POPULAR MATCHINGS: STRUCTURE AND STRATEGIC ISSUES *

MEGHANA NASRE †

Abstract. We consider the strategic issues of 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, in the falsified instance (i) every popular matching matches a_1 to a post that is at least as good as the most-preferred post that she gets when she was truthful, and (ii) some popular matching matches a_1 to a post better than the most-preferred post p that she gets when she was truthful, assuming that p is not one of a_1's (true) most-preferred posts. We show that the optimal cheating strategy for a manipulative 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.

1. Introduction. We consider the strategic issues of 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 $M'(a)$.

Definition 1.1. *A matching M is* more popular than M' *if the number of agents that prefer M to M' is greater than the number of agents that prefer M' to 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

*A preliminary version of this work appeared in STACS 2013 [14]. This work was supported in part by CSE/14-15/824/NFIG/MEGA.

†Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, 600036 (meghana@cse.iitm.ac.in).

to posts using the criteria of popularity. Since an instance may admit more than one popular matching, we assume that the central authority chooses an arbitrary popular matching. Note that in distinct popular matchings of an instance, an agent may get matched to different posts. 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. 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.

Let $\mathcal{P}_G(a_1) = \{q_1, q_2, \ldots, q_k\}$ denote the set of posts that a_1 gets matched to in distinct popular matchings of G. Let $\mathcal{P}_H(a_1) = \{q'_1, q'_2, \ldots, q'_t\}$ denote the set of posts that a_1 gets matched to in distinct popular matchings of H. In addition, assume that the posts in these sets are ordered with respect to a_1's preference. That is, for each $1 \leq i \leq k-1$, agent a_1 prefers q_i at least as much as q_{i+1}, denoted by $q_i \succeq_{a_1} q_{i+1}$. Similarly, assume for $1 \leq j \leq t-1$, we have $q'_j \succeq_{a_1} q'_{j+1}$. Agent a_1 wishes to falsify her preference list to ensure that (i) Every popular matching of H matches her to a post that she prefers at least as much as q_1. That is, for $1 \leq j \leq t$, $q'_j \succeq_{a_1} q_1$ and, (ii) Assuming that q_1 is not a_1's top choice post, there exists some popular matching in H that matches her to a post which she *strictly* prefers to q_1. That is, $\exists q'_j \in \mathcal{P}_H(a_1)$ such that $q'_j \succ_{a_1} q_1$. We term this strategy of a_1 as '*better always*' strategy.

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. We remark that in order to develop the better always strategy, we need to efficiently compute for an agent a, all the posts that she can get matched to in distinct popular matchings of a given instance. One of our contributions involves computing the *popular pairs* in an instance.

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. The switching graph characterization was studied for the case of strict lists by McDermid and Irving [13] and such a characterization was not known for the case of ties. The switching graph characterization is of independent interest and answers a part of the open questions in [13]. 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. This is in contrast to the case when preferences are strict, where McDermid and Irving [13] gave a linear time algorithm for the same problem.

1.2. Related work. The work in this paper is motivated by the work of Teo et al. [17] where they study the strategic issues of the stable marriage problem [3]. 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. [17] 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. Moreover, she is aware of the true preference lists of all the other women. Teo et al. [17] compute an optimal cheating strategy for a single woman under this model. Huang [5] studies the strategic issues of the stable room-mates problem [3] 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 [4] in the context of the stable marriage [3]. 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 [6, 8, 9, 11, 12]. However, to the best of our knowledge none of them is motivated by the strategic issues of the popular matchings problem.

Organization of the paper: The rest of the paper is organized as follows. In Section 2 we review the background of the popular matchings problem. In Section 3 we develop the switching graph characterization for the popular matchings problem with ties. In Section 4 we give some intuition and prove useful lemmas for computing our cheating strategies. In Section 5 we formulate the cheating strategies for a manipulative agent. We conclude the paper with some open questions in Section 6.

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 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. 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; its proof can be found in [15] or [7].

LEMMA 2.1 ([15]). *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 2.2 ([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) that is maximum in G_1 and that 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 the next section. Abraham et al. [1] also showed a simpler characterization for the popular matchings in case of strict lists which gives an $O(m + n)$ time algorithm to return a popular matching if one exists.

We now elaborate on the graphs G' and G'' constructed during the execution of Algorithm 2.1. 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' between a node in $\mathcal{O}_1$ and a node in $\mathcal{O}_1 \cup \mathcal{U}_1$. We use the following notation for the sake of brevity to refer to edges that have end points lying in particular partitions. We refer to an edge as an $\mathcal{O}_1\mathcal{U}_1$ edge if the edge has one end point in $\mathcal{O}_1$ and another end point in $\mathcal{U}_1$. Note that the $\mathcal{O}_1\mathcal{O}_1$ edges and the $\mathcal{O}_1\mathcal{U}_1$ edges deleted in Step 4 cannot be present in any maximum matching of G_1 (by Lemma 2.1(c)). In addition, observe that the edges deleted in Step 4 are only rank-1 edges, since posts in $\mathcal{O}_1 \cup \mathcal{U}_1$ have only rank-1 edges incident on them.

Our idea is to extend this and delete the edges from G' that cannot be present in any popular matching of 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', it is easy to see that M cannot contain edges of the form $\mathcal{O}_2\mathcal{O}_2$ and $\mathcal{O}_2\mathcal{U}_2$ (by Lemma 2.1(c)). However, note that 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 the graph G'. Therefore, in Step 9 of Algorithm 2.1 we delete from G' any edge between a node in $\mathcal{O}_2$ and a node in $\mathcal{U}_2$. Let G'' denote the resulting graph.

To illustrate these definitions, we use an example instance. We use it as a running example throughout the paper.

EXAMPLE 2.3. 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 2.1(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. We omit explicitly listing $\ell(a)$ at the end of each agent a's preference list. 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 graph G_1 on rank-1 edges of the instance along with a maximum matching $M_1 = \{(a_1, p_1), (a_4, p_2), (a_5, p_3), (a_7, p_4)\}$ is as shown in Figure 2.1(b). Finally, Figure 2.1(c) shows the graph G' and a popular matching $M = \{(a_1, p_6), (a_2, p_1), (a_3, p_8), (a_4, p_2), (a_5, p_3), (a_6, p_9), (a_7, p_4)\}$ in G. 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. The graph G'' thus formed is shown in Figure 3.1(a) which will be used to construct the switching graph in the next section.

Using the following simple observations we state Claim 2.4.

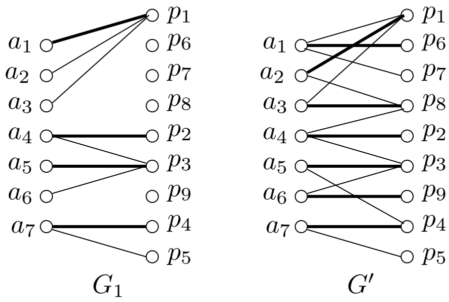
(i) Step 4 of Algorithm 2.1 deletes $\mathcal{O}_1\mathcal{O}_1$ and $\mathcal{O}_1\mathcal{U}_1$ edges,

(ii) Step 9 of Algorithm 2.1 deletes $\mathcal{O}_2\mathcal{U}_2$ edges, and

(iii) $\mathcal{P} \cap \mathcal{O}_2 = \emptyset$.

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) (c)

FIG. 2.1. *(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) The graph* G_1 *with bold edges denoting matching* M_1*. (c) The graph* G' *with bold edges denoting a popular matching* M*.*

CLAIM 2.4. *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 An $O(\sqrt{n}m)$-time algorithm for the popular matching problem from [1] (Steps 7–11 are added by us).

Input: $G = (\mathcal{A} \cup \mathcal{P}, E)$.
1: Construct $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$. Let M_1 be a maximum matching in G_1.
2: Partition $\mathcal{A} \cup \mathcal{P}$ as $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$ with respect to M_1 in G_1.
3: Construct $G' = (\mathcal{A} \cup \mathcal{P}, E')$, where $E' = \{(a, p) : a \in \mathcal{A}$ and $p \in f(a) \cup s(a)\}$.
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: If M is $\mathcal{A}$-complete then M is popular in G, else 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 2.5. *For an agent* a*, let* $choices(a)$ *be the set of posts* p *such that* (a, p) *is an edge in* G''*.*

It is easy to see that for any $a \in \mathcal{A}$, $choices(a) \subseteq f(a) \cup s(a)$. Using the above definition and the fact in Step 9 of Algorithm 2.1 we delete only those edges that do not belong to any popular matching in G, we make the following claim.

CLAIM 2.6. *Let* M *be any popular matching in* G *and* a *be any agent. The matching* M *is contained in* G'' *and* $M(a) \in choices(a)$*.*

We now show using the following lemma that the cardinality of $choices(a)$ for any agent a is no less than two. This fact will be used by the switching graph defined in the next section.

LEMMA 2.7. *For any agent* $a \in \mathcal{A}$*,* $|choices(a)| \geq 2$*.*

Proof. We first show that every agent has degree at least 2 in the graph G' constructed in Step 3 of Algorithm 2.1. Recall that G' denotes the graph in which every agent a has edges to $f(a) \cup s(a)$. Let $a \in \mathcal{A}$. If $|f(a)| \geq 2$, we are done. Else let $f(a)$ contain exactly one post and let $f(a) = \{p\}$. In this case, it is easy to see that $p \in \mathcal{O}_1 \cup \mathcal{U}_1$. Thus, by the definition of $s(a)$, we conclude that $s(a) \cap f(a) = \emptyset$. Moreover, for every agent a, $s(a)$ is well-defined due to the introduction of $\ell(a)$. This shows that, even when $|f(a)| = 1$, agent a has at least two edges incident on it in the graph G', one edge to $p \in f(a)$ and, one or more edges to posts in $s(a)$.

Next, observe that the graph G'' differs from G' because edges may get deleted in Step 4 and Step 9 of Algorithm 2.1. We now show that even after the deletions, every agent continues to have degree at least 2 in the resultant graph G''. This will prove that for any $a \in \mathcal{A}$, *choices*$(a) \geq 2$. Let M_1 be a maximum matching in G_1 and M be a popular matching in G.

Consider the edges deleted in Step 4. Recall that we never delete matched edges, since they are either of the form $\mathcal{OE}$ or $\mathcal{UU}$. We consider three cases depending on the partition in which a belongs when vertices are partitioned as $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$.

- Let $a \in \mathcal{E}_1$: By Claim 2.4, no edge incident on a gets deleted in Step 4, so we are done.
- Let $a \in \mathcal{O}_1$: This implies that a is matched in M_1 and let $M_1(a) = p$. In addition, there exists an odd length alternating path $T = \langle p_0, a_1, p_1, \ldots, a_k, p_k, a\rangle$ w.r.t. M_1 starting at an unmatched post p_0 leading to the agent a. The path T is incident on a with an unmatched edge (p_k, a). Note that this unmatched edge (p_k, a) is of the form $\mathcal{E}_1\mathcal{O}_1$ and hence does not get deleted. Thus, the two neighbors p_k and p of the agent a ensure that the degree of a is at least 2 even after some edges incident on a may get deleted in Step 4.
- Finally, let $a \in \mathcal{U}_1$: This implies that a is matched in M_1 and let $M_1(a) = p$. Since $a \in \mathcal{U}_1$, it implies that every post $p' \in f(a)$ is such that $p' \in \mathcal{O}_1 \cup \mathcal{U}_1$. Since $s(a)$ is the set of most preferred posts belonging to $\mathcal{E}_1$, we conclude that in this case, all posts in $s(a)$ are at rank greater than or equal to 2. Moreover, since Step 4 only deletes rank-1 edges, none of the edges from a to posts in $s(a)$ can get deleted in this step. Thus, counting the edge to p and one or more edges to posts in $s(a)$, we ensure that a has degree at least 2, even after some edges incident on a may get deleted in Step 4.

We now consider edges deleted in Step 9. Recall that $\mathcal{A} \cap \mathcal{E}_2 = \emptyset$. Consider $a \in \mathcal{O}_2$, then using the same argument as we used for $a \in \mathcal{O}_1$, we can prove that even after edges incident on a get deleted, degree of a remains at least 2. Finally, if $a \in \mathcal{U}_2$, then by Claim 2.4, no edge incident on a gets deleted in Step 9.

Thus, we have proved that in the graph G'', the degree of every agent is at least 2, which implies that *choices*$(a) \geq 2$, for every $a \in \mathcal{A}$. ∎

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 [13] 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 is determined by a popular matching M in G. In fact, a similar graph was defined even before that by Mahdian [11] (again for strict lists) to study the existence of popular matchings in random instances. McDermid and Irving showed that given a popular matching M in G, the switching graph G_M can be used to efficiently construct any other popular matching M' in G. They also exploited the switching graph to develop efficient algorithms for several problems such as computing popular pairs, and counting the number of popular matchings, to name a few. We use the notation and terminology from [13] to define the switching graph in case of ties.

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 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 $p_i \neq p_j$) in E_M 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:

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

It is easy to see that the graph $G_M = (\mathcal{P}, E_M)$ can be constructed in $O(\sqrt{n}m)$ time using Algorithm 2.1.

Consider a vertex p in G_M. Call p a sink vertex in G_M if the out-degree of p is zero in G_M. We will prove subsequently that sink vertices are posts that are unmatched in M. Let $\mathcal{X}$ be a connected component in the underlying undirected graph 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 $T = \langle p_0, p_1 \ldots, p_{k-1} \rangle$ in G_M is called a *switching path* if T ends in a sink vertex and $w(T) = 0$. Similarly, a cycle $C = \langle p_0, \ldots, p_{k-1}, p_0 \rangle$ in G_M is called a *switching cycle* if $w(C) = 0$. Let $\mathcal{A}_T = \{a_i : M(p_i) = a_i, \text{ for } i = 0 \ldots k-2\ \}$ 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 $\mathcal{A}_C = \{a_i : M(p_i) = a_i, \text{for } i = 0 \ldots k-1\ \}$ and denote by $M' = M \cdot C$ the matching obtained by *applying* the switching cycle to M, that is, for $a_i \in \mathcal{A}_C$, $M'(a_i) = p_{(i+1) \bmod k}$ whereas for $a \notin \mathcal{A}_C$, $M'(a) = M(a)$. Before we prove the structural properties of the switching graph, let us construct the switching graph for our example instance.

Example 3.1. Recall the instance in Figure 2.1(a) and the graph G'' constructed by Step 10 of Algorithm 2.1. Figure 3.1(a) shows the graph G'' for the instance with a popular matching $M = \{(a_1, p_6), (a_2, p_1), (a_3, p_8), (a_4, p_2), (a_5, p_3), (a_6, p_9), (a_7, p_4)\}$. Figure 3.1(b) shows the switching graph G_M with respect to M. 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 $M' = M \cdot T$ where $M' = \{(a_1, p_6), (a_2, p_1), (a_3, p_8), (a_4, p_2), (a_5, p_4), (a_6, p_3), (a_7, p_5)\}$. It is easy to verify that M' is also a popular matching in G.

3.1. Structural properties. In this section we prove some useful structural properties of the switching graph G_M. 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 2 of Algorithm 2.1). We show that this partition of vertices determines the weights on the edges of the switching graph. Additionally, 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). We show that this partition determines whether a post belongs to a sink component or a non-sink component.

Property 3.2. *A vertex p is a sink vertex of G_M iff p is unmatched in M. Furthermore, every sink vertex of G_M belongs to the set $\mathcal{E}_1$.*

Proof. We note that by the definition of the graph G_M, an unmatched post p does not have any out-going edge and hence is a sink vertex of G_M. To show that a sink vertex p is unmatched in M, it suffices to observe that for any $a \in \mathcal{A}$, we have

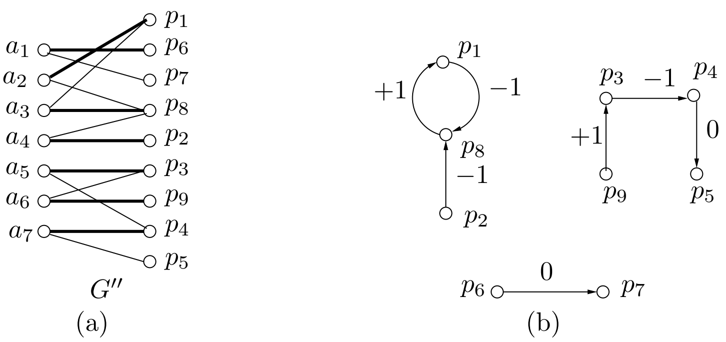


FIG. 3.1. *(a) The graph G'' with a popular matching M. (b) Switching graph G_M with respect to M in G.*

$|choices(a)| \geq 2$, by Lemma 2.7. Thus, every matched post will have out-degree at least one in G_M and hence only an unmatched post can be a sink vertex.

We now prove that every sink vertex belongs to $\mathcal{E}_1$. Assume for the sake of contradiction that a sink vertex p in G_M belongs to $\mathcal{O}_1 \cup \mathcal{U}_1$. Since p is a sink, it implies that p is unmatched in M. Moreover, since M is a popular matching, it implies that M is a maximum matching on rank-1 edges in G. However, every maximum matching on rank-1 edges of G matches every vertex in $\mathcal{O}_1 \cup \mathcal{U}_1$. Thus, if p is unmatched in M and $p \in \mathcal{O}_1 \cup \mathcal{U}_1$, it implies that M is not a maximum matching on rank-1 edges of G, a contradiction. ◻

PROPERTY 3.3. *A post p belongs to a sink component of G_M iff $p \in \mathcal{E}_2$. A post p belongs to a non-sink component of G_M iff $p \in \mathcal{U}_2$. Additionally, every post belonging to a sink component has a path to some sink vertex in G_M.*

Proof. We show that it suffices to prove the first statement. Let p be a post such that $p \in \mathcal{E}_2$. Then p is either unmatched in M or p has an even length alternating path starting at an unmatched vertex p' with respect to M in G'. If p is unmatched, then p is a sink vertex in G_M and hence we are done. Else let $\langle p = p_1, a_1, \dots, p_k, a_k, p_{k+1} = p' \rangle$ denote the alternating path and for every $1 \leq i \leq k$, we have $M(p_i) = a_i$. Note that every unmatched edge (a_i, p_{i+1}) is of the form $\mathcal{O}_2\mathcal{E}_2$ and hence none of these unmatched edges get deleted in Step 9 of Algorithm 2.1. Therefore, it is easy to see that the path $\langle p = p_1, p_2, \dots, p_{k+1} = p' \rangle$ is present in G_M and hence p belongs to the sink component that contains p'. This also implies that every $p \in \mathcal{E}_2$ has a directed path to some sink vertex in G_M.

To prove the other direction let $\mathcal{X}$ be a sink component in G_M and p' be a sink in $\mathcal{X}$. For the sake of contradiction let $p \in \mathcal{X}$ and $p \in \mathcal{U}_2$. Recall that $\mathcal{O}_2 \cap \mathcal{P} = \emptyset$. Now since p and p' lie in the same component, there is an (undirected) path between p and p' in the underlying undirected component of $\mathcal{X}$. Let $\langle p = p_1, p_2, \dots, p_k = p' \rangle$ denote this undirected path. Since $p_1 \in \mathcal{U}_2$ and $p_k \in \mathcal{E}_2$, it implies that there exists an $1 \leq i \leq k-1$ such that $p_i \in \mathcal{U}_2$ and $p_{i+1} \in \mathcal{E}_2$. Consider the two possible directions for the edge between p_i and p_{i+1} in G_M: (i) If the edge is directed from p_i to p_{i+1} in G_M, then we show that p_i has a directed path to some sink of $\mathcal{X}$. As proved earlier, p_{i+1} has a directed path T to some sink in $\mathcal{X}$. We now prefix the edge (p_i, p_{i+1}) to the path T to get a directed path from p_i to some sink. Such a directed path to a sink implies an even length alternating path with respect to M in G' from the sink to p_i. This contradicts the fact that $p_i \in \mathcal{U}_2$. (ii) Finally, if the edge is directed from p_{i+1} to p_i in G_M, then it implies that p_{i+1} is matched in M and let $M(p_{i+1}) = a_{i+1}$.

Since $p_{i+1} \in \mathcal{E}_2$ this implies that $a_{i+1} \in \mathcal{O}_2$. Thus the presence of the edge (p_{i+1}, p_i) in G_M implies that there is an $\mathcal{O}_2\mathcal{U}_2$ edge (a_{i+1}, p_i) in the graph G''. However, such an $\mathcal{O}_2\mathcal{U}_2$ edge should have been deleted by Step 9 of Algorithm 2.1. Hence such an edge cannot be present in G_M contradicting the fact that $p \in \mathcal{U}_2$. Thus, every post p belonging to a sink component belongs to the set $\mathcal{E}_2$.

The above proof immediately implies that a post p belongs to a non-sink component iff $p \in \mathcal{U}_2$. This finishes the proof of Property 3.3. ▯

PROPERTY 3.4. *For an edge (p_i, p_j) in G_M, the weight $w(p_i, p_j)$ is determined by which partition p_i and p_j belong to when vertices are partitioned as $\mathcal{O}_1, \mathcal{E}_1, \mathcal{U}_1$. That is, $w(p_i, p_j)$ can be determined using Table 3.1.*

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

TABLE 3.1

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.

Proof. To prove Property 3.4 we justify the entries in Table 3.1. Let (p_i, p_j) be an edge in G_M and let $M(p_i) = a$. The weight on the edge (p_i, p_j) is determined by the relative ranks of p_i and p_j in a's preference list. We note that a post $p \in \mathcal{O}_1 \cup \mathcal{U}_1$ has only rank-1 edges incident on it in the graph G'. Hence if $p_i \in \mathcal{O}_1 \cup \mathcal{U}_1$, then a treats p_i as her rank-1 post.

- $p_i \in \mathcal{O}_1$: a treats p_i as her rank-1 post and since posts in $\mathcal{O}_1$ remain matched to agents in $\mathcal{E}_1$, it implies that $a \in \mathcal{E}_1$.
  - $p_j \in \mathcal{O}_1$: a treats p_j as her rank-1 post, thus, $w(p_i, p_j) = 0$.
  - $p_j \in \mathcal{E}_1$: We show that a treats p_j as her non-rank-1 post and hence $w(p_i, p_j) = -1$. Assume for the sake of contradiction that a treats p_j as a rank-1 post. It implies that there is an $\mathcal{E}_1\mathcal{E}_1$ edge in the graph G_1, a contradiction (by part (c) Lemma 2.1).
  - $p_j \in \mathcal{U}_1$: We show that such an edge cannot exist in G_M. Recall that posts in $\mathcal{U}_1$ have only rank-1 edges incident on them, hence a treats p_j as her rank-1 post. This implies that there is an $\mathcal{E}_1\mathcal{U}_1$ edge in G_1, a contradiction (by part (c) of Lemma 2.1).
- $p_i \in \mathcal{E}_1$: Here we consider two cases:
  (i) a treats p_i as her rank-1 post: In this case, we note that $s(a) \subseteq f(a)$ and hence a has only rank-1 edges incident on it in the graph G' and all these edges are incident on posts which belong to $\mathcal{E}_1$. Thus the only case possible is, $p_j \in \mathcal{E}_1$ and $w(p_i, p_j) = 0$.
  (ii) a treats p_i as her non-rank-1 post: We first note that $a \in \mathcal{E}_1$ because agents in $\mathcal{O}_1 \cup \mathcal{U}_1$ remain matched along rank-1 edges in every popular matching. Consider the three different cases for p_j.
  - $p_j \in \mathcal{O}_1$: a treats p_j as her rank-1 post and hence $w(p_i, p_j) = +1$.
  - $p_j \in \mathcal{E}_1$: We show that a treats p_j as her non-rank-1 post and hence $w(p_i, p_j) = 0$. Assume for the sake of contradiction that a treats p_j as her rank-1 post. Then there exists an $\mathcal{E}_1\mathcal{E}_1$ edge in G_1 a contradiction (by part (c) of Lemma 2.1).

- $p_j \in \mathcal{U}_1$: We show that such an edge cannot exist in G_M. If such an edge exists there is an $\mathcal{E}_1\mathcal{U}_1$ edge in G_1 a contradiction (by part (c) of Lemma 2.1).

- $p_i \in \mathcal{U}_1$: a treats p_i as her rank-1 post and since posts in $\mathcal{U}_1$ remain matched along agents in $\mathcal{U}_1$, it implies that $a \in \mathcal{U}_1$.
  - $p_j \in \mathcal{O}_1$: a treats p_j as her rank-1 post however such an edge gets deleted as an $\mathcal{O}_1\mathcal{U}_1$ edge in Step 4 of Algorithm 2.1. Thus such an edge cannot be present in G_M.
  - $p_j \in \mathcal{E}_1$: We show that a treats p_j as her non-rank-1 post and hence $w(p_i, p_j) = -1$. Assume for the sake of contradiction that a treats p_j as a rank-1 post then, it implies that there is a $\mathcal{U}_1\mathcal{E}_1$ edge in the graph G_1, a contradiction (by part (c) of Lemma 2.1).
  - $p_j \in \mathcal{U}_1$: a treats p_j as her rank-1 post and therefore $w(p_i, p_j) = 0$.

This justifies all the entries of Table 3.1. ◻

PROPERTY 3.5. *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 $w(T) = +1$.*

Proof. It is easy to observe that if the edges have weights according to Table 3.1, then every path in G_M has weight belonging to $\{-1, 0, +1\}$. Furthermore, every cycle has to have weight 0. It remains to argue that in G_M there exists no path T of weight $+1$ which ends in a sink. For contradiction, assume that such a path exists in G_M and consider applying the path T to M to obtain the matching $M' = M \cdot T$. The number of agents that prefer M' to M is exactly one more than the number of agents that prefer M to M'. Thus M' is more popular than M, contradicting the fact that M was a popular matching. ◻

PROPERTY 3.6. *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 zero or more vertex disjoint switching paths and switching cycles in each of the of sink components of G_M together with zero or more vertex disjoint switching cycles in each of the non-sink components of G_M.*

Proof. We first show that for any switching path T, the matching obtained by applying T to M is popular in G. Let $T = \langle p_0, p_2, \ldots, p_{k-1} \rangle$ be a switching path in G_M with p_{k-1} unmatched in M and let $M' = M \cdot T$. Let $\mathcal{A}_T = \{a_i : M(p_i) = a_i,\ \text{for } i = 0 \ldots k-2\ \}$. We observe that for every $a_i \in \mathcal{A}_T$, $M'(a_i) \in f(a_i) \cup s(a_i)$ because edges of G_M are derived from a subset of graph $G' = (\mathcal{A} \cup \mathcal{P}, E')$ (refer to Algorithm 2.1). Furthermore, for any $a \notin \mathcal{A}_T$, $M'(a) = M(a)$. Finally, note that since $w(T) = 0$, for every agent that got demoted from her rank-1 post there exists a unique agent who got promoted to her rank-1 post in M'. Thus, M' is a maximum matching on rank-1 edges of G. It is therefore clear that M' satisfies both the properties defined by Theorem 2.2 and hence M' is a popular matching in G. A similar argument proves that for any switching cycle C, the matching $M \cdot C$ is also a popular matching in G.

Now consider any popular 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. Since G is bipartite, it is clear that the cycles are of even length. To see that the paths are of even length, we recall that the addition of $\ell(a)$ at the end of each agent a's preference list ensures that $|M| = |M'|$. Let $T_G = \langle p_1, a_1, \ldots, p_k, a_k, p_{k+1} \rangle$ 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 $M(p_i) = a_i$. Now, using Claim 2.6, we know that both M and M' are contained in G'', therefore the path T_G is contained in G''. Thus, it is easy to see that the path $T = \langle p = p_1, p_2, \ldots, p_{k+1} = p' \rangle$ is present in G_M and it ends in a sink. Note that $w(T)$ cannot be strictly positive since M is a popular matching. Similarly, $w(T)$ cannot be strictly negative. This is because since both M and M' are popular, $w(T) \leq -1$ implies that there exists another path T'_G (or a cycle C'_G) in $M \oplus M'$, whose corresponding path T' (resp. cycle C') in the graph G_M has a positive weight. However, this again contradicts the fact that M is a popular matching. Thus, the path T has weight 0 and ends in a sink and hence is a switching path. A similar argument shows that every cycle in $M \oplus M'$ has a corresponding switching cycle in G_M. Applying these switching paths and cycles to M gives us the desired matching M', thus completing the proof. □

Recall the definition of *choices*(a) for an agent as given by Definition 2.5. 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 $|\mathcal{A}_1| = |\mathcal{P}_1|$. In addition, for every $a \in \mathcal{A}_1$ we have $choices(a) \subseteq \mathcal{P}_1$. The concept of a tight-pair will be used in the next section where we define the cheating strategy. We show that a tight-pair exists whenever there is a non-sink component in the switching graph G_M.

LEMMA 3.7. *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 $\mathcal{A}_q = \{M(p) : p \in \mathcal{P}_q\}$. Since every $p \in \mathcal{P}_q$ is matched, we note that $|\mathcal{A}_q| = |\mathcal{P}_q|$. Consider any agent $a \in \mathcal{A}_q$; then $M(a) \in \mathcal{P}_q$ and note that $M(a) \in choices(a)$. Moreover, 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

$$PopPairs = \{(a, p) \in E : \ M \text{ is a popular matching in } G \text{ and } M(a) = p\}. \quad (3.1)$$

We show that the set *PopPairs* can be computed efficiently using our switching graph. In particular, we prove the following theorem.

THEOREM 3.8. *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.*

Proof. Let G_M be the switching graph with respect to a popular matching M in G. From Property 3.6 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 show how each of these conditions can be efficiently verified.

- The condition (i) can be checked in $O(\sqrt{n}m)$ time by running Algorithm 2.1 and obtaining a popular matching M.
- In order to check condition (iii), recall that every cycle in G_M has weight 0 and is therefore a switching cycle. This implies that every edge belonging to a strongly connected component of G_M is a popular pair. We can therefore use the linear time algorithm of Tarjan [16] to find strongly connected components of G_M and mark edges that satisfy condition (iii).
- In order to check condition (ii), recall that a switching path is a path which has weight 0 and ends in a sink. Therefore such paths can be found only

in sink components of G_M or equivalently paths beginning at vertices in $\mathcal{E}_2$. Furthermore, any sink vertex in G_M has to be a vertex in $\mathcal{E}_1$ according to the partition on rank-1 edges of G. Using the weights on the edges given by Table 3.1, it is easy to see that any 0 weight path ending in a sink has to begin at a vertex $p \in \mathcal{E}_1$. Thus, a simple depth-first search beginning at vertices in $\mathcal{E}_1 \cap \mathcal{E}_2$ and marking all edges that we encounter as popular pairs takes care of condition (ii). It is easy to see that this procedure also takes time linear in the size of G_M.

This completes the proof of the theorem. □

We now consider the problem of counting the number of popular matchings in an instance with ties. McDermid and Irving [13] showed that when preference lists are all strict, the problem of counting the number of popular matchings admits a linear time algorithm. In contrast, we show that the problem turns to be #P-Complete when ties are allowed.

THEOREM 3.9. *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.*

Proof. In order to prove the completeness result, we reduce from the problem of counting the number of perfect matchings in 3-regular bipartite graphs. This problem was shown to be #P-Complete by Dagum and Luby [2]. Let $H = (\mathcal{A} \cup \mathcal{P}, E)$ be a 3-regular bipartite graph. We construct an instance $G = H$ of the popular matching problem by assigning all the edges in E as rank-1 edges. It is well-known that a k-regular bipartite graph admits a perfect matching and it is easy to see that every perfect matching in H is a popular matching in G and vice versa. Thus, the theorem statement follows. □

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 (3.1)).

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 ranked posts of a_1. Since we will be working with another instance H obtained by falsifying the preference list of a_1, we use the following notation throughout. For an agent a, let $f_G(a)$ and $s_G(a)$ denote sets $f(a)$ and $s(a)$ respectively for agent a in G. We use a similar notation for the partition of vertices with respect to the instance under consideration. For instance, let $(\mathcal{O}_1)_G$ denote the set of vertices that belong to $\mathcal{O}_1$ 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. 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 (that is, posts in $s_G(a_1)$). Note that $t > 1$.
- 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 $H \succ_{a_1} G$ if agent a_1 is *better always* in H. It is instructive to consider examples in order to develop intuition regarding the cheating strategies.

EXAMPLE 4.1. Consider the instance G as shown in Figure 2.1(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 such that p_3 is her rank-1 post and p_8 as her rank-2 post.

$$a_5 : p_3 \quad p_8$$

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 single post p (in this case p_8) that will belong to $s_H(a)$ such that a can never be matched to p in any popular matching of H.

EXAMPLE 4.2. Consider the instance G shown in Figure 2.1(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 submits the preference list as follows: p_3 is her rank-1 post whereas p_8 is her rank-2 post.

$$a_1 : p_3 \quad p_8$$

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 should 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 as below and let H denote the falsified instance.

$$a_1 : p_2 \quad p_8$$

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 slightly modified instance as defined in Section 4.3.

In the rest of this section, we establish some facts crucial for developing the optimal cheating strategy. In Section 4.1 we establish that when a_1 manipulates to get better always, the set of s-posts of other agents remains unaffected. This gives us a useful handle on the modified instance obtained after manipulation. In Section 4.2 we show that if a_1 did not get matched to her rank-1 post in any popular matching by being truthful, then she cannot get matched to her rank-1 post even by falsifying her preferences. Finally, in Section 4.3 we formally define the modified instance $\widetilde{G}$.

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, then $s(a)$ for any other agent a remains unchanged. 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). Theorem 4.6 establishes that the set of *even* posts in G_1 and H_1 are the same. We need the following lemmas to prove the theorem.

We first show that if agent a_1 gets matched to her non-rank-1 post in some popular matching in G, then all the rank-1 posts of a_1 are *odd* in G_1. Intuitively, the set of rank-1 posts of a_1 are *highly in demand*.

LEMMA 4.3. *Let $a_1 \in \mathcal{A}_s \cup \mathcal{A}_{f/s}$ when she is truthful. Then, $f_G(a_1) \subseteq (\mathcal{O}_1)_G$.*

Proof. Since $a_1 \in \mathcal{A}_s \cup \mathcal{A}_{f/s}$, it implies that at least one popular matching in G matches a_1 to her non-rank-1 post. We show that if $f_G(a_1) \not\subseteq (\mathcal{O}_1)_G$, then every popular matching in G matches a_1 to her rank-1 post. Assume for the sake of contradiction that $f_G(a_1) \not\subseteq (\mathcal{O}_1)_G$. Let $q \in f_G(a_1)$ such that, $q \in (\mathcal{E}_1 \cup \mathcal{U}_1)_G$. This implies that $a_1 \in (\mathcal{O}_1 \cup \mathcal{U}_1)_G$. Consider the case when $a_1 \in (\mathcal{O}_1)_G$. Then, $s_G(a_1) \subseteq f_G(a_1)$ and therefore a_1 has no non-rank-1 edges incident on it in the graph G'. Thus, a_1 remains matched along her rank-1 edge in every popular matching of G. Now consider the case when $a_1 \in (\mathcal{U}_1)_G$. If a_1 gets matched to a non-rank-1 post in a popular matching M, then M is not a maximum matching on rank-1 edges of G. Thus, in this case also, a_1 remains matched along a rank-1 edge in every popular matching in G. This contradicts the fact that $a_1 \in \mathcal{A}_s \cup \mathcal{A}_{f/s}$. Therefore, $f_G(a_1) \subseteq (\mathcal{O}_1)_G$. □

We now argue that in any instance $H \succ_{a_1} G$, the set of rank-1 posts of a_1 in H must be posts which are *odd* or *unreachable* in G_1. Again, this intuitively implies that the set of rank-1 posts of a_1 in H are posts that are *in demand* in G.

LEMMA 4.4. *Let H be such that $H \succ_{a_1} G$. Then $f_H(a_1) \subseteq (\mathcal{O}_1 \cup \mathcal{U}_1)_G$.*

Proof. We first show that if $f_H(a_1) \cap (\mathcal{E}_1)_G \neq \emptyset$, then the size of the maximum matching on rank-1 edges of H is strictly larger than the size of the maximum matching on rank-1 edges of G. Let G_1 be the graph on rank-1 edges of G and let M_1 be a maximum matching in G_1 that leaves a_1 unmatched. Note that such a matching exists because by Lemma 4.3, $f_G(a_1) \subseteq (\mathcal{O}_1)_G$ which implies that $a_1 \in (\mathcal{E}_1)_G$. Consider the graph H_1, that is, the graph on rank-1 edges of H. Note that M_1 is a matching in H_1 as no other agent changes her preference list. Since each $q_i \in (\mathcal{E}_1)_G$ and $a_1 \in (\mathcal{E}_1)_G$, the addition of the edge (a_1, q_i) creates an augmenting path with respect to M_1 in the graph H_1. Note that if q_i is unmatched in M_1, then this augmenting path is of length 1 containing the unmatched edge (a_1, q_i). Thus, we get another matching M_2 in H_1 obtained by augmenting M_1, such that $|M_2| = |M_1| + 1$.

Assume for contradiction that there exists an instance $H \succ_{a_1} G$ and let $f_H(a_1) \cap (\mathcal{E}_1)_G = \{q_1, \ldots, q_k\}$. Recall that by assumption, $s_G(a_1)$ is a set of t-th ranked posts for a_1. Additionally, by definition, $s_G(a_1)$ denotes the set of most preferred posts on a_1's preference list that belong to $(\mathcal{E}_1)_G$. This implies that the rank of each q_i in a_1's preference list is t or worse. We show that if $f_H(a_1) \cap (\mathcal{E}_1)_G \neq \emptyset$, then every popular matching in H matches a_1 to one of $\{q_1, \ldots, q_k\}$. Thus, the rank of the most preferred post that a_1 gets in H is t or worse, a contradiction to $H \succ_{a_1} G$.

Now if every popular matching in H matches a_1 to one of $\{q_1, \ldots, q_k\}$ then this contradicts the fact that $H \succ_{a_1} G$ and we are done. Otherwise assume that there exists a popular matching M' in H which matches a_1 to some post other than $\{q_1, \ldots, q_k\}$. Let M'_1 denote the matching M' restricted to rank-1 edges of H. Since M' is a maximum matching on rank-1 edges of H, it is clear that $|M'_1| = |M_2|$. Moreover, since $M'(a_1) \in f_H(a_1) \cup s_H(a_1)$ let us consider the following cases:

- $M'(a_1)$ is a true rank-1 post of a_1: In this case M'_1 is a maximum matching in G_1 and note that $|M_1'| = |M_2| = |M_1| + 1$. This contradicts the fact that M_1 is a maximum matching in G_1.

- $M'(a_1)$ is a non rank-1 post of a_1: This implies that M_1' leaves a_1 unmatched. Thus, M_1' is also a matching in G_1 since no other agents changed their preferences. However, $|M_1'| = |M_1| + 1$ which contradicts the fact that M_1 was a maximum matching in G_1.

Thus, we have established that in any instance $H \succ_{a_1} G$, the rank-1 posts of a_1 in H are posts which are *odd* or *unreachable* in G_1. This finishes the proof of the lemma. □

Using the above lemma we now show that the sizes of maximum matching in G_1 and H_1 are indeed the same.

LEMMA 4.5. *Let M_1 be a maximum matching in G_1 such that M_1 leaves a_1 unmatched. Then, in any instance H such that $H \succ_{a_1} G$, M_1 is a maximum matching in H_1.*

Proof. We first note that such a maximum matching M_1 in G_1 which leaves a_1 unmatched exists, because $f_G(a_1) \subseteq (\mathcal{O}_1)_G$, hence $a_1 \in (\mathcal{E}_1)_G$. Assume that M_1 is not a maximum matching in H_1. Then there exists an augmenting path $\langle a_1, p_1, \ldots, a_k, p_k \rangle$ in H_1 with respect to M_1 where both a_1 and p_k are unmatched. However, using the path $\langle p_k, a_k, \ldots, p_1 \rangle$, we have an even length alternating path from p_k to p_1 which implies that $p_1 \in (\mathcal{E}_1)_{G_1}$. However, note that $p_1 \in f_H(a_1)$ and the fact that $p_1 \in (\mathcal{E}_1)_G$ contradicts Lemma 4.4. Thus M_1 is a maximum matching in H_1. □

Finally, using Lemma 4.5 we are ready to prove the main theorem of this subsection. We first show that the set of posts which are *even* in G_1 is the same as the set of posts which are *even* in H_1. This immediately implies that $s(a)$ for other agents remains unchanged. We also prove a useful fact that the set of *odd* agents in G_1 is the same as the set of *odd* agents in H_1.

THEOREM 4.6. *Let H be an instance such that $H \succ_{a_1} G$. Then, (i) $(\mathcal{E}_1)_G \cap \mathcal{P} = (\mathcal{E}_1)_H \cap \mathcal{P}$ and therefore $s_H(a) = s_G(a)$ for every $a \in \mathcal{A} \setminus \{a_1\}$ and, (ii) $(\mathcal{O}_1)_G \cap \mathcal{A} = (\mathcal{O}_1)_H \cap \mathcal{A}$.*

Proof. The case when $f_H(a_1) = f_G(a_1)$ is easy, since $H_1 = G_1$ and both (i) and (ii) are trivially true. Consider the case when $f_H(a_1) \neq f_G(a_1)$ and let M_1 be a maximum matching in G_1 such that M_1 leaves a_1 unmatched. By Lemma 4.5, M_1 is also a maximum matching in H_1. To prove $(\mathcal{E}_1)_G \cap \mathcal{P} = (\mathcal{E}_1)_H \cap \mathcal{P}$, consider the following two cases:

- $p \in (\mathcal{O}_1 \cup \mathcal{U}_1)_G \cap \mathcal{P}$: Assume for contradiction that $p \in (\mathcal{E}_1)_H \cap \mathcal{P}$. This implies that there exists an even length alternating path T with respect to M_1 in H_1 from an unmatched post to p in H_1. The path T cannot contain a_1, since a_1 is unmatched in M_1. Hence T is also present in G_1 contradicting the fact that $p \in (\mathcal{O}_1 \cup \mathcal{U}_1)_G \cap \mathcal{P}$.
- $p \in (\mathcal{E}_1)_G \cap \mathcal{P}$: Let T denote the even length alternating path w.r.t. M_1 in G_1 starting from an unmatched post in M_1. The path T again cannot contain a_1 and hence exists in H_1 thus proving that $p \in (\mathcal{E}_1)_H \cap \mathcal{P}$.

Now, since the preference lists of the agents $a \in \mathcal{A} \setminus \{a_1\}$ remain unchanged, it is clear that $s_H(a) = s_G(a)$.

To prove that $(\mathcal{O}_1)_G \cap \mathcal{A} = (\mathcal{O}_1)_H \cap \mathcal{A}$, we again consider two cases:

- $a \in (\mathcal{O}_1)_G \cap \mathcal{A}$: Let T denote the odd length alternating path w.r.t. M_1 in G_1 starting from an unmatched post in M_1. The path T again cannot contain a_1 and hence exists in H_1 thus proving that $p \in (\mathcal{O}_1)_H \cap \mathcal{P}$.
- $a \in (\mathcal{E}_1 \cup \mathcal{U}_1)_G \cap \mathcal{A}$: Assume for contradiction that $a \in (\mathcal{O}_1)_H \cap \mathcal{A}$. This implies that there exists an odd length alternating path T with respect to M_1

in H_1 from an unmatched post to a in H_1. This path again cannot contain a_1, and hence is present in G_1. This contradicts the fact that $a \in (\mathcal{E}_1 \cup \mathcal{U}_1)_G \cap \mathcal{A}$.

This finishes the proof of the theorem. □

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 prove it using Theorem 4.10 which requires the following lemmas. Using Lemma 4.7 we establish that every true rank-1 post q of a_1 belongs to a non-sink component of G_M. We then construct a tight-pair $\mathcal{A}_q, \mathcal{P}_q$ corresponding to each rank-1 post q and establish that the agent a_1 does not lie in $\mathcal{A}_q$. We further establish that for every agent $a \in \mathcal{A}_q$, the set $choices_H(a) \in \mathcal{P}_q$. We emphasize that the claim is for $choices_H(a)$ which need not be the same as $choices_G(a)$. Lemma 4.8 and Lemma 4.9 prove these facts. Using these facts a simple argument based on pigeon-hole principle allows us to establish the main result which is stated in Theorem 4.10.

LEMMA 4.7. *Let $a_1 \in \mathcal{A}_s$, and let $q \in f_G(a_1)$. Then, q belongs to a non-sink component of G_M and the edge $(M(a_1), q)$ is not contained in a cycle in G_M.*

Proof. We first show that the edge $(M(a_1), q)$ is not contained in a cycle in G_M. Observe that the edge (a_1, q) may get deleted from G' in either Step 4 or Step 9 of Algorithm 2.1. In such a case, we are already done since the edge $(M(a_1), q)$ does not exist in the graph G_M. Now consider the case when the edge (a_1, q) is not deleted in either of the steps. Moreover, for the sake of contradiction assume that there exists a cycle C in G_M which contains the edge $(M(a_1), q)$. Since every cycle in G_M has a weight 0, the cycle C is a switching cycle and hence we get another popular matching $M' = M \cdot C$ in which a_1 gets matched to q. Since $q \in f_G(a_1)$, this contradicts the fact that $a_1 \in \mathcal{A}_s$.

We now show that every $q \in f_G(a_1)$ belongs to a non-sink component of G_M. Assume for the sake of contradiction that there exist some $q \in f_G(a_1)$ such that q belongs to a sink component, say $\mathcal{X}$ of G_M. In this case we show that there exists a switching path T beginning at $M(a_1)$ which uses the edge $(M(a_1), q)$. Using T, we construct another popular matching $M' = M \cdot T$ where a_1 gets matched to q. Thus, we get the desired contradiction as $a_1 \in \mathcal{A}_s$.

It remains to prove that the switching path T exists. We first show that the edge $(M(a_1), q)$ exists in the graph G_M or equivalently, the edge (a_1, q) does not get deleted from G' in either Step 4 or Step 9 of Algorithm 2.1. Note that since $a_1 \in \mathcal{A}_s$ implies that $a_1 \in (\mathcal{E}_1)_G$. Thus, by Claim 2.4, no edge incident on a_1 gets deleted on Step 4. Also since q belongs to a sink component, $q \in (\mathcal{E}_2)_G$ and hence no edge incident on q gets deleted in Step 9. Thus the edge (a_1, q) exists in G'' and hence the edge $(M(a_1), q)$ exists in G_M. Finally note that, $w(M(a_1), q) = +1$ since $M(a_1) \in s_G(a_1)$ and $q \in f_G(a_1)$. We will use the edge $(M(a_1), q)$ as the first edge of our path T.

Observe that since q belongs to a sink component $\mathcal{X}$, by Property 3.3, q has a path to some sink vertex q' in $\mathcal{X}$. From Lemma 4.3, we know that $q \in \mathcal{O}_1$ and by Property 3.2, it is clear that $q' \in \mathcal{E}_1$. Using Table 3.1 of edge weights, we conclude that the path T_1 starting a vertex in $\mathcal{O}_1$ and ending in a vertex in $\mathcal{E}_1$ has weight $w(T_1) = -1$. Thus, we obtain the switching path $T = \langle M(a_1), q, T_1 \rangle$ which ends in the sink q' and has $w(T) = 0$. This completes the proof of the lemma. □

LEMMA 4.8. *Let $a_1 \in \mathcal{A}_s$ and let $q \in f_G(a_1)$. Let $\mathcal{P}_q$ be defined as*

$$\mathcal{P}_q = \{q\} \cup \{q' : \textit{there is a path from } q \textit{ to } q' \textit{ in } G_M\}$$

Let $\mathcal{A}_q = \{M(q') : q' \in \mathcal{P}_q\}$. Then, $a_1 \notin \mathcal{A}_q$.

Proof. Note that since $a_1 \in \mathcal{A}_s$ and $q \in f_G(a_1)$, by Lemma 4.7, q belongs to a non-sink component, say $\mathcal{Y}$, of G_M. If $M(a_1)$ does not belong to $\mathcal{Y}$, then it is clear that $M(a_1) \notin \mathcal{P}_q$ and therefore $a_1 \notin \mathcal{A}_q$.

Otherwise, assume for the sake of contradiction that $M(a_1)$ belongs to $\mathcal{Y}$ and there exists a path from q to $M(a_1)$ in G_M. In this case, we show that the path from q to $M(a_1)$ along with the edge $(M(a_1), q)$ creates a cycle in G_M containing the edge $(M(a_1), q)$ which is a contradiction by Lemma 4.7. Thus, we need to show that the edge $(M(a_1), q)$ does not get deleted in either Step 4 or Step 9 of Algorithm 2.1. Since $M(a_1)$ belongs to a non-sink component, therefore by Property 3.3, $M(a_1) \in (\mathcal{U}_2)_G$. This implies that $a_1 \in (\mathcal{U}_2)_G$. Furthermore, since $q \in f_G(a_1)$, by Lemma 4.3, we know that $q \in (\mathcal{O}_1)_G$. This implies that $a_1 \in (\mathcal{E}_1)_G$. Thus, from Claim 2.4, it is clear that no edge incident on a_1 gets deleted in either Step 9 or Step 4 of Algorithm 2.1. Therefore, the edge $M((a_1), q)$ exists in G_M and along with the path from q to $M(a_1)$ creates a cycle in G_M containing the edge $(M(a_1), q)$. This gives the desired contradiction establishing that $a_1 \notin \mathcal{A}_q$. ☐

LEMMA 4.9. *Let $a_1 \in \mathcal{A}_s$ and let $q \in f_G(a_1)$. Then, there exist sets $\mathcal{A}_q$ and $\mathcal{P}_q$ such that $|\mathcal{A}_q| = |\mathcal{P}_q|$ and for every $a \in \mathcal{A}_q$ we have $choices_H(a) \subseteq \mathcal{P}_q$.*

Proof. Since $a_1 \in \mathcal{A}_s$ and $q \in f_G(a_1)$, from Lemma 4.7, we know that q belongs to a non-sink component, say $\mathcal{Y}$, of G_M. Therefore, using Lemma 3.7, we know that there exists a tight-pair $\mathcal{A}_q$ and $\mathcal{P}_q$ such that $|\mathcal{A}_q| = |\mathcal{P}_q|$ and for each $a \in \mathcal{A}_q$, we have $choices_G(a) \subseteq \mathcal{P}_q$. To prove the lemma it suffices to show that for every $a \in \mathcal{A}_q$, $choices_H(a) \subseteq choices_G(a)$. By Lemma 4.8, $a_1 \notin \mathcal{A}_q$ and therefore, we know that $f_G(a) \cup s_G(a) = f_H(a) \cup s_H(a)$, for all $a \in \mathcal{A}_q$. If no edges had got deleted in Step 4 and Step 9 of Algorithm 2.1 then we would be done since $choices_G(a)$ would have been $f_G(a) \cup s_G(a)$. Therefore we consider edges that get deleted in Step 4 and Step 9 of the algorithm. Observe that since every $q' \in \mathcal{P}_q$ belongs to a non-sink component of G_M, therefore by Property 3.3, $q' \in (\mathcal{U}_2)_G$. This implies that every $a \in \mathcal{A}_q$ is such that $a \in (\mathcal{U}_2)_G$. Thus, by Claim 2.4 no edge incident on a gets deleted in Step 9 of Algorithm 2.1 when executed on G.

It remains to show that if $a' \in \mathcal{A}_q$ and edge (a', q') gets deleted in Step 4 of Algorithm 2.1 when executed on G, then (a', q') also gets deleted in Step 4 of Algorithm 2.1 when executed on H. We first show that if $a \in \mathcal{A}_q$ then $a \in (\mathcal{E}_1 \cup \mathcal{O}_1)_G$. To see this, observe by Lemma 4.3, the post $q \in (\mathcal{O}_1)_G$. Now consider any post $q' \in \mathcal{P}_q$. From the Table 3.1, it is easy to see that $q' \in (\mathcal{O}_1 \cup \mathcal{E}_1)_G$, since posts in $(\mathcal{U}_1)_G$ cannot be reached starting at a post in $(\mathcal{O}_1)_G$. Now since in any popular matching, posts in $(\mathcal{O}_1 \cup \mathcal{E}_1)$ remain matched to agents in $(\mathcal{O}_1 \cup \mathcal{E}_1)$, we conclude that if $a \in \mathcal{A}_q$ then $a \in (\mathcal{E}_1 \cup \mathcal{O}_1)_G$.

If $a \in (\mathcal{E}_1)_G$, by Claim 2.4 no edge incident on a gets deleted in Step 4 of Algorithm 2.1 when executed on G. Finally, let $a \in (\mathcal{O}_1)_G$. If the edge (a, q') got deleted in Step 4, then $q' \in (\mathcal{O}_1 \cup \mathcal{U}_1)_G$. Thus, by Theorem 4.6, $a \in (\mathcal{O}_1)_H$ and $q' \in (\mathcal{O}_1 \cup \mathcal{U}_1)_H$, thus the edge (a', q') continues to get deleted in Step 4 of Algorithm 2.1 when executed on H. This completes the proof of the lemma. ☐

Using the above lemmas we prove the following theorem.

THEOREM 4.10. *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 Lemma 4.7, the post q belongs to a non-sink component of G_M. Furthermore, by Lemma 4.9, 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 $choices_H(a) \subseteq \mathcal{P}_q$. 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 in $choices_H(a')$. 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 4.2 that a rank-1 edge which gets deleted from the graph G' in 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 *corresponds* to a popular matching in $\tilde{G}$ and, (ii) any edge (a, p) that gets deleted in Step 4 of Algorithm 2.1 when executed on $\tilde{G}$ also gets deleted in Step 4 when Algorithm 2.1 is executed on H such that $H \succ_{a_1} G$. 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 M_1 be a maximum matching in G_1. Let $\{q_1, \dots, q_k\}$ be the set of *unreachable* posts with respect to M_1 in G_1. Let us add to the instance G, a dummy agent b whose preference list is of length 1 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 $(q_1, \dots, q_k)$. 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 $\tilde{\mathcal{A}} = \mathcal{A} \cup \{b\}$, $\tilde{E} = E \cup \{(b, q_1), \dots, (b, q_k)\}$ and each (b, q_i) is a rank-1 edge. We recall that for each agent, including the agent b, we add a unique last-resort post at the end of the agent's preference list. By the choice of preference list of b, we note that $f_{\tilde{G}}(b) = \{q_1, \dots, q_k\}$ and $s_{\tilde{G}}(b) = \ell(b)$, the unique last-resort post for agent b.

We note that even after the addition of agent b, a maximum matching M_1 in G_1 continues to be 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 $\tilde{G}_1$. We show that the addition of b leaves the set $s(a)$ unchanged for every agent $a \in \mathcal{A}$.

LEMMA 4.11. *Let $\tilde{G}$ be the graph as defined above. Then, $(\mathcal{E}_1)_{\tilde{G}} \cap \mathcal{P} = (\mathcal{E}_1)_G \cap \mathcal{P}$. Therefore, for every $a \in \mathcal{A}$, we have $s_{\tilde{G}}(a) = s_G(a)$.*

Proof. Let M_1 be a maximum matching in G_1. Since M_1 is also a maximum matching in $\tilde{G}_1$, let us partition the vertices of $\tilde{\mathcal{A}} \cup \mathcal{P}$ w.r.t. M_1 in $\tilde{G}_1$. Note that the addition of agent b makes every post that was *unreachable* in G_1 as *odd* in $\tilde{G}_1$. Now, the set of *even* posts in G_1 and $\tilde{G}_1$ is same and the preferences of the agents are unchanged. This implies that for every $a \in \mathcal{A}$, we have $s_{\tilde{G}}(a) = s_G(a)$. ☐

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 $\tilde{M}$ satisfied both the properties of Theorem 2.2 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 $\tilde{G}$. Having made these definitions, we can now prove the following lemmas.

LEMMA 4.12. *Let (a, p) be an edge which gets deleted in Step 4 of Algorithm 2.1 when it is executed on $\tilde{G}$. Then (a, p) gets deleted in Step 4 when Algorithm 2.1 is executed on any instance H such that $H \succ_{a_1} G$.*

Proof. As mentioned earlier, all vertices in $\tilde{G}_1$ are either *odd* or *even*, hence if an edge (a, p) got deleted in Step 4 of Algorithm 2.1, then it implies that $\{a, p\} \in (\mathcal{O}_1)_{\tilde{G}}$. To prove the lemma statement, we show that such an edge gets deleted either as an

$\mathcal{O}_1\mathcal{O}_1$ edge, or as an $\mathcal{O}_1\mathcal{U}_1$ edge in H_1. To this end we establish the following two claims.

1. Any agent that is *odd* in $\tilde{G}_1$ is also *odd* in H_1. That is, $(\mathcal{O}_1)_{\tilde{G}} \cap \mathcal{A} = (\mathcal{O}_1)_H \cap \mathcal{A}$: Consider a maximum matching M_1 in $\tilde{G}_1$ which leaves a_1 and b unmatched. As argued earlier, this is a maximum matching in G_1 and hence a maximum matching in H_1. We consider two cases:
   - $a \in (\mathcal{O}_1)_{\tilde{G}} \cap \mathcal{A}$: Let T denote the odd length alternating path w.r.t. M_1 in $\tilde{G}_1$ starting at an unmatched post in M_1 to the agent a. The path T cannot contain a_1 since a_1 is unmatched in M_1. This implies the path T exists in H_1 thus proving that $a \in (\mathcal{O}_1)_H \cap \mathcal{P}$.
   - $a \in (\mathcal{E}_1 \cup \mathcal{U}_1)_{\tilde{G}} \cap \mathcal{A}$: Assume for contradiction that $a \in (\mathcal{O}_1)_H \cap \mathcal{A}$. This implies that there exists an odd length alternating path T with respect to M_1 in H_1 from an unmatched post to a in H_1. This path again cannot contain a_1, and hence is present in $\tilde{G}_1$. This contradicts the fact that $a \in (\mathcal{E}_1 \cup \mathcal{U}_1)_{\tilde{G}} \cap \mathcal{A}$.
2. Any post that is *odd* in $\tilde{G}_1$ is either *odd* or *unreachable* in H_1. That is, $(\mathcal{O}_1)_{\tilde{G}} \cap \mathcal{P} = (\mathcal{O}_1 \cup \mathcal{U}_1)_H \cap \mathcal{P}$: Note that, using Lemma 4.11 and Theorem 4.6(i) we can conclude that $(\mathcal{E}_1)_{\tilde{G}} \cap \mathcal{P} = (\mathcal{E}_1)_H \cap \mathcal{P}$. This implies that $(\mathcal{O}_1)_{\tilde{G}} \cap \mathcal{P} = (\mathcal{O}_1 \cup \mathcal{U}_1)_H \cap \mathcal{P}$. (Recall that $(\mathcal{U}_1)_{\tilde{G}} = \emptyset$.)

Thus, using (1) and (2) above we conclude that any $\mathcal{O}_1\mathcal{O}_1$ edge deleted in Step 4 of Algorithm 2.1 when executed on $\tilde{G}$ is also deleted in Step 4 when the same algorithm is executed on any $H \succ_{a_1} G$. □

At this moment, we recall Example 4.2 and the instance H which was obtained by falsifying the preference list of a_1 to p_2, p_8. This example motivated us to define the instance $\tilde{G}$. We now emphasize that we could not work directly with the instance G, because for every agent $a \in \mathcal{A} \setminus \{a_1\}$ we cannot guarantee that $choices_H(a) \subseteq choices_G(a)$. In fact, the above example shows that $choices_G(a_4) = \{p_2, p_8\}$ whereas $choices_H(a_4) = \{p_2, p_3, p_8\}$. Our next lemma shows that the careful definition of $\tilde{G}$ ensures that for an agent a in a non-sink component of $\tilde{G}_{\tilde{M}}$ we have $choices_H(a) \subseteq choices_{\tilde{G}}(a)$. This lemma along with the tight-pair defined earlier will be a useful tool for proving our results in the next section.

LEMMA 4.13. *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 $H \succ_{a_1} G$. Then $choices_H(a) \subseteq choices_{\tilde{G}}(a)$.*

Proof. Recall that for any $a \in \mathcal{A} \setminus \{a_1\}$, $f_{\tilde{G}}(a) = f_G(a) = f_H(a)$ and $s_{\tilde{G}}(a) = s_G(a) = s_H(a)$. We also know that, in any instance, for an agent a, $choices(a) \subseteq f(a) \cup s(a)$. Thus, for an agent $a \in \mathcal{A} \setminus \{a_1\}$, if it were the case that $choices_{\tilde{G}}(a) = f_{\tilde{G}}(a) \cup s_{\tilde{G}}(a)$, then the lemma statement holds trivially. However due to deletion of edges in Step 4 and Step 9 of Algorithm 2.1 when executed on $\tilde{G}$, it may be the case that $choices_{\tilde{G}}(a) \subset f_{\tilde{G}}(a) \cup s_{\tilde{G}}(a)$. We note that since $M(a)$ belongs to a non-sink component it implies that both $\{a, M(a)\} \in (\mathcal{U}_2)_{\tilde{G}}$. Therefore, by Claim 2.4, no edge incident on a gets deleted in Step 9. Moreover, by Lemma 4.12, it is clear that if an edge (a, p) gets deleted in Step 4 of Algorithm 2.1 run in $\tilde{G}$, then the same edge gets deleted in Step 4 when executed on H. This proves that, for any $a \in \mathcal{A} \setminus \{a_1\}$, such that, a belongs to a non-sink component, we have $choices_H(a) \subseteq choices_{\tilde{G}}(a)$. □

Armed with these observations we now develop the cheating strategies for the manipulative agent a_1.

5. Cheating strategies. In this section we develop a 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. Thus, $f_G(a_1) = P_1$ 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 $\tilde{M} = M \cup \{(b, \ell(b))\}$. 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 5.1) 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.

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 one of 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 q be some post in $f_G(a_1)$. Let $a_2 = \tilde{M}(q)$ and $p' \in s_G(a_2)$. Set p' as the rank-2 post of a_1 in the falsified instance.
5. $\mathcal{L}_f = p, p'$.

FIG. 5.1. *Cheating strategy for $a_1 \in \mathcal{A}_s$.*

We first show that the post p' chosen as a rank-2 post for a_1 is valid, that is, p' is not one of the dummy last-resort posts that we introduce.

LEMMA 5.1. *The post p' chosen in Step 4 of Figure 5.1 is matched in $\tilde{M}$ and $p' \neq \ell(a_2)$. Furthermore, p' is a non-rank-1 post for a_2.*

Proof. We note that post p' is chosen in Step 4 of Figure 5.1 as follows: Let q be some post such that $q \in f_G(a_1)$ and let $a_2 = \tilde{M}(q)$. Then post p' is such that $p' \in s_G(a_2)$. Also recall that the popular matching $\tilde{M}$ in $\tilde{G}$ is obtained from a popular matching M in G. Therefore, to show that p' is matched in $\tilde{M}$, it suffices to show that p' is matched in M. Assume for contradiction, p' is not matched in M. Then, we can obtain another matching M' in G by demoting a_2 to p' and promoting a_1 to q and leaving rest of the agents matched as it is. It is easy to see that M' is popular in G and $M'(a_1) \in f_G(a_1)$. This contradicts the fact that $a_1 \in \mathcal{A}_s$. Thus, p' is matched in M and therefore matched in $\tilde{M}$. Finally, if $p' = \ell(a_2)$, then it is clear that p' has to be unmatched in M since a_2 is the only agent that has an edge to $\ell(a_2)$ and $\tilde{M}(a_2) = M(a_2) = q$.

To show that p' is a non-rank-1 post for a_2, note that by assumption, $p' \in s_G(a)$. Assume for contradiction that p' is a rank-1 post for a_2. This implies that $a_2 \in (\mathcal{O}_1)_G$. Also as $q \in f_G(a_1)$, by Lemma 4.3, we know that $q \in (\mathcal{O}_1)_G$. Thus, the edge (a_2, q) should have been deleted as on $\mathcal{O}_1\mathcal{O}_1$ edge. However, since $M(a_2) = q$, this edge did not get deleted. This gives us the desired contradiction and proves that p' is in fact a non-rank-1 post for a_2.

This completes the proof of the lemma. □

The following two lemmas establish the correctness of our strategy.

LEMMA 5.2. *Let H denote the instance obtained when a_1 submits $\mathcal{L}_f = p, p'$ as computed by the algorithm in Figure 5.1. Then, there exists a popular matching in H that matches a_1 to p.*

Proof. We begin by noting that $s_H(a_1) = \{p'\}$. This is because $p' \in s_G(a_2)$ and hence $p' \in (\mathcal{E}_1)_G$, therefore by Theorem 4.6 (i), we conclude that $p' \in (\mathcal{E}_1)_H$. We now construct a popular matching M' in H such that $M'(a_1) = p$. Consider a popular matching M in G and the corresponding matching $\tilde{M}$ in $\tilde{G}$. Let $N = \tilde{M} \setminus \{(a_1, \tilde{M}(a_1)), (b, \tilde{M}(b))\}$. This leaves the post $\tilde{M}(a_1)$ unmatched in N. We now consider the switching graph $\tilde{G}_{\tilde{M}}$. Recall the choice of rank-1 post p of a_1 from the algorithm in Figure 5.1. If p has a path to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$, then let T denote a path from p to $\tilde{M}(a_1)$ in the graph $\tilde{G}_{\tilde{M}}$. Else if p belongs to a sink component $\mathcal{X}$ of $\tilde{G}_{\tilde{M}}$, then let T denote a path from p to a sink in $\mathcal{X}$. (Recall from Property 3.3 that every post belonging to a sink component has a path to a sink.) We note that the path T does not contain $\tilde{M}(b)$ since $\tilde{M}(b) = \ell(b)$ and $\ell(b)$ does not have any incoming edges. Now, consider the matching $N' = (N \cdot T) \cup \{(a_1, p)\}$. We prove that N' is popular in H. This has two parts:

- We claim that for every $a \in \mathcal{A}$, $N'(a) \in f_H(a) \cup s_H(a)$. Note that for every $a \in \mathcal{A} \setminus \{a_1\}$, we have $N'(a) \in f_{\tilde{G}}(a) \cup s_{\tilde{G}}(a)$ which implies that $N'(a) \in f_H(a) \cup s_H(a)$. Also, $N'(a_1) = p$ and note that $f_H(a_1) = \{p\}$.
- Finally, it remains to show that N' is a maximum matching on rank-1 edges of H. Recall that the size of a maximum matching in $\tilde{G}_1$ is the same as the size of a maximum matching in H_1. Thus, to show that N' is a maximum matching on rank-1 edges of H, we show that the number of rank-1 edges in N' is equal to the number of rank-1 edges in $\tilde{M}$. Now, recall the definition of path T from above and the way N' was obtained from $\tilde{M}$. It is easy to note that the number of rank-1 edges in $\tilde{M}$ and N is the same. Furthermore, since $N' = (N \cdot T) \cup \{(a_1, p)\}$, we conclude that the difference between the number of rank-1 edges in N' and N is exactly equal to $w(T) + 1$. Note that the edge (a_1, p) is a rank-1 edge in H. Thus to show that N' is a maximum matching in H_1, or equivalently, the number of rank-1 edges in N' and $\tilde{M}$ is equal, it suffices to show that $w(T) = -1$.
  We now establish that $w(T) = -1$. For this, we examine the end points of T, namely the source which is post p and the target which is either a sink in $\tilde{G}_{\tilde{M}}$ or the post $\tilde{M}(a)$. Since p is a post ranked $2, \ldots, t-1$ in a_1's true preference list, and the rank of posts in $s_G(a_1)$ is exactly t, it implies that $p \in (\mathcal{O}_1 \cup \mathcal{U}_1)_G$. Therefore we can conclude that that $p \in (\mathcal{O}_1)_{\tilde{G}}$. Moreover, if the target is a sink, by Property 3.2 it belongs to $(\mathcal{E}_1)_{\tilde{G}}$. Finally, if the target is $\tilde{M}(a_1)$, by the assumption that $a_1 \in \mathcal{A}_s$, it implies that $\tilde{M}(a_1) \in (\mathcal{E}_1)_{\tilde{G}}$. Therefore using Table 3.1 we conclude that $w(T) = -1$. This finishes our argument that M' is a maximum matching on rank-1 edges of H.

This completes the proof of the lemma. □

LEMMA 5.3. *Let H denote the instance obtained when a_1 submits $\mathcal{L}_f = p, p'$ as computed by the algorithm in Figure 5.1. Then, every popular matching in H matches a_1 to p.*

Proof. We have already shown using Lemma 5.2 that there exists some popular matching in H that matches a_1 to p. For the purpose of proving that every popular matching in H matches a_1 to p, we will work with the graph G and the switching graph G_M corresponding to a popular matching M in G. For contradiction assume that there exists a popular matching M'' in H such that $M''(a_1) = p'$. Note that the rank-2 post p' of a_1 is chosen as follows. Let q be some post such that $q \in f_G(a_1)$ and let $a_2 = \tilde{M}(q)$. Then the post p' is such that $p' \in s_G(a_2)$. Recall that $\tilde{M}$ is a popular matching in $\tilde{G}$ which is obtained from a popular matching M in G. Note that since $q \in f_G(a_1)$, by Lemma 4.7, q belongs to a non-sink component in G_M. Moreover, by Lemma 4.9, there exists sets $\mathcal{A}_q$ and $\mathcal{P}_q$ such that $|\mathcal{A}_q| = |\mathcal{P}_q|$, $a_1 \notin \mathcal{A}_q$ and for every $a \in \mathcal{A}_q$, we have $choices_H(a) \subseteq \mathcal{A}_q$. We show that the post p' also belongs to $\mathcal{P}_q$ and therefore if $M''(a_1)$ matches a_1 to p' then there exists at least one agent $a \in \mathcal{A}_q$ who does not have a post to be matched in $choices_H(a)$. Thus, M'' cannot be a popular matching in H.

It remains to prove that $p' \in \mathcal{P}_q$. Note that $q = M(a_2) \in f_G(a_2)$ and $p' \in s_G(a_2)$. If G_M contains the edge (q, p'), then we are done; since by definition of $\mathcal{P}_q$, it is clear that $p' \in \mathcal{P}_q$. The edge (q, p') can be missing in G_M only if it gets deleted either in Step 4 or in Step 9 of Algorithm 2.1. By Lemma 5.1, we know that a_2 treats p' as her non-rank-1 post. Thus, the edge (a_2, p') cannot get deleted in Step 4 since only rank-1 edges get deleted in Step 4. Since q belongs to a non-sink component in G_M, by Property 3.3 it implies that $q \in (\mathcal{U}_2)_G$. Therefore $M(q) = a_2$ also belongs to $(\mathcal{U}_2)_G$. Thus by Claim 2.4, the edge (a_2, p') does not get deleted in Step 9 of Algorithm 2.1 and hence $p' \in \mathcal{P}_q$. This completes the proof of the lemma. □

The following lemma establishes the optimality of our strategy.

LEMMA 5.4. *The strategy output by the algorithm in Figure 5.1 is the optimal strategy for a_1 to get* better always*, when $a_1 \in \mathcal{A}_s$.*

Proof. To prove that our strategy in Figure 5.1 is optimal, we note that when a_1 was truthful, she got matched to her t-th ranked post in every popular matching in G. Using our strategy she either gets matched to her k-th ranked post in every popular matching in H where $k < t$ or we declare that true preference list is optimal in which case she remains matched to her t-th ranked post where $k = t$. Then there exists no instance H such that a popular matching in H matches a_1 to a post q' which a_1 strictly prefers to her k-th ranked post.

For the sake of contradiction assume that there exists such an instance H obtained by falsifying the preference list of a_1 alone. Let M' be some popular matching of H such that $M'(a_1) = q$ and a_1 strictly prefers q to her k-th ranked post. Since our strategy in Figure 5.1 did not find q, the post q belongs to a non-sink component $\mathcal{Y}$ in $\tilde{G}_{\tilde{M}}$ and moreover there exists no path from q to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$. Now consider the two sets $\mathcal{P}_q$ and $\mathcal{A}_q$ as defined by Lemma 3.7. We know that $|\mathcal{A}_q| = |\mathcal{P}_q|$. Moreover, for every $a \in \mathcal{A}_q$, we have $\{choices_{\tilde{G}}(a)\} \subseteq \mathcal{P}_q$. Note that, $a_1 \notin \mathcal{A}_q$, otherwise $\tilde{M}(a_1) \in \mathcal{P}_q$ which implies that there exists a path from q to $\tilde{M}(a_1)$, a contradiction. Furthermore, note that $\ell(b) \notin \mathcal{P}_q$ and therefore, $b \notin \mathcal{A}_q$. By Lemma 4.13, we know that for every $a \in \mathcal{A}_q$, $choices_H(a) \subseteq choices_{\tilde{G}}(a)$, which implies $choices_H(a) \subseteq \mathcal{P}_q$. Therefore, if M' matches a_1 to q, there exists at least one agent $a \in \mathcal{A}_q$ who does not have a post to be matched in $choices_H(a)$ and hence M' is not a popular matching in H. This completes the proof of the lemma. □

Using Lemma 5.3 and Lemma 5.4 we conclude the following theorem.

THEOREM 5.5. *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* $\tilde{G}_{\tilde{M}}$.

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 5.2) is to search for an *even* post p' in G_1 which belongs to a non-sink component of $\tilde{G}_{\tilde{M}}$. In addition the post p' does not have a path T to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$.

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 $\tilde{G}_{\tilde{M}}$.
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.
4. $\mathcal{L}_f = p, p'$.

FIG. 5.2. *Cheating strategy for* $a_1 \in \mathcal{A}_{f/s}$ *to get better always.*

We prove the correctness and optimality of our strategy using the following two lemmas.

LEMMA 5.6. *Let H denote the instance obtained when a_1 submits $\mathcal{L}_f = p, p'$ as computed by the algorithm in Figure 5.2. Then, every popular matching in H matches a_1 to p.*

Proof. We first show that a popular matching M in G which matches a_1 to her rank-1 post is in fact a popular matching in H. This follows from the fact that M is a maximum matching on rank-1 edges of G, and it continues to be a maximum matching on rank-1 edges of H. In addition, for every $a \in \mathcal{A} \setminus \{a_1\}$, we have $M(a) \in f_G(a) \cup s_G(a)$ which implies that $M(a) \in f_H(a) \cup s_H(a)$. Thus, M is a popular matching in H.

We now show that every popular matching of H matches a_1 to p. Assume not. Then let M' be a popular matching in H such that $M'(a_1) \in s_H(a_1)$. It is easy to see that $s_H(a_1) = \{p'\}$. This is because by choice, $p' \in (\mathcal{E}_1)_G \cap \mathcal{P}$ and therefore by Theorem 4.6, we have $p' \in (\mathcal{E}_1)_H \cap \mathcal{P}$. Thus, p' is the most preferred *even* post in H_1 for agent a_1 and therefore $s_H(a_1) = \{p'\}$.

We know that p' belongs to a non-sink component $\mathcal{Y}_1$ of $\tilde{G}_{\tilde{M}}$ and p' does not have a path to $\tilde{M}(a_1) = p$. Let us define tight-pair $\mathcal{P}_{p'}$ and $\mathcal{A}_{p'}$ as in Lemma 3.7. Thus, we have $|\mathcal{P}_{p'}| = |\mathcal{A}_{p'}|$ and every $a \in \mathcal{A}_{p'}$ satisfies $choices_{\tilde{G}}(a) \subseteq \mathcal{P}_{p'}$. Since p' does not have a path to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$, it is clear that $\tilde{M}(a_1) \notin \mathcal{P}_{p'}$. Therefore $a_1 \notin \mathcal{A}_{p'}$. Furthermore, $\tilde{M}(b) \notin \mathcal{P}_{p'}$ since $\tilde{M}(b) = \ell(b)$ does not have any incoming edges. Since every $a \in \mathcal{A}_{p'}$ belongs to a non-sink component of $\tilde{G}_{\tilde{M}}$, using Lemma 4.13 we know that $choices_H(a) \subseteq choices_{\tilde{G}}(a)$. Thus, for every $a \in \mathcal{A}_{p'}$, we have $choices_H(a) \subseteq \mathcal{P}_{p'}$. Now if $M'(a_1) = p'$ then there exists at least one agent $a \in \mathcal{A}_{p'}$ which does not have a post to be matched in $choices_H(a)$. Thus, M' cannot be a popular matching in H. ∎

LEMMA 5.7. *Let $a_1 \in \mathcal{A}_{f/s}$ when she is truthful. Then the strategy output by the algorithm in Figure 5.2 is the optimal strategy for a_1 to get* better always*.*

Proof. To prove the lemma statement, assume that no post satisfies the two properties in Step 1 of Figure 5.2. And for the sake of contradiction assume that there exists an instance H where every popular matching in H matches a_1 to a post in $f_G(a_1)$. Let $q' \in s_H(a_1)$, then by the definition of $s(a)$, we know that $q' \in (\mathcal{E}_1)_H \cap \mathcal{P}$. We first claim that q' is not a true rank-1 post for a_1, that is, $q' \notin f_G(a_1)$. This is because, if $q' \in f_G(a_1)$, then Lemma 4.3 implies that $q' \in (\mathcal{O}_1)_G$. Moreover, by Theorem 4.6(i), we can conclude that $q' \in (\mathcal{O}_1 \cup \mathcal{U}_1)_H \cap \mathcal{P}$, which contradicts the fact that $q' \in (\mathcal{E}_1)_H \cap \mathcal{P}$. Now, since our algorithm in Figure 5.2 did not find q', either:

1. q' belongs to a sink component in $\tilde{G}_{\tilde{M}}$, or
2. q' belongs to a non-sink component of $\tilde{G}_{\tilde{M}}$ and q' has a path T to $\tilde{M}(a_1)$.

In each of the above two cases we will construct a popular matching in H that matches a_1 to q'. This will give us the desired contradiction. We now split the two cases mentioned above depending on whether the post q' has a path to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$.

- q' does not have a path to $\tilde{M}(a_1)$: Since q' was not found by our algorithm in Figure 5.2, and q' does not have a path to $\tilde{M}(a_1)$, it implies that q' belongs to a sink component, say $\mathcal{X}_i$ of $\tilde{G}_{\tilde{M}}$. Our intermediate goal is to obtain a popular matching N in $\tilde{G}$ in which the post q' is unmatched. If q' is a sink vertex in $\tilde{G}_{\tilde{M}}$, then it is unmatched and in this case $N = \tilde{M}$. Otherwise since q' belongs to a sink component $\mathcal{X}_i$ of $\tilde{G}_{\tilde{M}}$, there exists a path T_1 starting at q' which ends in a sink in $\mathcal{X}_i$. Let $\mathcal{A}_{T_1} = \{a : p \in T \text{ and } \tilde{M}(p) = a\}$ denote the set of agents matched to posts in T_1. Since q' does not have a directed path to $\tilde{M}(a_1)$, the agent $a_1 \notin \mathcal{A}_{T_1}$. Moreover, $b \notin \mathcal{A}_{T_1}$ since $\tilde{M}(b) = \ell(b)$ does not have any incoming edges. In this case, let $N = M \cdot T_1$ which leaves the post q' unmatched. To see that N is indeed popular in $\tilde{G}$, it suffices to show that $w(T_1) = 0$. Consider the two end-points of T_1 – the source q' and the destination which is a sink vertex in $\tilde{G}_{\tilde{M}}$. Note that $q' \in s_H(a_1)$ implies $q' \in (\mathcal{E}_1)_H \cap \mathcal{P} = (\mathcal{E}_1)_{\tilde{G}} \cap \mathcal{P}$ and the end point of the path T_1 is a sink vertex in $\tilde{G}_{\tilde{M}}$ which again belongs to $(\mathcal{E}_1)_{\tilde{G}} \cap \mathcal{P}$. Therefore from Table 3.1, it is clear that $w(T) = 0$. Thus, we have obtained a matching N which is popular in $\tilde{G}$ and leaves the post q' unmatched.

  Now consider the popular matching N in $\tilde{G}$. We note that since $a_1 \in \mathcal{A}_{f/s}$, there exists a post say $q \in s_G(a_1) = s_{\tilde{G}}(a_1)$ such that (a_1, q) is a popular pair in both G and $\tilde{G}$. (Recall that by the definition of $\tilde{G}$, every popular matching in G corresponds to a popular matching in $\tilde{G}$.) Now, consider the switching graph $\tilde{G}_N$. Since (a_1, q) is a popular pair in $\tilde{G}$ and $N(a_1) \neq q$, we know that the edge (a_1, q) belongs to either a switching path, say T_2, or a switching cycle say C_2 of $\tilde{G}_N$. Let us denote the path T_2 or the cycle C_2 by T. We first note that $w(T) = 0$, since T is either a switching path or a switching cycle. Now consider the matching $N' = N \cdot T$ in which a_1 is matched to q. Next consider the matching $N'' = N' \setminus \{(a_1, q), (b, \ell(b))\} \cup \{(a_1, q')\}$. We now argue that N'' is a popular matching in H. First note that for every $a \in \mathcal{A} \setminus \{a_1\}$, we have $N''(a) \in f_{\tilde{G}}(a) \cup s_{\tilde{G}}(a)$ which implies that $N''(a) \in f_H(a) \cup s_H(a)$. For agent a_1, $N''(a) \in s_H(a_1)$ by the assumption that $q' \in s_H(a_1)$. Next, observe that the number of rank-1 edges in $\tilde{M}$, N and N' is the same since they are obtained by applying switching paths or cycles. Finally, the number of rank-1 edges of N'' is the same as N' since a_1 is matched to a non-rank-1 post in both the matchings. This completes the proof that N'' is a popular matching in H which matches a_1 to q'. Thus, we get the desired contradiction that every popular matching in H matches a_1 to one of her rank-1 posts.

- q' has a path to $\tilde{M}(a_1)$: In this case, q' may belong to a sink component or a non-sink component of $\tilde{G}_M$. In either case, let T denote the path from q' to $\tilde{M}(a_1)$. Since $q' \in (\mathcal{E}_1)_{\tilde{G}}$ and $\tilde{M}(a_1) \in (\mathcal{O}_1)_{\tilde{G}}$, using Table 3.1, it is clear that $w(T) = +1$.
  We obtain N'' as follows: let $N = \tilde{M} \setminus \{(a_1, \tilde{M}(a_1)), (b, \tilde{M}(b))\}$. This leaves the post $\tilde{M}(a_1)$ unmatched in N. Let $N' = N \cdot T$ and finally let $N'' = N' \cup \{(a_1, q')\}$. Using the same arguments as above it is possible to show that for every $a \in \mathcal{A}$, $M'(a) \in f_H(a) \cup s_H(a)$. We note that since $w(T) = +1$ and a_1 no longer remains matched to one of her rank-1 posts, the number of rank-1 edges in M' and $\tilde{M}$ is the same. Thus, N'' is a maximum matching on rank-1 edges in H. Therefore, N'' is popular in H and $N''(a_1) \in s_H(a_1)$ which gives us the required contradiction.

This finishes the proof of the lemma. ☐

Using Lemma 5.6 and Lemma 5.7, we conclude the following theorem.

THEOREM 5.8. *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 from p' to $\tilde{M}(a_1)$ in $\tilde{G}_{\tilde{M}}$.

Time complexity: We now discuss the time complexity of computing the optimal cheating strategy using Theorem 5.9.

THEOREM 5.9. *The optimal falsified preference list for a 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:
1. Construct the switching graph.
2. Compute the set of popular pairs.
3. Run the algorithm given by Figure 5.1 or Figure 5.2 as appropriate for the 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 5.1 and Figure 5.2 have checks which can be done in time which is linear in the size of the switching graph. Thus the steps (1) and (2) defined above decide the complexity of our cheating strategy. In case of ties, we have shown that both steps can be computed in $O(\sqrt{n}m)$ time. In case of strict lists, using the switching graph given by McDermid and Irving [13], both steps can be computed in $O(m+n)$ time. Thus we have the desired result. ☐

Remark: In each case we constructed a falsified preference list for the manipulative agent which is strict and of length exactly two. However, by appending the rest of the posts in $\mathcal{P}$ at the end of a_1's preference list, there is no change in the popular matchings that the instance H admits. Thus, we conclude that, if an agent can manipulate to get *better always* she can achieve the same when preference lists are required to be complete.

6. Conclusion. In this paper we presented cheating strategies for a manipulative agent to get *better always*. It would be interesting to study how two or more agents co-operate and falsify their preference lists in order to get *better always*. We leave this as an open problem. Another contribution of the paper is the switching graph characterization of the popular matchings problem with ties. McDermid and

Irving [13] have used their characterization in case of strict lists to give efficient algorithms for the *optimal* popular matchings problem [10, 13]. It would be useful to exploit the characterization developed here and design efficient algorithms for the *optimal* popular matchings problem with ties allowed. We leave that as another open question.

Acknowledgment: The author would like to thank Prof. Vijaya Ramachandran for several useful discussions on the problem and also to the anonymous reviewers for the detailed comments which improved the presentation considerably.

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] P. Dagum and M. Luby. Approximating the permanent of graphs with large factors. *Theor. Comput. Sci.*, 102(2):283–305, 1992.

[3] D. Gale and L. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–14, 1962.

[4] P. Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioural Sciences*, 20:166–173, 1975.

[5] 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.

[6] 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.

[7] R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. Rank-maximal matchings. *ACM Transactions on Algorithms*, 2(4):602–610, 2006.

[8] 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.

[9] T. Kavitha, J. Mestre, and M. Nasre. Popular mixed matchings. *Theoretical Computer Science*, 412(24):2679–2690, 2011.

[10] T. Kavitha and M. Nasre. Note: Optimal popular matchings. *Discrete Applied Mathematics*, 157(14):3181–3186, 2009.

[11] M. Mahdian. Random popular matchings. In *Proceedings of 7th ACM Conference on Electronic Commerce*, pages 238–242, 2006.

[12] 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.

[13] E. McDermid and R. W. Irving. Popular matchings: structure and algorithms. *Journal of Combinatorial Optimization*, 22(3):339–358, 2011.

[14] M. Nasre. Popular Matchings: Structure and Cheating Strategies. In *Proceedings of 30th Annual Symposium on Theoretical Aspects of Computer Science*, pages 412–423, 2013.

[15] W. R. Pulleyblank. *Handbook of Combinatorics (Vol. 1)*, chapter Matchings and Extensions, pages 179–232. MIT Press, Cambridge, MA, USA, 1995.

[16] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[17] 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.