directed cycle in G_M . As proved in Lemma 3, every cycle in G_M is a switching cycle.

Now we consider paths in $M \oplus M'$. All the paths are of even length, since all the rank-maximal matchings are of the same cardinality. Let $T_G = \langle p_1, a_1, \dots, p_k, a_k, p_{k+1} \rangle$ be any even-length path in $M \oplus M'$ with p_{k+1} unmatched in M and p_1 unmatched in M' . For every $1 \leq i \leq k$, let $M(p_i) = a_i$. It is easy to see that the path $T = \langle p = p_1, p_2, \dots, p_{k+1} = p' \rangle$ is present in G_M and it ends in a sink p' . Our goal is to show that $w(T) = 0$. For this, we prove that $p_1 \in \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{r+1}$. Note that M' is a rank-maximal matching in G and M' leaves the post $p = p_1$ unmatched. As every post in $\mathcal{O}_i \cup \mathcal{U}_i$ for any i is matched in every rank-maximal matching, $p_1 \notin \mathcal{O}_i \cup \mathcal{U}_i$ for $1 \leq i \leq r + 1$. Therefore $p_1 \in \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{r+1}$; Thus, using Lemma 4, we can conclude that the path T has weight $w(T) = 0$ in G_M , and hence is a switching path in G_M .

Applying these switching paths and cycles to M gives us the desired matching M' , thus completing the proof. \square

3.2 Generating all rank-maximal pairs

In this section we give an efficient algorithm to compute the set of rank-maximal pairs, defined below:

Definition 5. *An edge (a, p) is a rank-maximal pair if there exists a rank-maximal matching M in G such that $M(a) = p$.*

We refer to rank-maximal pairs as *rmm-pairs*. We show that the set of rmm-pairs can be computed in time linear in the size of the switching graph G_M constructed with respect to any rank-maximal matching M . We prove the following theorem:

Theorem 2. *The set of rmm-pairs for an instance $G = (\mathcal{A} \cup \mathcal{P}, E)$ can be computed in $O(\min(n + r, r\sqrt{n})m)$ time.*

Proof. We note that, by Theorem 1, an edge (a, p) is a rmm-pair iff (i) $(a, p) \in M$ or, (ii) the edge $(M(a), p)$ belongs to a switching cycle in G_M or, (iii) the edge $(M(a), p)$ belongs to a switching path in G_M .

Condition (i) can be checked by computing a rank-maximal matching M which takes $O(\min(n + r, r\sqrt{n})m)$ time. Condition (ii) can be checked by computing strongly connected components of G_M , which takes time linear in the size of G_M .

To check Condition (iii), note that a post p has a zero-weight path to a sink if and only if $p \in \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{r+1}$ by Lemma 4 (3). Moreover, all the paths from such a post p to a sink have weight zero. Therefore, performing a DFS from each $p \in \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{r+1}$ and marking all the edges encountered in the DFS (not just the tree edges) gives all the pairs which satisfy Condition (iii). \square

4 Counting Rank-Maximal Matchings

We prove that the problem of counting the number of rank-maximal matchings in an instance is #P-complete, and give an FPRAS for the same.

4.1 Hardness of Counting

We prove #P-hardness by reducing the problem of counting the number of matchings in 3-regular bipartite graphs to counting the number of rank-maximal matchings. The former was shown to be #P-complete by Dagum and Luby [3].

Reduction for lists with ties: First let us consider the case when preference lists may contain ties⁴. Let $H = (X \cup Y, E)$ be a 3-regular bipartite graph. We construct an instance G of the rank-maximal matchings problem by setting $G = H$ and assigning rank 1 to all the edges in E . It is well-known that a k -regular bipartite graph admits a perfect matching for any k . It is easy to see that every perfect matching in H is a rank-maximal matching in G and vice versa. This proves the #P-hardness of the problem for the case of ties.

Reduction for strict lists: Let $H = (X \cup Y, E)$ be a 3-regular bipartite graph, with $|X| = |Y| = n$. The corresponding instance $G = (\mathcal{A} \cup \mathcal{P}, E_G)$ of the rank-maximal matchings problem is as follows:

$$\mathcal{A} = \{a_x : x \in X\} \cup \{ad_1, ad_2, \dots, ad_{n-3}\}; \mathcal{P} = \{p_y : y \in Y\} \cup \{pd_1, pd_2, \dots, pd_{n-3}\}$$

Here $ad_i, pd_i, 1 \leq i \leq n - 3$ are dummy agents and dummy posts respectively.

To construct the preference lists of agents in \mathcal{A} , we fix an arbitrary ordering on the vertices in Y i.e. $order : Y \rightarrow \{1, \dots, n\}$. This assigns an ordering on the posts in \mathcal{P} . The preference lists of the agents can be described as below:

- A dummy agent ad_i has a preference list of length one, with dummy post pd_i as his rank 1 post.
- The preference list of an agent a_x consists of posts $p_{y_1}, p_{y_2}, p_{y_3}$ ranked at $order(y_1), order(y_2)$, and $order(y_3)$ respectively, where y_1, y_2, y_3 denote the 3 neighbors of x in H . The remaining places in the preference list of a_x are filled using the $n - 3$ dummy posts.

Following Lemma (see [4] for proof) shows the correctness of the reduction.

Lemma 5. *Let H be a 3-regular bipartite graph and let G be the rank-maximal matchings instance constructed from H as above. There is a one-to-one correspondence between perfect matchings in H and rank-maximal matchings in G .*

Using Lemma 5 and our observation for ties, we conclude the following:

Theorem 3. *The problem of counting the number of rank-maximal matchings in an instance is #P-Complete for both strict and tied preference lists.*

4.2 An FPRAS for Counting Rank-Maximal Matchings

Given the hardness result in Section 4.1, it is unlikely to be able to count the number of rank-maximal matchings in an instance in polynomial time. We now show that there exists a fully polynomial-time randomized approximation scheme (FPRAS) for the problem. We use the following result by Jerrum et al. [9]:

⁴ Recall that preference lists are said to contain ties if an applicant ranks two or more posts at the same rank.

Theorem 4 ([9]). *There exists an FPRAS for the problem of counting the number of perfect matchings in a bipartite graph.*

We give a polynomial-time reduction from the problem of counting the number of rank-maximal matchings (denoted as #RMM) to the problem of counting the number of perfect matchings in a bipartite graph (denoted as #BPM).

Reduction from #RMM to #BPM: Given an instance $G = (\mathcal{A} \cup \mathcal{P}, E)$ of the rank-maximal matchings problem, we first construct another instance H of the rank-maximal matchings problem, which is used to get an instance I of the bipartite perfect matchings problem. The steps of the construction are as follows:

1. For every $a \in \mathcal{A}$, introduce a dummy last-resort post $\ell(a)$ ranked $r + 1$. This ensures that every rank-maximal matching is \mathcal{A} -complete.
2. Let M be any rank-maximal matching in G , let G' be the reduced graph obtained by Irving et al.'s algorithm (ref. Section 2).
3. Let k be the number of unmatched posts in G' . Introduce k dummy applicants ad_1, \dots, ad_k . The preference list of each dummy applicant consists of all the posts in G' which are in $\mathcal{E}_1 \cap \dots \cap \mathcal{E}_{r+1}$, tied at rank $r + 2$.
4. The instance H consists of all the applicants in G and their preference lists in G , together with the dummy applicants and their preference lists introduced above. The set of posts in H is the same as that in G .
5. The instance I of bipartite perfect matchings problem is simply the reduced graph H' , obtained by executing Irving et al.'s algorithm on H .

Correctness of the reduction follows from the following lemma, the proof (in [4]) uses the switching graph characterization.

Lemma 6. *Let G be the rank-maximal matchings instance and let H and I be the rank-maximal matchings instance and the bipartite perfect matchings instance respectively constructed as above. Then, the following hold:*

1. *Corresponding to each rank-maximal matching M in G , there are exactly $k!$ distinct rank-maximal matchings in H .*
2. *Each rank-maximal matching in H matches all the applicants and posts, and all its edges appear in I . Hence it is a perfect matching in the instance I .*
3. *A matching in G that is not rank-maximal has no corresponding perfect matching in I .*

The FPRAS for #RMM involves the following steps:

1. The reduction from #RMM instance G to #BPM instance I ,
2. Running Jerrum et al.'s FPRAS on I to get an approximate count, say C , of the number of perfect matchings in I ,
3. Dividing C by $k!$ to get an approximate count of number of rank-maximal matchings in G .

Steps 1 and 2 clearly work in polynomial time. For step 3, note that both C and k are at most $n!$ and can be represented in $O(n \log n)$ bits, which is polynomial in the size of G . Therefore Step 3 also works in polynomial time. This completes the FPRAS for #RMM problem.

5 Popularity of Rank-Maximal Matchings

As mentioned earlier, an instance of the rank-maximal matchings problem may admit more than one rank-maximal matching. To choose one rank-maximal matching, it is natural to impose an additional optimality criterion. Such a question has been considered earlier in the context of popular matchings by [10, 12] and also in the context of the stable marriage problem [8]. The additional notion of optimality that we impose is the notion of popularity, defined below:

Definition 6 (Popular matching). *A matching M is more popular than matching M' (denoted by $M \succ_p M'$) if the number of applicants that prefer M to M' is more than the number of applicants that prefer M' to M . A matching M is popular if no matching M' is more popular than M .*

An applicant a prefers matching M to M' if either (i) a is matched in M and unmatched in M' , or (ii) a is matched in both and prefers the post $M(a)$ to $M'(a)$. We consider the following question: Given an instance of the rank-maximal matchings problem, is there a rank-maximal matching that is popular in the set of all rank-maximal matchings? We refer to such a matching as a *popular rank-maximal matching*. There are simple instances in which there is no popular matching; further there is no popular rank-maximal matching. However, if a popular rank-maximal matching exists, it seems an appealing choice since it enjoys both rank-maximality and popularity. We make partial progress on this question. Using the switching graph characterization developed in Section 3, we give a simple algorithm to determine if a given rank-maximal matching M is a popular rank-maximal matching. If not, our algorithm outputs a rank-maximal matching M' which is more popular than M .

Outline of the algorithm: Given a graph $G = (\mathcal{A} \cup \mathcal{P}, E)$ and a rank-maximal matching M in G , the algorithm first constructs the switching graph G_M corresponding to M . Now consider the following re-weighted graph \tilde{G}_M where positive weights of edges in G_M are replaced by +1 weights and negative weights by -1 . Thus a -1 weight edge (p_i, p_j) in \tilde{G}_M implies that $M(p_i)$ prefers p_j to p_i .

Let T be a switching path in G_M , and let \tilde{T} be the corresponding path in \tilde{G}_M . It is easy to see that if $w(\tilde{T}) < 0$ in \tilde{G}_M , then $M' = M \cdot T$ is more popular than M . Same holds for a switching cycle in G_M . Therefore, M is a popular rank-maximal matching, if and only if there is no negative-weight path to sink or negative-weight cycle in \tilde{G}_M .

To check this, we use shortest path computations using Bellman-Ford algorithm in a suitably modified graph. The details of the algorithm and proof of the following lemma, which establishes correctness, can be found in [4].

Lemma 7. *A given rank-maximal matching M is popular if and only if there is no negative-weight path to a sink or a negative-weight cycle in the re-weighted switching graph.*

Thus we get an $O(mn)$ time algorithm for checking whether a given rank-maximal matching is a popular rank-maximal matching, where m and n are number of edges and vertices in the switching graph respectively.

Acknowledgment: We thank Partha Mukhopadhyay for a proof of Lemma 2.

References

1. D. J. Abraham, K. Cechlárová, D. F. Manlove, and K. Mehlhorn. Pareto-optimality in house allocation problems. In *Proceedings of 15th ISAAC*, pages 3–15, 2004.
2. D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37(4):1030–1045, 2007.
3. P. Dagum and M. Luby. Approximating the permanent of graphs with large factors. *Theor. Comput. Sci.*, 102(2):283–305, 1992.
4. P. Ghoshal, M. Nasre, and P. Nimbhorkar. Rank maximal matchings – structure and algorithms. *CoRR*, abs/1409.4977, 2014.
5. A. Hylland and R. Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979.
6. R. W. Irving. Greedy matchings. *Technical Report, University of Glasgow*, TR-2003-136, 2003.
7. R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. E. Paluch. Rank-maximal matchings. *ACM Transactions on Algorithms*, 2(4):602–610, 2006.
8. R. W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the “optimal” stable marriage. *Journal of the ACM*, 34(3):532–543, 1987.
9. M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.
10. T. Kavitha and M. Nasre. Note: Optimal popular matchings. *Discrete Applied Mathematics*, 157(14):3181–3186, 2009.
11. T. Kavitha and C. D. Shah. Efficient algorithms for weighted rank-maximal matchings and related problems. In *Proceedings of 17th ISAAC*, pages 153–162, 2006.
12. E. McDermid and R. W. Irving. Popular matchings: structure and algorithms. *Journal of Combinatorial Optimization*, 22(3):339–358, 2011.
13. M. Nasre. Popular Matchings: Structure and Cheating Strategies. In *Proceedings of 30th STACS*, pages 412–423, 2013.
14. K. E. Paluch. Capacitated rank-maximal matchings. In *Proceedings of 8th CIAC*, pages 324–335, 2013.
15. W. R. Pulleyblank. Handbook of combinatorics (vol. 1). chapter Matchings and Extensions, pages 179–232. MIT Press, Cambridge, MA, USA, 1995.
16. Y. Yuan. Residence exchange wanted: A stable residence exchange problem. *European Journal of Operational Research*, 90(3):536 – 546, 1996.