Popular Matchings: Structure and Cheating Strategies*

Meghana Nasre[1]

1 **University of Texas at Austin, USA.**

Abstract

We consider the cheating strategies for the popular matchings problem. 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 the edges in E are ranked. Each agent ranks a subset of posts in an order of preference, possibly involving ties. A matching M is popular if there exists no matching M' such that the number of agents that prefer M' to M exceeds the number of agents that prefer M to M'. Consider a centralized market where agents submit their preferences and a central authority matches agents to posts according to the notion of popularity. Since a popular matching need not be unique, we assume that the central authority chooses an arbitrary popular matching. Let a_1 be the sole manipulative agent who is aware of the true preference lists of all other agents. The goal of a_1 is to falsify her preference list to get *better always*, that is, to improve the set of posts she gets matched to in the falsified instance. We show that the optimal cheating strategy for a single agent to get *better always* can be computed in $O(m + n)$ time when preference lists are all strict and in $O(\sqrt{n}m)$ time when preference lists are allowed to contain ties. Here $n = |\mathcal{A}| + |\mathcal{P}|$ and $m = |E|$.

To compute the cheating strategies, we develop a *switching graph* characterization of the popular matchings problem involving ties. The switching graph characterization was studied for the case of strict lists by McDermid and Irving (J. Comb. Optim. 2011) and was open for the case of ties. We show an $O(\sqrt{n}m)$ time algorithm to compute the set of *popular pairs* using the switching graph. These results are of independent interest and answer a part of the open questions posed by McDermid and Irving.

1998 ACM Subject Classification G.2.1 Combinatorial algorithms G.2.2 Graph algorithms

Keywords and phrases bipartite matchings, preferences, cheating strategies

Digital Object Identifier 10.4230/LIPIcs.STACS.2013.412

1 Introduction

We consider the cheating strategies for the popular matchings problem. 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 the edges in E are ranked. Each agent ranks a subset of posts in an order of preference, possibly involving ties. This ranking of posts by an agent is called the preference list of the agent. An agent a prefers post p_i to post p_j if the rank of post p_i is smaller than the rank of post p_j in a's preference list. An agent a is indifferent between posts p_i and p_j if they have the same rank on a's preference list. When agents can be indifferent between posts, the preference lists are said to contain ties, otherwise the preference lists are strict. A matching M of G is a subset of edges, no two of which share an end point. For a matched vertex u, let $M(u)$ denote its partner in the matching M. An agent a prefers a matching M to another matching M' if (i) a is matched in M but unmatched in M', or (ii) a prefers $M(a)$ to $M'(a)$.

* This work was supported in part by NSF grant CCF-0830737.

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 412–423
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

▶ **Definition 1.** A matching M is *more popular than* M' if the number of agents that prefer M is greater than the number of agents that prefer M'. A matching M is *popular* if there is no matching M' that is more popular than M.

There exist simple instances that do not admit any popular matching – however, when an instance admits a popular matching, there may be more than one popular matching. Abraham et al. [1] characterized the instances that admit popular matchings and gave efficient algorithms to compute a popular matching if one exists.

Our problem. Consider a centralized matching market where each agent $a \in \mathcal{A}$ submits a preference over a subset of posts and a central authority matches agents to posts using the criteria of popularity. Let a_1 be the sole manipulative agent who is aware of the true preference lists of all other agents and the preference lists of $a \in \mathcal{A} \setminus \{a_1\}$ remain fixed throughout. The goal of a_1 is clear: she wishes to falsify her preference list so as to improve the post that she gets matched to as compared to the post she got when she was truthful. Since there may be more than one popular matching in an instance, we assume that the central authority chooses an arbitrary popular matching. Let $G = (\mathcal{A} \cup \mathcal{P}, E)$ denote the instance where ranks on the edges represent true preferences of all the agents. Let H denote the instance obtained by falsifying the preference list of a_1 alone. We assume that G admits a popular matching and a_1 falsifies in order to create an instance H which also admits a popular matching. Note that it may be possible for a_1 to falsify her preference list such that H does not admit any popular matching. But we do not consider such a falsification.

Agent a_1 wishes to falsify her preference list to ensure that (i) every popular matching in H matches her to a post that is at least as good as the most-preferred post that she gets matched to in G, and (ii) some popular matching in H matches a_1 to a post better than the most-preferred post p that she gets matched to in G, assuming that p is not a_1's true first choice post. We term this strategy of a_1 as '*better always*' strategy.

1.1 Our contributions

- Let a_1 be the sole manipulative agent who wishes to get *better always*. The optimal strategy for a_1 can be computed in $O(m + n)$ time when preference lists are all strict and in $O(\sqrt{n}m)$ time when preference lists are allowed to contain ties.
- To compute the cheating strategies, we develop a *switching graph* characterization of the popular matchings problem involving ties. Such a characterization was studied for the case of strict lists by McDermid and Irving [10] and it was open for the case of ties. Using the switching graph, we show an $O(\sqrt{n}m)$ time algorithm to compute the set of *popular pairs*. An edge $(a, p) \in E$ is a popular pair if there exists a popular matching M in G such that $(a, p) \in M$. We also show that counting the total number of popular matchings in an instance with ties is #P-Complete. The switching graph characterization is of independent interest and answers a part of the open questions in [10].

1.2 Related work

The work in this paper is motivated by the work of Teo et al. [13] where they study the strategic issues of the stable marriage problem [2]. The stable marriage problem is a generalization of our problem where both the sides of the bipartition (usually referred to as men and women) rank members of the opposite side in order of their preference. Teo et al. [13] study the strategic issues of the stable marriage problem where women are required to give complete preference lists and there is a sole manipulative woman. Further, she is aware of the true preference lists of all the other women. Teo et al. [13] compute an optimal cheating strategy for a single woman under this model. Huang [4] studies the strategic issues of the stable room-mates problem [2] under a similar model. In the same spirit, we study the strategic issues of the popular matchings problem.

The notion of popular matchings was introduced by Gärdenfors [3] in the context of the stable marriage [2]. Abraham et al. [1] studied the problem for one-sided preference lists and gave a characterization of instances which admit a popular matching. Subsequent to this result, the popular matchings problem has received a lot of attention [8] [9] [7] [5] [6]. However, to the best of our knowledge none of them is motivated by the strategic issues of the popular matchings problem.

2 Background

We first review the following well known 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: a vertex $v \in \mathcal{A} \cup \mathcal{P}$ is *even* (resp. *odd*) if there is an even (resp. odd) length alternating path in G w.r.t. M from an unmatched vertex to v. A vertex v is *unreachable* if there is no alternating path from an unmatched vertex to v. Denote by $\mathcal{E}$, $\mathcal{O}$, and $\mathcal{U}$ the sets of even, odd, and unreachable vertices, respectively, in G. The following lemma is well known in matching theory; refer [12] for a detailed exposition and proof.

▶ **Lemma 2** ([12] Dulmage Mendelsohn). *Let $\mathcal{E}$, $\mathcal{O}$, and $\mathcal{U}$ be the sets of vertices defined by a maximum matching M in G. Then,*

(a) $\mathcal{E}$, $\mathcal{O}$, and $\mathcal{U}$ are pairwise disjoint, and independent of the maximum matching M 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 between a vertex in $\mathcal{O}$ and a vertex in $\mathcal{O} \cup \mathcal{U}$. Also, G contains no edge between a vertex in $\mathcal{E}$ and a vertex in $\mathcal{E} \cup \mathcal{U}$.

We now review the characterization of the popular matchings problem from [1]. As was done in [1], we create a unique last-resort post $\ell(a)$ for each agent a. In this way, we can assume that every agent is matched, since any unmatched agent a can be paired with $\ell(a)$. For an agent a, let $f(a)$ be the set of rank-1 posts for a. To define $s(a)$, let us consider the graph $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$ on rank-1 edges in G and let M_1 be any maximum matching in G_1. Let $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$ define the partition of vertices $\mathcal{A} \cup \mathcal{P}$ with respect to M_1 in G_1. For any agent a, let $s(a)$ denote the set of most preferred posts which belong to $\mathcal{E}_1$ by the above partition. Abraham et al. [1] proved the following theorem.

▶ **Theorem 3** ([1]). *A matching M is popular in G iff*
(1) $M \cap E_1$ is a maximum matching of $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$, and
(2) for each agent a, $M(a) \in \{f(a) \cup s(a)\}$.

The algorithm for solving the popular matching problem is as follows: each $a \in \mathcal{A}$ determines the sets $f(a)$ and $s(a)$. An $\mathcal{A}$-complete matching (a matching that matches all agents) which is maximum in G_1 and matches each a to a post in $\{f(a) \cup s(a)\}$ needs to be determined. If no such matching exists, then G does not admit a popular matching. Abraham et al. [1] gave an $O(\sqrt{n}m)$ time algorithm to compute a popular matching in G which is presented as Algorithm 2.1. Steps 7–11 are added by us and will be used to define the switching graph in Section 3. Abraham et al. [1] also showed a simpler characterization in case of strict lists which results in an $O(m + n)$ time algorithm to return a popular matching if one exists.

Let $G' = (\mathcal{A} \cup \mathcal{P}, E')$ denote the graph in which every agent a has edges incident to $\{f(a) \cup s(a)\}$. Step 4 of Algorithm 2.1 deletes edges from G' which cannot be present in any maximum matching of G_1. We extend this further and in Step 9 delete edges from G' which cannot be present in any popular matching in G. For this, let us partition the vertex set $\mathcal{A} \cup \mathcal{P}$ as $\mathcal{O}_2, \mathcal{E}_2$ and $\mathcal{U}_2$ with respect to a popular matching M in G'. Since any popular matching M is a maximum matching in G', by Lemma 2(c), the matching M cannot contain edges of the form $\mathcal{O}_2\mathcal{O}_2$ and $\mathcal{O}_2\mathcal{U}_2$. However, since M matches every agent, it implies that $\mathcal{A} \cap \mathcal{E}_2 = \emptyset$ and $\mathcal{P} \cap \mathcal{O}_2 = \emptyset$. Thus, there are no $\mathcal{O}_2\mathcal{O}_2$ edges in G'. Hence, any edge (a, p) deleted in Step 9 is of the form $a \in \mathcal{O}_2$ and $p \in \mathcal{U}_2$. We can now claim the following.

▶ Claim 4. Let a be an agent such that $a \in \mathcal{U}_2$. Then, in Step 9 of Algorithm 2.1, no edge incident on a gets deleted. Let a be an agent such that $a \in \mathcal{E}_1$. Then, in Step 4 of Algorithm 2.1, no edge incident on a gets deleted.

Algorithm 2.1 $O(\sqrt{n}m)$-time algorithm for the popular matching problem [1] (Steps 1–6).

Input: $G = (\mathcal{A} \cup \mathcal{P}, E)$.

1: Construct the graph $G' = (\mathcal{A} \cup \mathcal{P}, E')$, where $E' = \{(a, p) : a \in \mathcal{A}$ and $p \in f(a) \cup s(a)\}$.
2: Construct the graph $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$ and let M_1 be any maximum matching in G_1.
3: Partition $\mathcal{A} \cup \mathcal{P}$ as $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$ with respect to M_1 in G_1.
4: Remove any edge in G' between a node in $\mathcal{O}_1$ and a node in $\mathcal{O}_1 \cup \mathcal{U}_1$.
5: Determine a maximum matching M in G' by augmenting M_1.
6: Return M if it is $\mathcal{A}$-complete, otherwise return *"no popular matching"*.
7: **if** G admits a popular matching **then**
8: Partition $\mathcal{A} \cup \mathcal{P}$ as $\mathcal{O}_2, \mathcal{E}_2, \mathcal{U}_2$ with respect to M in G'.
9: Remove any edge in G' between a node in $\mathcal{O}_2$ and a node in $\mathcal{U}_2$.
10: Denote the resulting graph as $G'' = (\mathcal{A} \cup \mathcal{P}, E'')$.
11: **end if**

▶ **Definition 5.** For $a \in \mathcal{A}$, let $choices(a) = \{p \in \mathcal{P} : (a, p)$ is an edge in $G''\}$.

3 The switching graph characterization

In this section we develop the *switching graph* for the popular matchings problem with ties. In case of strict lists, McDermid and Irving [10] defined a switching graph $G_M = (\mathcal{P}, E_M)$ as a directed graph on the posts of G and the edge set E_M was determined by a popular matching M in G. In fact, a similar graph was defined even before that by Mahdian [8] (again for strict lists) to study existence of popular matchings in random instances.

Let G be an instance of the popular matchings problem with ties and let M be a popular matching in G. The switching graph $G_M = (\mathcal{P}, E_M)$ is a directed weighted graph on the posts $\mathcal{P}$ of G and is defined with respect to a popular matching M in G. The edge set E_M is defined using the pruned graph $G'' = (\mathcal{A} \cup \mathcal{P}, E'')$ constructed in Step 10 of Algorithm 2.1. There exists an edge from p_i to p_j (with $p_i \neq p_j$) iff for some $a \in \mathcal{A}$, $p_i = M(a)$ and $(a, p_j) \in E''$. The weight of an edge $w(M(a), p_j)$ is defined as:

$$\begin{aligned} w(M(a), p_j) &= 0 && \text{if } a \text{ is indifferent between } M(a) \text{ and } p_j \\ &= -1 && \text{if } a \text{ prefers } M(a) \text{ to } p_j \\ &= +1 && \text{if } a \text{ prefers } p_j \text{ to } M(a). \end{aligned}$$

The graph $G_M = (\mathcal{P}, E_M)$ can be easily constructed in $O(\sqrt{n}m)$ time using Algorithm 2.1.

Consider a vertex p in G_M. A post p is a *sink* vertex in G_M if and only if p is unmatched by M in G. This follows from observing that $M(p)$, that is, the agent matched to p by M, has degree at least 2 in the graph G'. Further, any agent continues to have degree at least 2 in the graph G''. We refer the reader to the full version [11] for a detailed proof. Let $\mathcal{X}$ be a maximal weakly connected component of G_M. Call $\mathcal{X}$ a *sink component* if $\mathcal{X}$ contains one or more sink vertices, otherwise call $\mathcal{X}$ a *non-sink component*.

For a path T (resp. cycle C) in G_M, the weight of the path $w(T)$ (resp. $w(C)$) is the sum of the weights on the edges in T (resp. C). (Whenever we refer to paths and cycles in G_M we imply directed paths and directed cycles respectively.) A path $T = \langle p_1, \ldots, p_k \rangle$ in G_M is called a *switching path* if T ends in a sink vertex and $w(T) = 0$. Similarly, a cycle $C = \langle p_1, \ldots, p_k, p_1 \rangle$ in G_M is called a *switching cycle* if $w(C) = 0$. Let $\mathcal{A}_T = \{a_i : M(p_i) = a_i, \text{ for } i = 1 \ldots k\ \}$ and denote by $M' = M \cdot T$ the matching obtained by *applying* the switching path to M, that is, for $a_i \in \mathcal{A}_T$, $M'(a_i) = p_{i+1}$ whereas for $a \notin \mathcal{A}_T$, $M'(a) = M(a)$. Similarly, for a switching cycle C, define $\mathcal{A}_C = \{a_i : M(p_i) = a_i, \text{for } i = 1 \ldots k\ \}$ and denote by $M' = M \cdot C$ the matching obtained by *applying* the switching cycle to M, that is, for $a_i \in \mathcal{A}_C$, $M'(a_i) = p_{i+1 \mod k}$ whereas for $a \notin \mathcal{A}_C$, $M'(a) = M(a)$.

▶ **Example 6.**

Consider an instance G where $\mathcal{A} = \{a_1, \ldots, a_7\}$ and $\mathcal{P} = \{p_1, \ldots, p_9\}$. The preference lists of the agents are shown in Figure 1(a). The preference lists can be read as follows: agent a_1 ranks posts p_1, p_2, p_3 as her rank-1, rank-2 and rank-3 posts respectively and the two posts p_6 and p_7 are tied as her rank-4 posts. For every agent a, the posts which are bold denote the set $f(a)$, whereas the posts which are underlined denote the set $s(a)$. The instance G admits a popular matching; M and M' shown below are both popular in G.

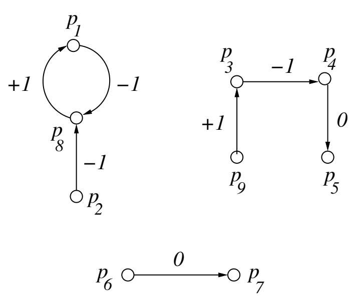
$$M = \{(a_1, p_6), (a_2, p_1), (a_3, p_8), (a_4, p_2), (a_5, p_3), (a_6, p_9), (a_7, p_4)\} \tag{1}$$

$$M' = \{(a_1, p_6), (a_2, p_1), (a_3, p_8), (a_4, p_2), (a_5, p_4), (a_6, p_3), (a_7, p_5)\} \tag{2}$$

Figure 1(b) shows the switching graph G_M with respect to the popular matching M. We note that the edges (a_4, p_3) and (a_1, p_1) get deleted in Step 4 and Step 9 of Algorithm 2.1, respectively. Hence the switching graph G_M does not have the edges $(M(a_4) = p_2, p_3)$ and $(M(a_1) = p_6, p_1)$ respectively. Consider the switching path $T = \langle p_9, p_3, p_4, p_5 \rangle$ in G_M. By *applying* T to M we get $M' = M \cdot T$ (see Equation (2)) which is also popular in G.

a_1 :	$\mathbf{p_1}$	p_2	p_3	$(\underline{p_6}, \underline{p_7})$
a_2 :	$\mathbf{p_1}$	p_2	$\underline{p_8}$	
a_3 :	$\mathbf{p_1}$	$\underline{p_8}$		
a_4 :	$(\mathbf{p_2}, \mathbf{p_3})$	p_1	$\underline{p_8}$	
a_5 :	$\mathbf{p_3}$	$(p_2, \underline{p_4})$		
a_6 :	$\mathbf{p_3}$	$\underline{p_9}$	p_1	
a_7 :	$(\underline{\mathbf{p_4}}, \underline{\mathbf{p_5}})$	p_1		

(a)



(b)

Figure 1 (a) Preference lists of agents $\{a_1, \ldots, a_7\}$. The posts which are bold denote $f(a)$ and the posts which are underlined denote $s(a)$. (b) Switching graph G_M with respect to the popular matching M in G.

3.1 Some useful properties

In this section we state some useful properties of the switching graph G_M (refer [11] for proof of correctness of these properties). Recall that the vertices $\mathcal{A} \cup \mathcal{P}$ are partitioned as $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$ w.r.t. a maximum matching M_1 in G_1 (see Step 3 of Algorithm 2.1). Further, the vertices $\mathcal{A} \cup \mathcal{P}$ are partitioned as $\mathcal{O}_2, \mathcal{E}_2, \mathcal{U}_2$ w.r.t. a popular matching M in G' (see Step 8 of Algorithm 2.1).

▶ Property 7. All sink vertices of G_M belong to the set $\mathcal{E}_1$.

▶ Property 8. Every post p belonging to a sink component has a path to a sink and hence belongs to the set $\mathcal{E}_2$. Every post belonging to a non-sink component belongs to the set $\mathcal{U}_2$.

▶ Property 9. For an edge (p_i, p_j) in G_M, the weight $w(p_i, p_j)$ is determined by the partition p_i and p_j belong to when vertices are partitioned as $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$. The weight $w(p_i, p_j)$ can be determined using Table 1.

Table 1 Table shows $w(p_i, p_j)$ for an edge (p_i, p_j) in G_M. The weight is determined by the partition of vertices as $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$. The $\times$ denotes that such an edge is not present in G_M.

p_i \ p_j	$\mathcal{O}_1$	$\mathcal{E}_1$	$\mathcal{U}_1$
$\mathcal{O}_1$	0	-1	$\times$
$\mathcal{E}_1$	$+1$	0	$\times$
$\mathcal{U}_1$	$\times$	-1	0

▶ Property 10. Every path T in G_M has $w(T) \in \{-1, 0, +1\}$. Every cycle C in G_M has $w(C) = 0$. There exists no path T in G_M ending in a sink vertex with $w(T) = +1$.

▶ Property 11. For any switching path T (or switching cycle C) in G_M, the matching $M' = M \cdot T$ ($M' = M \cdot C$ resp.) is a popular matching in G. Every popular matching M' in G can be obtained from M by applying to M one or more vertex disjoint switching paths and switching cycles in each of a subset of sink components of G_M together with one or more vertex disjoint switching cycles in each of a subset of the non-sink components of G_M.

Recall the definition of $choices(a)$ for an agent as given by Definition 5. It is easy to see that for any $a \in \mathcal{A}$, $choices(a) \subseteq \{f(a) \cup s(a)\}$. Further, if M is a popular matching in G, then $M(a) \in choices(a)$. We now define the notion of a *tight-pair*, that is, a set of agents $\mathcal{A}_1$ and a set of posts $\mathcal{P}_1$ with $|\mathcal{A}_1| = |\mathcal{P}_1|$. Further, for every $a \in \mathcal{A}_1$ we have $choices(a) \subseteq \mathcal{P}_1$. We show that a tight-pair exists whenever there is a non-sink component in the switching graph G_M.

▶ **Lemma 12.** *Let $\mathcal{Y}$ be a non-sink component in G_M and $q \in \mathcal{Y}$. Let,*

$$\mathcal{P}_q = \{q\} \cup \{p : \ q \text{ has a path to } p \text{ in } G_M \ \}.$$

Then there exists a set of agents $\mathcal{A}_q$ such that (i) $|\mathcal{A}_q| = |\mathcal{P}_q|$, and (ii) for every $a \in \mathcal{A}_q$, $choices(a) \subseteq \mathcal{P}_q$.

Proof. Let $\mathcal{A}_q = \{a : a = M(p) \text{ and } p \in \mathcal{P}_q\}$. Since every $p \in \mathcal{P}_q$ is matched, we note that $|\mathcal{A}_q| = |\mathcal{P}_q|$. For any $a \in \mathcal{A}_q$, we have $M(a) \in \mathcal{P}_q$ and note that $M(a) \in choices(a)$. Further, note that, for every $p' \in choices(a) \setminus \{M(a)\}$, we have an edge $(M(a), p')$ in G_M. Thus, every such p' also belongs to $\mathcal{P}_q$. This proves that for every $a \in \mathcal{A}_q$, $choices(a) \subseteq \mathcal{P}_q$. ◀

3.2 Generating popular pairs and counting popular matchings

Let $G = (\mathcal{A} \cup \mathcal{P}, E)$ be an instance of the popular matchings problem. Define

$$PopPairs = \{(a, p) \in E : \ M \text{ is a popular matching in } G \text{ and } M(a) = p\}. \quad (3)$$

Using the switching graph defined in the previous section, it is easy to compute the set $PopPairs$ in G. Let G_M be the switching graph with respect to a popular matching M in G. From *Property 11* we can conclude that an edge $e = (a, p)$ is a popular pair if and only if (i) $e \in M$ or, (ii) the edge $(M(a), p)$ belongs to some switching path in G_M or, (iii) the edge $(M(a), p)$ belongs to some switching cycle in G_M.

We note that edges satisfying condition (i) can be marked in $O(\sqrt{n}m)$ time using Algorithm 2.1 and edges satisfying conditions (ii) or (iii) can be marked in linear time in the size of the switching graph. Thus, we conclude the following theorem (see [11] for proof).

▶ **Theorem 13.** *The set of popular pairs for an instance $G = (\mathcal{A} \cup \mathcal{P}, E)$ of the popular matchings problem with ties can be computed in $O(\sqrt{n}m)$ time.*

We now show that given an instance of the popular matchings problem with ties, the problem of counting the number of popular matchings is #P-Complete. The result readily follows by (i) reducing the problem of computing the number of perfect matchings in 3-regular bipartite graphs to the popular matchings problem, and (ii) the fact that k-regular bipartite graphs admit a perfect matching.

▶ **Theorem 14.** *Given an instance $G = (\mathcal{A} \cup \mathcal{P}, E)$ of the popular matchings problem with ties, counting the total number of popular matchings in G is #P-Complete.*

4 Cheating strategies – preliminaries

In this section we set up the notation useful in formulating our cheating strategies. We begin by partitioning the set of agents $\mathcal{A}$ depending on the posts that a particular agent gets matched to when each agent is truthful, that is, in the instance G.

$$\begin{aligned} \mathcal{A}_f &= \{a : \text{every popular matching in } G \text{ matches } a \text{ to one of her rank-1 posts}\} \\ \mathcal{A}_s &= \{a : \text{every popular matching in } G \text{ matches } a \text{ to one of her non-rank-1 posts}\} \\ \mathcal{A}_{f/s} &= \mathcal{A} \setminus (\ \mathcal{A}_f \cup \ \mathcal{A}_s\ ). \end{aligned}$$

The set $\mathcal{A}_{f/s}$ denotes the set of agents a such that a gets matched to one of her rank-1 posts in some popular matching in G, whereas to one of her non-rank-1 posts in some other popular matching in G. It is easy to see that the above partition can be readily obtained once we have the set of popular pairs $PopPairs$ (defined by Equation (3)).

Let a_1 be the sole manipulative agent who is aware of the true preference lists of all other agents. Let $\mathcal{L} = P_1, P_2, \ldots, P_t, \ldots, P_l$ denote the true preference list of a_1 where P_i denotes the set of i-th rank posts of a_1. Since we will be working with another instance H obtained by falsifying the preference list of a_1, we now qualify the sets $f(a)$ and $s(a)$ for every agent with the instance under consideration. For an agent a, let $f_G(a)$ and $s_G(a)$ denote sets $f(a)$ and $s(a)$ respectively for an agent a in G. We note that $f_G(a_1) = P_1$. Assume that $s_G(a_1) \subseteq P_t$ is the set of t-th ranked posts of a_1, where $t > 1$.

Recall the strategy – *better always* defined for a single manipulative agent. If agent $a_1 \in \mathcal{A}_f$, then she does not have any incentive to manipulate her preference list. Thus, in this case we are done and $\mathcal{L}$ is her optimal strategy. We therefore focus on $a_1 \in \mathcal{A}_s \cup \mathcal{A}_{f/s}$. Let H denote the instance obtained by falsifying the preference list of a_1 alone.

- If $a_1 \in \mathcal{A}_s$, then in order to get *better always* her goal is to force at least some popular matching in H to match her to a post which she strictly prefers to her t-th ranked post.
- If $a_1 \in \mathcal{A}_{f/s}$, then in order to get *better always* her goal is to force every popular matching in H to match her to one of her true rank-1 posts.

Denote by $H \succ G$ with respect to a_1 if agent a_1 is *better always* in H. It is instructive to consider examples in order to get intuition regarding the cheating strategies.

▶ **Example 15.**

Consider the instance G as shown in Figure 1(a) and let a_5 be the manipulative agent. It can be seen that $a_5 \in \mathcal{A}_{f/s}$ in G. Now consider the instance H where a_5 alone falsifies her preference list. The preference list of a_5 in H is a strict list of length two and contains p_3 as her rank-1 post and p_8 as her rank-2 post. It is easy to verify that every popular matching in H matches a_5 to p_3 which is her true rank-1 post. The idea for an $\mathcal{A}_{f/s}$ agent a is to choose a post in $s_H(a)$ (here p_8) to which a can never be matched in a popular matching in H. We will show that such a post can be chosen whenever there exists a non-sink component in the switching graph and therefore a tight-pair (in this case $\mathcal{P}_1 = \{p_8, p_1\}$ and $\mathcal{A}_1 = \{a_2, a_3\}$).

▶ **Example 16.**

Consider the instance G shown in Figure 1(a) and let a_1 be the manipulative agent. Every popular matching in G matches a_1 to either p_6 or p_7 and therefore $a_1 \in \mathcal{A}_s$. Let H denote the instance where a_1 falsifies her preference list. The preference list of a_5 in H is a strict list of length two and contains p_3 as her rank-1 post and p_8 as her rank-2 post. It can be verified that every popular matching in H matches a_1 to p_3. The intuition here is that, a post to which a_1 wishes to get matched (here p_3), should have a path to an unmatched post or roughly speaking, belong to a sink component of G_M. We also choose a post in $s_H(a_1)$ (in this case p_8) to which a_1 can never get matched in any popular matching in H.

However, in this example, this is not the best that a_1 can get by falsifying. Let a_1 falsify her preference list to rank p_2 as her rank-1 post and p_8 as her rank-2 post. Consider the matching $M'' = \{(a_1, p_2), (a_2, p_1), (a_3, p_8), (a_4, p_3), (a_5, p_4), (a_6, p_9), (a_7, p_5)\}$ in H. It can be verified that M'' is popular in H and in fact every popular matching in H matches a_1 to p_2. However, our intuition that p_2 should belong to a sink component does not hold. This is because the edge (a_4, p_3) which got deleted in Step 4 of Algorithm 2.1 is being used after a_1 falsifies her preference list. In order to deal with such cases we will work with a modified instance as defined in Section 4.3.

We now formalize these intuitions in the rest of the section. In the interest of space we omit proof details and refer the reader to the full version [11].

4.1 $s(a)$ for other agents remains unchanged

Let H denote the instance obtained by falsifying the preference list of a_1 alone. Since the rest of the agents are truthful, for every agent $a \in \mathcal{A} \setminus \{a_1\}$, we have $f_H(a) = f_G(a)$. However, since $s_H(a)$ depends on the rank-1 posts of the rest of the agents, it may be the case that when a_1 falsifies her preference list, $s_H(a) \neq s_G(a)$ for an agent $a \in \mathcal{A} \setminus \{a_1\}$. We claim that if a_1 falsifies her preference list only to improve the rank of the post that she gets matched to, the rest of the agents do not change their $s(a)$. Recall that by definition, $s_H(a)$ is the set of most preferred posts of a which are *even* in the graph H_1 (the graph H on rank-1 edges). The following theorem summarizes the discussion.

▶ **Theorem 17.** *Let H be an instance such that $H \succ G$ w.r.t. a_1. Then, (i) $(\mathcal{E}_1)_G \cap \mathcal{P} = (\mathcal{E}_1)_H \cap \mathcal{P}$ and hence $s_H(a) = s_G(a)$ for every $a \in \mathcal{A} \setminus \{a_1\}$ and, (ii) $(\mathcal{O}_1)_G \cap \mathcal{A} = (\mathcal{O}_1)_H \cap \mathcal{A}$.*

4.2 An $\mathcal{A}_s$ agent cannot get one of her true rank-1 posts

In this section we show that if $a_1 \in \mathcal{A}_s$, then by falsifying her preference list alone, she cannot get matched to one of her true rank-1 posts in any popular matching in H. We state the result as Theorem 18 which requires the following claims. Let M be a popular matching in G and G_M denote the corresponding switching graph.

(I) If $a_1 \in \mathcal{A}_s$, then every post $q \in f_G(a_1)$ belongs to a non-sink component of G_M. We further claim that the edge $(M(a_1), q)$ does not belong to any cycle in G_M.

(II) Since every $q \in f_G(a_1)$ belongs to a non-sink component of G_M, using Lemma 12, we can define a tight-pair $\mathcal{P}_q$ and $\mathcal{A}_q$ w.r.t. q. Here, $\mathcal{P}_q$ denotes the set of posts reachable from q in G_M whereas $\mathcal{A}_q$ denotes the set of agents matched to the posts in $\mathcal{P}_q$. We claim that the post $M(a_1)$ does not belong to $\mathcal{P}_q$ and hence a_1 does not belong to $\mathcal{A}_q$.

(III) From the definition of tight-pair, we know that $|\mathcal{A}_q| = |\mathcal{P}_q|$ and for each $a \in \mathcal{A}_q$, $\mathit{choices}_G(a) \subseteq \mathcal{P}_q$. However, we claim that the same pair of sets is *tight* in H, that is, for every $a \in \mathcal{A}_q$, $\mathit{choices}_H(a) \subseteq \mathcal{P}_q$.

Using the above facts we prove the following theorem.

▶ **Theorem 18.** *Let $a_1 \in \mathcal{A}_s$. Then by falsifying her preference list alone, she cannot get matched to a post $q \in f_G(a_1)$ in any popular matching in the falsified instance.*

Proof. For contradiction assume that there exists a falsified instance H such that in a popular matching M' of H, agent a_1 gets matched to $q \in f_G(a_1)$. By (I), the post q belongs to a non-sink component of G_M. Further by (III), there exists a set of agents $\mathcal{A}_q$ and a set of posts $\mathcal{P}_q$ such that $|\mathcal{A}_q| = |\mathcal{P}_q|$, $a_1 \notin \mathcal{A}_q$ and for every $a \in \mathcal{A}_q$, we have $\mathit{choices}_H(a) \subseteq \mathcal{P}_q$. Thus, if a_1 gets matched to q in M', then there is at least one agent $a' \in \mathcal{A}_q$ which does not have a post to be matched to in $\mathit{choices}_H(a')$. This contradicts the fact that M' is a popular matching in H. ◀

4.3 The modified instance $\tilde{G}$

As mentioned earlier, we need to define a modified instance, call it $\tilde{G}$ to develop our cheating strategies. Recall from Example 16 that an edge which gets deleted from the graph G' in Step 4 of Algorithm 2.1 can be used in a popular matching in a falsified instance. Thus, we define $\tilde{G}$ from the instance G which has the following properties: (i) every popular matching in G is a popular matching in $\tilde{G}$ and, (ii) any edge (a, p) that gets deleted in Step 4 of Algorithm 2.1 when run on $\tilde{G}$ also gets deleted in Step 4 when Algorithm 2.1 is run on H such that $H \succ G$ w.r.t. a_1. However, the definition of $\tilde{G}$ is independent of the agent a_1.

The graph $\tilde{G}$ is defined as follows: Let G_1 be the graph on rank-1 edges of G and let $\{q_1, \ldots, q_k\}$ be the set of *unreachable* posts in G_1. Let us add to the instance G a dummy agent b whose preference list is of length one and has all the *unreachable* posts in G_1 tied as her rank-1 posts. That is, the preference list of b can be written as $(q_1, \ldots, q_k)$. The set of posts as well as the preference lists of all the agents $a \in \mathcal{A}$ remain the same as in G. Formally, $\tilde{G} = (\tilde{\mathcal{A}} \cup \mathcal{P}, \tilde{E})$ where $\tilde{\mathcal{A}} = \mathcal{A} \cup \{b\}$ and $\tilde{E} = E \cup \{(b, q_1), \ldots, (b, q_k)\}$ and each (b, q_i) is a rank-1 edge. By the choice of preference list of b, we note that $f_{\tilde{G}}(b) = \{q_1, \ldots, q_k\}$ and $s_{\tilde{G}}(b) = \ell(b)$, the unique last-resort post that we add for convenience.

We note that even after the addition of agent b, a maximum matching M_1 in G_1 continues to be a maximum matching in $\tilde{G}_1$. However, with respect to the partition of vertices on rank-1 edges in $\tilde{G}$, every vertex is either *odd* or *even* in $\tilde{G}_1$. Further, we claim that the set of *even* posts in $\tilde{G}_1$ is the same as the set of *even* posts in G_1. Thus, we can state the following lemma.

▶ **Lemma 19.** *For every $a \in \mathcal{A}$, we have $s_{\tilde{G}}(a) = s_G(a)$.*

Now let M be a popular matching in G, then let $\tilde{M}$ denote the corresponding matching in $\tilde{G}$ such that for every $a \in \mathcal{A}$ we have $\tilde{M}(a) = M(a)$ and $\tilde{M}(b) = \ell(b)$, the unique last-resort post of b. Note that $\tilde{M}$ is a maximum matching on rank-1 edges in $\tilde{G}$ and for every $a \in \mathcal{A}$, we have $\tilde{M}(a) \in \{f_{\tilde{G}}(a) \cup s_{\tilde{G}}(a)\}$. Finally, $\tilde{M}(b) \in s_{\tilde{G}}(b)$ since $s_{\tilde{G}}(b) = \{\ell(b)\}$. It is clear that $\tilde{M}$ satisfies both the properties of Theorem 3 and therefore is a popular matching in $\tilde{G}$. We can now construct the switching graph $\tilde{G}_{\tilde{M}}$ w.r.t. $\tilde{M}$ in $\tilde{G}$. With the definition of $\tilde{G}$ in place, we can now state the following lemmas.

▶ **Lemma 20.** *Let (a, p) be an edge which gets deleted in Step 4 of Algorithm 2.1 run on $\tilde{G}$. Then (a, p) gets deleted in Step 4 when Algorithm 2.1 is run on H where $H \succ G$ w.r.t. a_1.*

▶ **Lemma 21.** *Let $a \in \mathcal{A} \setminus \{a_1\}$ such that $\tilde{M}(a)$ belongs to a non-sink component of $\tilde{G}_{\tilde{M}}$. Let H be an instance such that $H \succ G$ w.r.t. a_1. Then $choices_H(a) \subseteq choices_{\tilde{G}}(a)$.*

5 Cheating strategies

In this section we develop an efficient characterization of the conditions under which a_1 can falsify her preference list. We formulate the strategy of a_1 depending on whether $a_1 \in \mathcal{A}_s$ or $a_1 \in \mathcal{A}_{f/s}$. Throughout, we assume that the true preference list of a_1 is denoted by $\mathcal{L} = P_1, \ldots, P_t, \ldots, P_l$ where P_i denotes the set of i-th ranked posts of a_1. Further, $f_G(a_1) = P_1$ and $s_G(a_1) \subseteq P_t$. We will use the modified instance $\tilde{G}$ to formulate our strategies.

5.1 $\mathcal{A}_s$ agent

Let $a_1 \in \mathcal{A}_s$ and let M be any popular matching in G and $\tilde{M}$ denote the corresponding popular matching in $\tilde{G}$ which matches b to $\ell(b)$. It follows from the definition of $\mathcal{A}_s$ that, $M(a_1) = \tilde{M}(a_1) \in s_G(a_1)$ and therefore, $M(a_1) \in P_t$. We first characterize whether a_1 can get *better always* using the graph $\tilde{G}$ and the switching graph $\tilde{G}_{\tilde{M}}$.

Our cheating strategy for a_1 (as shown in Figure 2) is simple: it checks if any of a_1's i-th ranked posts $p \in P_i$ where $i = 2 \ldots t-1$, either belongs to a sink component in $\tilde{G}_{\tilde{M}}$ or has a path to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$. If there exists such a post p, our strategy ensures that every popular matching in the falsified instance H matches a_1 to p. We denote by $\mathcal{L}_f$ the falsified preference list of a_1. We now state the main theorem in this section.

1. For $i = 2 \ldots t-1$ check if there exists a post $p \in P_i$ in a_1's preference list such that
   (a) p belongs to a sink component in $\tilde{G}_{\tilde{M}}$ or,
   (b) p has a path to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$.
2. If no post satisfies (a) or (b) above, then true preference list $\mathcal{L}$ is optimal for a_1.
3. Else let p denote the most preferred post of a_1 satisfying one of the above two properties. Set post p as a_1's rank-1 post in the falsified preference list.
4. To obtain the rank-2 post for a_1, let a_2 be some agent such that $\tilde{M}(a_2) \in f_G(a_1)$. Let $p' \in s_G(a_2)$. Set p' as the rank-2 post of a_1 in the falsified instance. $\mathcal{L}_f = p, p'$.

Figure 2 Cheating strategy for $a_1 \in \mathcal{A}_s$.

▶ **Theorem 22.** *Let $a \in \mathcal{A}_s$. Then there exists a cheating strategy for a_1 to get* better always *if and only if there exists a post p ranked $2 \ldots t-1$ on a_1's preference list satisfying either*

(a) p belongs to a sink component in $\tilde{G}_{\tilde{M}}$ or,
(b) p has a path to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$.

Proof. (Sketch) Assume that a post p satisfying one of the above two properties exists. Let $\mathcal{L}_f = p, p'$ be the falsified preference list for a_1 as returned by Step 4 of Figure 2. Let H denote the instance where a_1 submits $\mathcal{L}_f$ and the rest of the agents are truthful. We show that every popular matching in H matches a_1 to p. The idea is to use the path starting at p which ends in an unmatched vertex and construct another matching which matches a_1 to p. Further, to show that every popular matching matches a_1 to p, we use the tight-sets $\mathcal{A}_{p'}$ and $\mathcal{P}_{p'}$. Finally, to show that our strategy is optimal, from Theorem 18, we know that a_1 cannot get matched to any of her true rank-1 posts. Now let q be a post which does not satisfy any of the conditions in Theorem 22 and is more preferred by a_1 than the post that it got matched to after running the strategy in Figure 2. We show the existence of tight-sets $\mathcal{A}_q$ and $\mathcal{P}_q$ for such a post q which implies that no popular matching in the falsified instance can match a_1 to q. Refer [11] for a full proof. ◀

5.2 $\mathcal{A}_{f/s}$ agent

Let $a_1 \in \mathcal{A}_{f/s}$ when she submits her true preference list. In order to get *better always*, the goal of a_1 is to falsify her preference list such that every popular matching in the falsified instance H matches a_1 to posts in P_1.

Let M be a popular matching in G such that $M(a_1) = p$ and $p \in f_G(a_1)$. Let $\tilde{M}$ denote the corresponding popular matching in $\tilde{G}$ which matches b to $\ell(b)$. Consider the switching graph $\tilde{G}_{\tilde{M}}$. Our strategy for a_1 to get better always (as described in Figure 3) is to search for an *even* post p' in G_1 which belongs to a non-sink component of $\tilde{G}_{\tilde{M}}$ and further the post p' does not have a path T to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$ where $w(T) = +1$.

1. For every $p' \in (\mathcal{E}_1)_G \cap \mathcal{P}$ check if
   (a) p' belongs to a non-sink component, say $\mathcal{Y}_1$, of $\tilde{G}_{\tilde{M}}$ and,
   (b) p' does not have a path T to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$ such that $w(T) = +1$.
2. If no post satisfies both properties, declare true preference list $\mathcal{L}$ is optimal for a_1.
3. Else set $M(a_1) = p$ and p' as the rank-1 and rank-2 posts respectively in the falsified preference list of a_1. $\mathcal{L}_f = p, p'$.

Figure 3 Cheating strategy for $a_1 \in \mathcal{A}_{f/s}$ to get better always.

▶ **Theorem 23.** *Let $a_1 \in \mathcal{A}_{f/s}$. There exists a cheating strategy for a_1 to get* better always *if and only if there exists a post p' in $(\mathcal{E}_1)_G$ satisfying the following two properties*
(a) p' belongs to a non-sink component, say $\mathcal{Y}_1$, of $\tilde{G}_{\tilde{M}}$, and
(b) there exists no path T from p' to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$ such that $w(T) = +1$.

Proof. (Sketch) Assume that a post p' satisfying the above two properties exists. Then by falsifying her preference list as $\mathcal{L}_f = p, p'$, agent a_1 can force every popular matching in H to match a_1 to p. The proof uses the existence of the tight-pair $\mathcal{A}_{p'}, \mathcal{P}_{p'}$. On the other hand, assume that no such post exists and for the sake of contradiction, there exists an instance H such that every popular matching in H matches a_1 to a post in $f_G(a_1)$. In this case we show that there exists a popular matching M' in H such that $M'(a_1) \in s_H(a_1)$. Further, $s_H(a_1)$ cannot contain any of a_1's true rank-1 posts, therefore this contradicts the fact that every popular matching in H matches a_1 to one of her true rank-1 posts. ◀

Using Theorem 22 and Theorem 23 we conclude the following.

▶ **Theorem 24.** *The optimal falsified preference list for a single manipulative agent to get* better always *can be computed in* $O(\sqrt{n}m)$ *time if preference lists contain ties and in time* $O(m+n)$ *time if preference lists are all strict.*

Proof. The main steps of our strategy are (i) to compute the set of popular pairs, (ii) to construct the switching graph, (iii) run the algorithm given by Figure 2 or Figure 3 as appropriate for the single manipulative agent. We note that we use the modified graph $\tilde{G}$ for computing our strategies and let $\tilde{n}$ and $\tilde{m}$ denote the vertices and edges in $\tilde{G}$ respectively. Clearly, $\tilde{n} = n + 1$ and $\tilde{m} < m + n = O(m)$. Once the switching graph is constructed, we observe that the algorithms in Figure 2 and Figure 3 have checks which can be done in time which is linear in the size of the switching graph. Thus the steps (i) and (ii) defined above decide the complexity of our cheating strategy. In case of ties, we have shown that both the steps can be computed in $O(\sqrt{n}m)$ time. In case of strict lists, using the switching graph given by McDermid and Irving [10], both the steps can be computed in $O(m + n)$ time. Thus we have the desired result. ◀

Acknowledgment: The author is grateful to Prof. Vijaya Ramachandran for useful discussions on the problem and to an anonymous reviewer for the helpful comments.

References

1. D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. Popular Matchings. *SIAM Journal on Computing*, 37(4):1030–1045, 2007.
2. D. Gale and L. Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, 69:9–14, 1962.
3. P. Gärdenfors. Match making: Assignments based on bilateral preferences. *Behavioural Sciences*, 20:166–173, 1975.
4. C.-C. Huang. Cheating to Get Better Roommates in a Random Stable Matching. In *Proceedings of 24th Annual Symposium on Theoretical Aspects of Computer Science*, pages 453–464, 2007.
5. C.-C. Huang and T. Kavitha. Near-Popular Matchings in the Roommates Problem. In *Proceedings of the 19th Annual European Symposium on Algorithms*, pages 167–179, 2011.
6. T. Kavitha. Popularity vs maximum cardinality in the stable marriage setting. In *Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms*, pages 123–134, 2012.
7. T. Kavitha, J. Mestre, and M. Nasre. Popular Mixed Matchings. *Theoretical Computer Science*, 412(24):2679–2690, 2011.
8. M. Mahdian. Random Popular Matchings. In *Proceedings of 7th ACM Conference on Electronic Commerce*, pages 238–242, 2006.
9. R. M. McCutchen. The Least-Unpopularity-Factor and Least-Unpopularity-Margin Criteria for Matching Problems with One-Sided Preferences. In *Proceedings of the 15th Latin American Symposium on Theoretical Informatics*, pages 593–604, 2008.
10. E. McDermid and R. W. Irving. Popular matchings: structure and algorithms. *Journal of Combinatorial Optimization*, 22(3):339–358, 2011.
11. M. Nasre. Popular Matchings: Structure and Cheating Strategies. *CoRR*, abs/1301.0902, 2013.
12. W. R. Pulleyblank. *Handbook of Combinatorics (Vol. 1)*, chapter Matchings and Extensions, pages 179–232. MIT Press, Cambridge, MA, USA, 1995.
13. C.-P. Teo, J. Sethuraman, and W.-P. Tan. Gale-Shapley Stable Marriage Problem Revisited: Strategic Issues and Applications. *Management Science*, 47(9):1252–1267, 2001.