Robust Popular Matchings

Martin Bullinger
University of Oxford
Oxford, United Kingdom
martin.bullinger@cs.ox.ac.uk

Rohith Reddy Gangam
University of California, Irvine
Irvine, USA
rgangam@uci.edu

Parnian Shahkar
University of California, Irvine
Irvine, USA
shahkarp@uci.edu

ABSTRACT

We study popularity for matchings under preferences. This solution concept captures matchings that do not lose against any other matching in a majority vote by the agents. A popular matching is said to be *robust* if it is popular among multiple instances. We present a polynomial-time algorithm for deciding whether there exists a robust popular matching if instances only differ with respect to the preferences of a single agent while obtaining NP-completeness if two instances differ only by a downward shift of one alternative by four agents. Moreover, we find a complexity dichotomy based on preference completeness for the case where instances differ by making some options unavailable.

KEYWORDS

Matchings under preferences; popularity; robustness

ACM Reference Format:
Martin Bullinger, Rohith Reddy Gangam, and Parnian Shahkar. 2024. Robust Popular Matchings. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Matchings under preferences have been an enduring object of study for many decades with an abundance of applications ranging as far as labor markets, organ transplantation, or dating. The general idea is to match two types of agents that each possess a ranking of the agents from the other side. One of the most celebrated results in this area is the Deferred Acceptance Algorithm by Gale and Shapley [15] for identifying so-called stable matchings. These are matchings that do not admit a blocking pair of agents preferring each other to their designated matching partners. Subsequently, many related algorithms and solution concepts have been developed and investigated. Among these, the concept of popular matchings proposed by Gärdenfors [19] has caused substantial research, see, e.g., the book chapter by Cseh [10]. A matching is said to be popular if it does not lose a majority election against any other matching. In this election, the agents vote according to their preferences between their respective matching partners. As already shown by Gärdenfors, stable matchings are popular, but the converse is not necessarily true.

A common feature of real-world scenarios is that it can be hard for agents to express their exact preferences. For instance, an agent might report their preferences but alter them at a later stage. In a matching market, this situation can easily occur when the interaction with other agents changes the opinion about these agents. Or an agent might maintain their preferences but then some event happens that turns some of their options into unacceptable or impossible options, or creates new opportunities. Again, such situations frequently occur, for instance, when some of the matching partners move away, or when an agent gets to know new agents.

In terms of algorithmic solutions, it would be desirable to establish a solution that is robust to changes. To formalize this idea, we propose *robust popular matchings*, which are popular matchings across multiple instances. We then consider RobustPopularMatching, the algorithmic problem of computing a robust popular matching, or to decide that no such matching exists. Specifically, we consider this problem for the two scenarios described above. First, we assume that the set of available matching partners is maintained, but agents may *alter their preferences*. We present a polynomial-time algorithm for the case where only a single agent alters their preferences. The key idea for this algorithm is to define a set of hybrid instances on which we search for popular matchings that include a predefined edge. By contrast, we show that RobustPopularMatching becomes NP-complete if four agents may perform a particularly simple type of preference alteration called a downward shift. Second, we consider RobustPopularMatching for the case where the preference orders of the agents are maintained but options may *become unavailable*. We find a complexity dichotomy based on whether one of the input instances has every potential partner available. We conclude by discussing related problems, such as robustness for related popularity notions.

2 RELATED WORK

Popularity was first considered by Gärdenfors [19] under the name of "majority assignments." He also introduced strong popularity, the version of the solution concept where a matching has to beat every other matching in a majority election. In a broader interpretation, popular and strongly popular matchings correspond to weak and strong Condorcet winners in social choice [9]. The book chapters by Cseh [10] and Manlove [23, Chapter 7] provide a good overview of previous work on popular matchings.

Our research continues a stream of algorithmic results on popularity. In this line of work, close relationships between popularity and stability often play an important role: Popular matchings can have different sizes and stable matchings are popular matchings of minimum size. Moreover, popular matchings of maximum size can be computed efficiently [11]. By contrast, somewhat surprisingly, it is NP-hard to decide if there exists a popular matching that is neither stable nor of maximum size [13].

An important algorithmic problem for our research is the problem of computing a popular matching containing a predefined set

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

of edges. A polynomial time algorithm for this problem exists if only a single edge has to be included in the matching [11] but the problem is NP-hard if at least two edges are forced [13]. However, this hardness heavily relies on the fact that some matching partners are unavailable. If the preference orders of the agents encompass the complete set of agents of the other side, then popular matchings of maximum weight can be computed in polynomial time [11]. Consequently, by setting appropriate weights, one can find popular matchings containing or preventing any subset of edges for complete instances.

Popular matchings have also been considered in related domains. Biró et al. [4] consider popular matchings for weak preferences, where their computation becomes NP-hard. However, in a house allocation setting, where one side of the agents corresponds to objects without preferences, popular matchings can be computed efficiently, even for weak preferences [1]. One can also relax the allowed input instances by considering the roommate setting where every pair of agents may be matched. Then, popularity already leads to computational hardness for strict preferences [13, 20].

In addition, some work considers a probabilistic variant of popularity called mixed popularity, where the output is a probability distribution over matchings [21]. Mixed popular matchings are guaranteed to exist by the Minimax Theorem, and can be computed efficiently if the output are matchings, even in the roommate setting [6, 21]. However, finding matchings in the support of popular outcomes becomes intractable in a coalition formation scenario where the output may contain coalitions of size three [6].

Robustness of outcomes with respect to perturbations of the input has also been studied in other scenarios of multiagent systems, such as voting [7, 14, 26]. There, robustness is commonly studied under the lens of bribery, i.e., deliberately influencing an election by changing its input. Then, a cost is incurred for modifying votes, often measured with respect to the swap distance of the original and modified votes [12].

More closely related, a series of papers study the complexity of finding stable matchings across multiple instances. Mai and Vazirani [22] initiate this stream of work and propose a polynomial-time algorithm if one agent shifts down a single alternative. Subsequently, this result was improved by Gangam et al. [17] for arbitrary changes of the preference order by a single agent and in very recent work even for arbitrary changes by all agents of one side [18]. Notably, the algorithmic approach of these papers is to exploit the combinatorial structure of the lattice of stable matchings. By contrast, a similar structure for popular matchings is unknown, and we develop an alternative technique. In addition, if all agents may change their preference lists, a computational intractability is also obtained for stable matchings [24]. Notably, in the reduction by Miyazaki and Okamoto [24], the number of agents that change their preference order is linear with respect to the total number of agents, and these agents apply extensive changes. This is a contrast to our hardness result, which holds even if only four agents perform the simple operation of shifting down an alternative.

Finally, some work on stable matchings considers other models in which multiple instances interact. Aziz et al. [2, 3] propose a model with uncertainty for the true preference relations. They ask for matchings that are possibly or necessarily stable, or stable with high probability. In addition, Boehmer et al. [5] study bribery for stable matchings, i.e., the strategic behaviour for achieving certain goals like forcing a given edge into a stable matching by deliberately manipulating preference orders.

3 PRELIMINARIES

In this section, we introduce our formal model.

An instance $\mathcal{I}$ of *matchings under preferences* (MP) consists of a bipartite graph $G^{\mathcal{I}} = (W \cup F, E^{\mathcal{I}})$, where the elements of $W \cup F$ are called *agents*. In addition, every agent $x \in W \cup F$ is equipped with a linear order $\succ_x^{\mathcal{I}}$, their so-called *preference order*, over $N_x^{\mathcal{I}} := \{y \in W \cup F\colon \{x, y\} \in E^{\mathcal{I}}\}$, i.e., the set of their neighbors in G. We usually refer to the sets W and F as *workers* and *firms*, respectively. Note that we have superscripts pointing to the instance for most of our notation because we will soon consider different instances in parallel. However, we might omit the superscript if the instance is clear from the context. Moreover, workers and firms are always identical across instances and we entirely omit superscripts for these.

Given a graph $G = (W \cup F, E)$, a *matching* is a subset $M \subseteq E$ of pairwise disjoint edges, i.e., $m \cap m' = \emptyset$ for all $m, m' \in M$. For a matching M, we call an agent $x \in W \cup F$ *matched* if there exists $m \in M$ with $x \in m$, and *unmatched*, otherwise. If x is matched, we denote their matching partner by $M(x)$.

Assume now that we are given an instance $\mathcal{I}$ of MP together with an agent $x \in W \cup F$ and two matchings M and M'. We say that x *prefers* M over M' if x is matched in M and unmatched in M', or if x is matched in both M and M' with $M(x) \succ_x M'(x)$.

The notion of popularity depends on a majority vote of the agents between matchings according to their preferences. Therefore, we define the following notation for a vote between matchings:

$$\text{vote}_x^{\mathcal{I}}(M, M') := \begin{cases} 1 & x \text{ prefers } M \text{ over } M', \\ -1 & x \text{ prefers } M' \text{ over } M, \\ 0 & \text{otherwise.} \end{cases}$$

Given a set of agents $N \subseteq W \cup F$, we define $\text{vote}_N^{\mathcal{I}}(M, M') := \sum_{x \in N} \text{vote}_x^{\mathcal{I}}(M, M')$. The *popularity margin* between M and M' is defined as $\Delta^{\mathcal{I}}(M, M') := \text{vote}_{W \cup F}^{\mathcal{I}}(M, M')$. Now, a matching M is called *popular* with respect to instance $\mathcal{I}$ if, for every matching M', it holds that $\Delta^{\mathcal{I}}(M, M') \geq 0$. In other words, a matching is popular if it does not lose a majority vote among the agents in an election against any other matching. Moreover, a matching M is called *stable* if for every edge $e = \{x, y\} \in E \setminus M$, it holds that x is matched and prefers $M(x)$ to y or y is matched and prefers $M(y)$ to x. As we discussed before, all stable matchings are popular.

We are interested in matchings that are popular across multiple instances. For this, we consider a pair of instances $(\mathcal{I}_A, \mathcal{I}_B)$ of MP where we assume that they are defined for the same set of workers and firms.

Given a pair of instances $(\mathcal{I}_A, \mathcal{I}_B)$, a matching is called a *robust popular matching* with respect to $\mathcal{I}_A$ and $\mathcal{I}_B$ if it is popular with respect to both $\mathcal{I}_A$ and $\mathcal{I}_B$ individually. Note that this implies that a robust popular matching is in particular a matching for both $\mathcal{I}_A$ and $\mathcal{I}_B$ and therefore a subset of the edge set of both underlying graphs. We are interested in the computational problem of computing robust popular matchings, specified more precisely as follows.

> RobustPopularMatching
> **Input:** Pair $(\mathcal{I}_A, \mathcal{I}_B)$ of instances of MP.
> **Question:** Does there exist a robust popular matching with respect to $\mathcal{I}_A$ and $\mathcal{I}_B$?

In particular, we consider the cases where the underlying graph or the underlying preferences remain the same across instances.

First, if the underlying graphs are the same, i.e., $G^{\mathcal{I}_A} = G^{\mathcal{I}_B}$, we say that $\mathcal{I}_B$ is a *perturbed instance* with respect to $\mathcal{I}_A$. Hence, a perturbed instance only differs with respect to the preference orders of the agents over the identical sets of neighbors. As a special case, we consider the case where agents simply push down a single alternative in their preference order. Given an agent $x \in W \cup F$ and two preference orders $\succ_x^{\mathcal{I}_A}, \succ_x^{\mathcal{I}_B}$ over N_x, we say that $\succ_x^{\mathcal{I}_B}$ evolves from $\succ_x^{\mathcal{I}_A}$ by a *downshift* if there exists an agent $y \in N_x$ such that

- for all agents $z, z' \in N_x \setminus \{y\}$, it holds that $z \succ_x^{\mathcal{I}_A} z'$ if and only if $z \succ_x^{\mathcal{I}_B} z'$, and
- for all agents $z \in N_x \setminus \{y\}$, it holds that $z \succ_x^{\mathcal{I}_A} y$ implies $z \succ_x^{\mathcal{I}_B} y$.

In other words, the preference order of x only changes by making y worse, and maintaining the order among all other agents. Moreover, we say that $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by *downshifts* if, for all agents $x \in W \cup F$ with $\succ_x^{\mathcal{I}_A} \neq \succ_x^{\mathcal{I}_B}$, it holds that $\succ_x^{\mathcal{I}_B}$ evolves from $\succ_x^{\mathcal{I}_A}$ by a downshift.

Second, we consider the case of identical preference orders. More formally, we say that $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by *altering availability* if, for every agent $x \in W \cup F$, there exists a preference order $\succ_x$ on $N_x^{\mathcal{I}_A} \cup N_x^{\mathcal{I}_B}$ such that for all $y, z \in N_x^{\mathcal{I}_A}$, it holds that $y \succ_x^{\mathcal{I}_A} z$ if and only if $y \succ_x z$ and for all $y, z \in N_x^{\mathcal{I}_B}$, it holds that $y \succ_x^{\mathcal{I}_B} z$ if and only if $y \succ_x z$. In other words, the underlying graphs of the two input instances may differ but the preferences for common neighbors are identical. As a special case, we say that $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by *reducing availability* if $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by altering availability and $E^{\mathcal{I}_B} \subseteq E^{\mathcal{I}_A}$.

Before presenting our results, we illustrate central concepts in an example.

Example 3.1. Consider an instance $\mathcal{I}_A$ of MP with $W = \{w_1, w_2, w_3, w_4\}$ and $F = \{f_1, f_2, f_3, f_4\}$ where the graph and preferences are defined as depicted in Figure 1. The instance contains a unique stable matching $M_1 = \{\{w_1, f_1\}, \{w_2, f_3\}, \{w_3, f_2\}, \{w_4, f_4\}\}$. Moreover, there exists another popular matching $M_2 = \{\{w_1, f_3\}, \{w_2, f_1\}, \{w_3, f_2\}, \{w_4, f_4\}\}$. Note that this matching is not stable, because of the edge $\{w_1, f_1\}$.

Now, consider the instance $\mathcal{I}_B$ that is obtained from instance $\mathcal{I}_A$ by having agent w_1 change their preferences to $f_2 \succ_{w_1}^{\mathcal{I}_B} f_3 \succ_{w_1}^{\mathcal{I}_B} f_1$, and leaving everything else the same. Hence, $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by a downshift of agent w_1. The unique popular (and therefore stable) matching in $\mathcal{I}_B$ is M_2. In particular, it therefore holds that $(\mathcal{I}_A, \mathcal{I}_B)$ is a Yes-instance of RobustPopularMatching. ◁

Our results make use of existing results in the literature. First, we consider StableMatching, the problem of computing a stable matching in a given instance of MP, which can be solved in polynomial time by the famous Deferred Acceptance Algorithm [15].

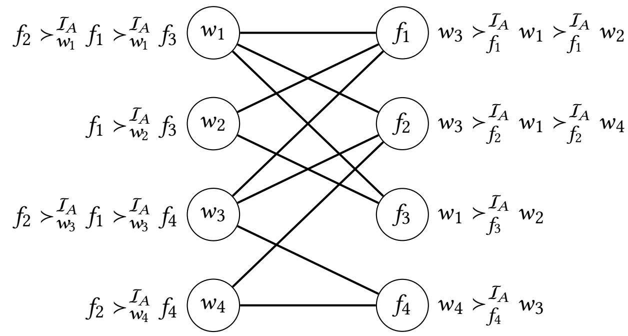

Figure 1: Instance $\mathcal{I}_A$ in Example 3.1. The perturbed instance $\mathcal{I}_B$ is obtained by having agent w_1 swap their preferences for f_1 and f_3.

Second, we consider PopularEdge, the problem of computing a popular matching in a given instance of MP containing a designated edge, or deciding that no such matching exists. This problem can also be solved in polynomial time [11].

4 RESULTS

In this section, we present our results.

4.1 Perturbations of One Agent

First, we consider instances of RobustPopularMatching with identical underlying graphs, where the perturbed instance only differs with respect to the preferences of a single agent. We will eventually show that there exists a polynomial-time algorithm that solves RobustPopularMatching under this restriction. For this, we perform two key steps. First, we define a set of hybrid instances, which allow us to answer if there exists a robust popular matching that contains a given edge. Second, we deal with the case of robust popular matchings where the agent with perturbed preferences remains unmatched. Combining these insights with known algorithmic and structural results about popular matchings, we obtain a polynomial-time algorithm.

We start by defining hybrid instances. Consider an instance $(\mathcal{I}_A, \mathcal{I}_B)$ of RobustPopularMatching where $\mathcal{I}_B$ only differs from $\mathcal{I}_A$ with respect to the preferences of agent x. Let $G = (W \cup F, E)$ be the underlying graph and consider an edge $e \in E$ with $x \in E$, say $e = \{x, y\}$. Define $P^A = \{z \in W \cup F : z \succ_x^{\mathcal{I}_A} y\}$ and $P^B = \{z \in W \cup F : z \succ_x^{\mathcal{I}_B} y\}$, i.e., P^A and P^B are the agents preferred to y by x in instance $\mathcal{I}_A$ and $\mathcal{I}_B$, respectively. Consider any linear order $\succ'$ of the neighbors N_x of x in G that satisfies $z \succ' y$ if $z \in P^A \cup P^B$, as well as $y \succ' z$ if $z \in N_x \setminus (P^A \cup P^B \cup \{y\})$. Hence, $\succ'$ is a preference order, where P^A and P^B are ordered arbitrarily at the top, then agent y, and finally all other neighbors of x in an arbitrary order.

The *hybrid instance* $\mathcal{H}_e$ of $(\mathcal{I}_A, \mathcal{I}_B)$ with respect to e is defined as the instance of MP where $\succ_z^{\mathcal{H}_e}$ is equal to $\succ_z^{\mathcal{I}_A}$ for all $z \neq x$ and $\succ_x^{\mathcal{H}_e}$ is equal to $\succ'$. Note that we illustrate hybrid instances in Example 4.7, where we also illustrate our main proof. We now prove two important lemmas that create a correspondence of popular matchings in $\mathcal{H}_e$ and robust popular matchings for $(\mathcal{I}_A, \mathcal{I}_B)$. The first lemma considers popular matchings in $\mathcal{H}_e$ containing e.

Lemma 4.1. *Let M be a matching and $e \in M$ with $x \in e$. If M is popular in $\mathcal{H}_e$, then it is popular in both $\mathcal{I}_A$ and $\mathcal{I}_B$.*

Proof. Let M be a matching and $e \in M$ with $x \in e$. Assume that M is popular in $\mathcal{H}_e$. As the definition of the hybrid instance $\mathcal{H}_e$ is symmetric with respect to $\mathcal{I}_A$ and $\mathcal{I}_B$, we only prove that M is popular in $\mathcal{I}_A$.

Let M' be any other matching. We determine the popularity margin between M and M' in $\mathcal{I}_A$ by considering the votes of all agents. First, let $z \in (W \cup F) \setminus \{x\}$. Then, since the preferences of z are identical in $\mathcal{H}_e$ and $\mathcal{I}_A$, it holds that $\text{vote}_z^{\mathcal{I}_A}(M, M') = \text{vote}_z^{\mathcal{H}_e}(M, M')$. Second, let us consider the vote of agent x. By construction of the hybrid instance, for all agents $z \in N_x$, it holds that $y \succ_x^{\mathcal{I}_A} z$ whenever $y \succ_x^{\mathcal{H}_e} z$. Hence, since $M(x) = y$, we can conclude that $\text{vote}_x^{\mathcal{I}_A}(M, M') \geq \text{vote}_x^{\mathcal{H}_e}(M, M')$.

Combining these two insights, we obtain

$$\Delta^{\mathcal{I}_A}(M, M') = \sum_{z \in W \cup F} \text{vote}_z^{\mathcal{I}_A}(M, M') \geq \sum_{z \in W \cup F} \text{vote}_z^{\mathcal{H}_e}(M, M') = \Delta^{\mathcal{H}_e}(M, M') \geq 0.$$

The last inequality holds because of the popularity of M in $\mathcal{H}_e$. Hence, M is popular in $\mathcal{I}_A$. □

Lemma 4.2. *Let M be a matching and $e \in M$ with $x \in e$. If M is popular in both $\mathcal{I}_A$ and $\mathcal{I}_B$, then it is popular in $\mathcal{H}_e$.*

Proof. Let M be a matching and $e \in M$ with $x \in e$. Assume that M is popular in both $\mathcal{I}_A$ and $\mathcal{I}_B$.

Let M' be any other matching. We will compute the popularity margin between M and M' in $\mathcal{H}_e$. Let $z \in (W \cup F) \setminus \{x\}$. As in the proof of the previous lemma, since the preferences of z are identical in $\mathcal{H}_e$, $\mathcal{I}_A$, and $\mathcal{I}_B$, it holds that $\text{vote}_z^{\mathcal{H}_e}(M, M') = \text{vote}_z^{\mathcal{I}_A}(M, M') = \text{vote}_z^{\mathcal{I}_B}(M, M')$.

We make a case distinction with respect to the vote of agent x. If $\text{vote}_x^{\mathcal{H}_e}(M, M') = 1$, then the previous observation immediately implies that $\Delta^{\mathcal{H}_e}(M, M') \geq \Delta^{\mathcal{I}_A}(M, M') \geq 0$. If $\text{vote}_x^{\mathcal{H}_e}(M, M') = 0$, i.e., $M'(x) = M(x)$, then $\Delta^{\mathcal{H}_e}(M, M') = \Delta^{\mathcal{I}_A}(M, M') \geq 0$. If $\text{vote}_x^{\mathcal{H}_e}(M, M') = -1$, then $M'(x) \succ_x^{\mathcal{H}_e} M(x)$ where $M(x) = y$, and therefore $M'(x) \in P^A \cup P^B$. Without loss of generality, we may assume that $M'(x) \in P^A$. Then, by definition, $M'(x) \succ_x^{\mathcal{I}_A} M(x)$, and therefore $\text{vote}_x^{\mathcal{I}_A}(M, M') = -1$. Combining this with the votes of the other agents, it follows that $\Delta^{\mathcal{H}_e}(M, M') = \Delta^{\mathcal{I}_A}(M, M') \geq 0$. Since we have exhausted all cases, we conclude that $\Delta^{\mathcal{H}_e}(M, M') \geq 0$. As M' was an arbitrary matching, it follows that M is popular. □

Combining Lemmas 4.1 and 4.2, we can find robust popular matchings containing a specific edge by solving an instance of PopularEdge.

Corollary 4.3. *The instance $(\mathcal{I}_A, \mathcal{I}_B)$ contains a robust popular matching containing edge e if and only if PopularEdge for the hybrid instance with designated edge e is a Yes-instance.*

It remains to figure out whether there exist robust popular matchings that leave the agents with perturbed preferences unmatched. For this, we make another observation.

Lemma 4.4. *Let M be a matching that leaves agent x unmatched. Then, M is popular for $\mathcal{I}_A$ if and only if M is popular in $\mathcal{I}_B$.*

Proof. Let M be a matching that leaves x unmatched. Then, for every matching M', it holds that $\text{vote}_x^{\mathcal{I}_A}(M, M') = \text{vote}_x^{\mathcal{I}_B}(M, M')$. Hence, since x is the only agent to perturb their preferences, it follows that $\Delta^{\mathcal{I}_A}(M, M') = \Delta^{\mathcal{I}_B}(M, M')$. Therefore, as M' was an arbitrary matching, it holds that M is popular for $\mathcal{I}_A$ if and only if M is popular for $\mathcal{I}_B$. □

As a consequence, we can tackle this case by finding a popular matching in $\mathcal{I}_A$ that leaves x unmatched, or decide that no such matching exists. This problem has a surprisingly easy solution: It suffices to compute any stable matching. The key insight is captured in the next lemma by Cseh and Kavitha [11], a lemma that resembles the fundamental Rural Hospitals Theorem for stable matchings [16, 25].

Lemma 4.5 (Cseh and Kavitha [11]). *If an agent is unmatched in some popular matching, then it is unmatched in all stable matchings.*

We combine all our insights to state an algorithm for RobustPopularMatching if the perturbed input instance only differs with respect to the preference order of one agent.

Algorithm 1 RobustPopularMatching for changes of one agent

Input: Instance $(\mathcal{I}_A, \mathcal{I}_B)$ of RobustPopularMatching where only one agent x perturbs their preference order
Output: Robust popular matching for $(\mathcal{I}_A, \mathcal{I}_B)$ or statement that no such matching exists

```
1: Compute stable matching M for I_A
2: if M leaves x unmatched then return M.
3: end if
4: for e ∈ E with x ∈ e do
5:     if there exists a popular matching M for H_e with e ∈ M then return M
6:     end if
7: end for
8: return "No robust popular matching exists"
```

The algorithm first checks a stable matching to attempt finding a robust popular matching that leaves x unmatched. Then, it checks the hybrid instances to search for robust popular matchings where x is matched. The correctness and running time of this algorithm are captured in the main theorem of this section.

Theorem 4.6. *RobustPopularMatching can be solved in polynomial time if the perturbed input instance only differs with respect to the preference order of one agent.*

Proof. The polynomial running time follows because StableMatching and PopularEdge can be solved in polynomial time [11, 15].

Let us consider the correctness of Algorithm 1. For this we show that Algorithm 1 returns a matching if and only if there exists a robust popular matching in the considered instance. First, note that if Algorithm 1 returns a matching in line 2, then it returns a popular matching for $\mathcal{I}_A$ because stable matchings are popular [19]. Hence, by Lemma 4.4, it returns a robust popular matching in this case. Moreover, if Algorithm 1 returns a matching in line 5, it is a robust popular matching according to Corollary 4.3. Hence, if Algorithm 1 returns a matching, then it is a robust popular matching.

Conversely, assume that there exists a robust popular matching M. Assume first that M leaves x unmatched. Then, by Lemma 4.5, every stable matching leaves x unmatched, and Algorithm 1 returns a matching in line 2. In addition, if x is matched in M by an edge e, then, by Corollary 4.3, $\mathcal{H}_e$ contains a popular matching containing e. Hence, Algorithm 1 returns a matching in line 5. □

Example 4.7. We illustrate the proof as well as hybrid instances by continuing Example 3.1. In this example, the two input instances $\mathcal{I}_A$ and $\mathcal{I}_B$ only differ with respect to the preferences of agent w_1.

Since the stable matching M_1 for $\mathcal{I}_A$ matches w_1, we can conclude that there exists no robust popular matching that leaves w_1 unmatched. Hence, we have to consider the hybrid instances $\mathcal{H}_e$ for $e \in \{\{w_1, f_1\}, \{w_1, f_2\}, \{w_1, f_3\}\}$, i.e., all edges incident to w_1. Interestingly, $\mathcal{I}_A$ can serve as a hybrid instance for $e \in \{\{w_1, f_2\}, \{w_1, f_3\}\}$ and $\mathcal{I}_B$ can serve as a hybrid instance for $e = \{w_1, f_1\}$. In fact, this incidence generalizes: Whenever $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by a downshift of agent y in the preference order of agent x, then $\mathcal{I}_B$ serves as a hybrid instance for $e = \{x, y\}$, and $\mathcal{I}_A$ for all other edges containing x.

Now, since M_2 is popular in $\mathcal{I}_A$ for the hybrid instance where $e = \{w_1, f_3\}$, Algorithm 1 finds the robust popular matching for $(\mathcal{I}_A, \mathcal{I}_B)$. ◁

Finally, by straightforward extensions of the techniques developed in this section, we can generalize our result for the case of more than two instances that all differ only with respect to the preferences of one agent x. To find a robust popular matching containing a specific edge $e = \{x, y\}$, we define the preference order of x in a generalized hybrid instance by putting the agents preferred to y by x in *any* input instance above y. This ensures that whenever we contest the popularity of a matching in the hybrid instance with a matching where x receives a better partner z, then the popularity of this matching is also contested in the input instances that have z ranked above y.

Theorem 4.8. *There exists a polynomial-time algorithm for the following problem: Given a collection of MP instances $(\mathcal{I}_1, \dots, \mathcal{I}_k)$, which are all defined for the same underlying graph and differ only with respect to the preferences of a single agent, does there exist a matching that is popular for $\mathcal{I}_i$ for all $1 \leq i \leq k$?*

However, once we consider changes by more than one agent, Corollary 4.3 breaks down. In fact, it ceases to hold even in the case where two agents of the same class, e.g., workers, each swap their preferences over two adjacent agents. We provide such an example in the full version of the paper [8].

4.2 Perturbation by Four Downshifts

In this section, we continue the consideration of instance pairs with the identical underlying graph. While we have previously seen a polynomial-time algorithm for solving RobustPopularMatching if the perturbed instance only differs by a single agent permuting their preferences, we now allow several agents to change their preference orders. In this case, we obtain a computational intractability if four agents are allowed to permute their preferences, even if the only allowed changes are downshifts.

Our proof idea is to reduce from the problem of finding a popular matching where a designated set of two edges is forbidden. In the reduced instance of RobustPopularMatching, we use one of the instances to represent the input instance and contain all originally popular matchings. In the second instance, we perform a downshift by the four agents involved in the two designated edges to prevent these matchings. For this, one cannot simply move down their respective partners in the original instance, because this might have no effect, for example, if a designated edge represents the only available option for one of its endpoints. Instead, we introduce auxiliary agents that represent the case of agents being unmatched. Then, moving the matching partners in the designated edges below the auxiliary agents has the desired effect.

Theorem 4.9. *RobustPopularMatching is* NP*-complete even if the perturbed instance only differs by a downshift of four agents.*

Proof. First, note that membership in NP is straightforward. A robust popular matching with respect to two given input instances of MP serves as a polynomial-size certificate for a Yes-instance. We can verify it by simply checking whether the matching is popular in both instances in polynomial time [4, Theorem 9].

For NP-hardness, we perform a reduction from the ForbiddenEdge problem [13]. The input of this problem is an instance of MP on a graph $G = (W \cup F, E)$ and two designated disjoint edges $e, e' \in E$. An instance is a Yes-instance if and only if there exists a popular matching M with $\{e, e'\} \cap M = \emptyset$. This problem is known to be NP-hard [13, Theorem 4.1].[1]

We are ready to define the reduction. Consider an instance of ForbiddenEdge given by an instance $\mathcal{I}$ of MP on the graph $(W \cup F, E)$ together with two designated edges $e, e' \in E$, say $e = \{a, b\}$ and $e' = \{c, d\}$, where $a, c \in W$ and $b, d \in F$. We denote $C = \{a, b, c, d\}$ as the *critical* agents in the source instance, and for $x \in C$, we denote by $f(x)$ their forbidden partner, for instance, $f(a) = b$.

The idea of the reduced instance is to enhance the source instance by a set of auxiliary agents for the agents contained in the designated edges. For each agent $x \in C$, we add a *last agent* ℓ_x to which x can be matched when they would be unmatched as well as an agent r_x, which can be matched with ℓ_x in case that x is already matched. A downshift of the agents in the designated edges in the perturbed instance then ensures that they are forbidden in popular matchings.

We start by defining the graph of the reduced instance more precisely. Let $W' = W \cup \{r_a, \ell_b, r_c, \ell_d\}$ and $F' = F \cup \{\ell_a, r_b, \ell_c, r_d\}$, and $E' = E \cup \{\{x, \ell_x\}, \{\ell_x, r_x\} : x \in C\}$. Define $G' = (W' \cup F', E')$. For $x \in C$ we define $S_x = \{x, \ell_x, r_x\}$.

The preferences are mostly inherited from the source instance. For $i \in (W \cup F) \setminus C$, we define $>_i^{\mathcal{I}_A}$ and $>_i^{\mathcal{I}_B}$ as identical to $>_i^{\mathcal{I}}$. Now let $x \in C$. The preferences for the agents in S_x are indicated in Figure 2. First, the agent r_x only has one neighbor in G' and therefore possesses the trivial preference order only ranking ℓ_x. Second, we have $x >_{\ell_x}^{\mathcal{I}_A} r_x$ and $x >_{\ell_x}^{\mathcal{I}_B} r_x$.

[1]The validity of the restriction that the two designated edges can be assumed to be disjoint immediately follows from the proof by Faenza et al. [13] and simplifies our reduction a bit.

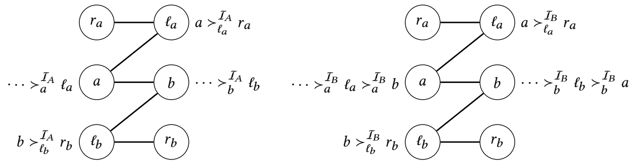

Figure 2: Key gadget of the reduced instances in the proof of Theorem 4.9. The preferences in $\mathcal{I}_A$ and $\mathcal{I}_B$ are described in the left and right picture, respectively.

The only preference order that differs is for the agent x. For all $y, z \in N'_x \setminus \{f(x), \ell_x\}$, we define $y \succ^{\mathcal{I}_A}_x z$ and $y \succ^{\mathcal{I}_B}_x z$ if and only if $y \succ^{\mathcal{I}}_x z$. Moreover, we define $y \succ^{\mathcal{I}_A}_x \ell_x$ and $y \succ^{\mathcal{I}_B}_x \ell_x$. The difference in the preferences is concerning the agent $f(x)$. For all $y \in N'_x \setminus \{f(x), \ell_x\}$, we have $y \succ^{\mathcal{I}_A}_x f(x)$ if and only if $y \succ^{\mathcal{I}}_x f(x)$ and, we have $f(x) \succ^{\mathcal{I}_A}_x \ell_x$. However, in $\mathcal{I}_B$, the forbidden partner $f(x)$ is pushed to the bottom of the preference order. For all $y \in N'_x \setminus \{f(x)\}$, we have $y \succ^{\mathcal{I}_B}_x f(x)$.

We are ready to prove the correctness of the reduction. To this end, we will show that $\mathcal{I}$ contains a popular matching M with $M \cap \{e, e'\} = \emptyset$ if and only if the reduced instance contains a matching popular for both $\mathcal{I}_A$ and $\mathcal{I}_B$.

$\Longrightarrow$ Assume first that $\mathcal{I}$ contains a popular matching M with $M \cap \{e, e'\} = \emptyset$. Let $U \subseteq C$ be the subset of unmatched agents among C with respect to M. We define the matching $M' = M \cup \{\{x, \ell_x\} : x \in U\} \cup \{\{\ell_x, r_x\} : x \in C \setminus U\}$. Assume for contradiction that M' is not popular for instance $\mathcal{I}_A$ and that there exists a matching $\hat{M}'$ in G' with $\Delta^{\mathcal{I}_A}(\hat{M}', M') > 0$.

Define $\hat{M} = \{e \in \hat{M}' : e \subseteq W \cup F\}$, i.e., the matching $\hat{M}'$ restricted to agents present in the source instance. We will argue that $\Delta^{\mathcal{I}}(\hat{M}, M) > 0$. Let $i \in (W \cup F) \setminus C$. Then, it holds that $\hat{M}(i) = \hat{M}'(i)$ if i is matched in $\hat{M}'$ or i is unmatched in both. Since the preferences of i are identical in $\mathcal{I}$ and $\mathcal{I}_A$, we have that $\mathrm{vote}^{\mathcal{I}}_i(\hat{M}, M) = \mathrm{vote}^{\mathcal{I}_A}_i(\hat{M}', M')$. We refer to this as Observation ($\diamond$).

Now let $x \in C$. We claim that $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) \geq \mathrm{vote}^{\mathcal{I}_A}_{S_x}(\hat{M}', M')$ and we refer to this claim as Observation ($\diamond\diamond$). We prove this by a case distinction with respect to the matching partner of ℓ_x in M'.

First, assume that $\{x, \ell_x\} \in M'$. This means that x is unmatched in M. If $\hat{M}'(x) \in W \cup F$, then x is also matched in $\hat{M}$ and $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = 1 = \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M')$. Moreover, then $\mathrm{vote}^{\mathcal{I}_A}_{\ell_x}(\hat{M}', M') = -1$ and $\mathrm{vote}^{\mathcal{I}_A}_{r_x}(\hat{M}', M') \leq 1$ and the claim is true. If $\{x, \ell_x\} \in \hat{M}'$, then x is unmatched in $\hat{M}$ and therefore $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = 0 = \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M')$. Moreover, then $\mathrm{vote}^{\mathcal{I}_A}_{\ell_x}(\hat{M}', M') = 0$ and r_x is unmatched, hence $\mathrm{vote}^{\mathcal{I}_A}_{r_x}(\hat{M}', M') = 0$. Again, the claim is true. Finally, if x is unmatched in $\hat{M}'$, then $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = 0 > -1 = \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M')$. The claim follows because $\mathrm{vote}^{\mathcal{I}_A}_{\ell_x}(\hat{M}', M') = -1$ and $\mathrm{vote}^{\mathcal{I}_A}_{r_x}(\hat{M}', M') \leq 1$.

Second, assume that $\{\ell_x, r_x\} \in M'$, which implies that x is matched in M. If x is matched in $\hat{M}$, then x compares exactly the same partners in $\mathcal{I}$ and $\mathcal{I}_A$ and therefore $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M')$. In this case, ℓ_x and r_x cannot improve in $\hat{M}'$ compared to M', and the claim is true. If $\{x, \ell_x\} \in \hat{M}'$, then x is worse off in $\hat{M}'$ compared to M', but unmatched in $\hat{M}$. Hence, $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = -1 = \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M')$. In this case $\mathrm{vote}^{\mathcal{I}_A}_{\ell_x}(\hat{M}', M') = 1$ but $\mathrm{vote}^{\mathcal{I}_A}_{r_x}(\hat{M}', M') = -1$, and the claim is true. Finally if x is unmatched in $\hat{M}'$, then $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = -1 = \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M')$, and none of ℓ_x and r_x can have improved. This proves the final case of the claim.

Combining Observations ($\diamond$) and ($\diamond\diamond$), we obtain

$$
\begin{aligned}
\Delta^{\mathcal{I}}(\hat{M}, M) &= \sum_{x \in (W \cup F) \setminus C} \mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) + \sum_{x \in C} \mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) \\
&\geq \sum_{x \in (W \cup F) \setminus C} \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M') + \sum_{x \in C} \mathrm{vote}^{\mathcal{I}_A}_{S_x}(\hat{M}', M') \\
&= \Delta^{\mathcal{I}_A}(\hat{M}', M') > 0.
\end{aligned}
$$

This contradicts the popularity of M. Hence, we have derived a contradiction and M' is popular for $\mathcal{I}_A$.

Now, since $\mathcal{I}_B$ only differs from $\mathcal{I}_A$ by a downshift of agents that are not matching partners, we have that for every matching $\hat{M}'$ and every agent $x \in W' \cup F'$, it holds that $\mathrm{vote}^{\mathcal{I}_B}_x(M', \hat{M}') \geq \mathrm{vote}^{\mathcal{I}_A}_x(M', \hat{M}')$. Therefore, $\Delta^{\mathcal{I}_B}(M', \hat{M}') \geq \Delta^{\mathcal{I}_A}(M', \hat{M}')$. Hence, the popularity of M' in $\mathcal{I}_B$ follows from the popularity of M' in $\mathcal{I}_A$. This concludes the proof of the first implication.

$\Longleftarrow$ Conversely, assume that M' is a matching that is popular for both $\mathcal{I}_A$ and $\mathcal{I}_B$. Define the matching $M = \{e \in M' : e \subseteq W \cup F\}$. We will first show that M is popular in the source instance and subsequently that it does not contain the forbidden edges.

Assume for contradiction that there exists a matching $\hat{M}$ on $\mathcal{I}$ with $\Delta^{\mathcal{I}}(\hat{M}, M) > 0$. Let $U \subseteq C$ be the subset of unmatched agents among C with respect to $\hat{M}$ and consider the matching $\hat{M}' = \hat{M} \cup \{\{x, \ell_x\} : x \in U\} \cup \{\{\ell_x, r_x\} : x \in C \setminus U\}$.

We will show that $\Delta^{\mathcal{I}_A}(\hat{M}', M') > 0$. Let $x \in (W \cup F) \setminus C$. Then, it holds that $\hat{M}(x) = \hat{M}'(x)$ if x is matched in $\hat{M}'$ or x is unmatched in both. Since the preferences of x are identical in $\mathcal{I}$ and $\mathcal{I}_A$, we have that $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = \mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M')$. We refer to this as Observation ($\star$).

Now, let $x \in C$. We claim that $\mathrm{vote}^{\mathcal{I}_A}_{S_x}(\hat{M}', M') \geq \mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M)$ and we refer to this claim as Observation ($\star\star$). We make a case distinction with respect to the matching status of agent x.

First, if x is matched in M and $\hat{M}$, then, since the preferences of x for agents in $W \cup F$ coincide in $\mathcal{I}$ and $\mathcal{I}_A$, it holds that $\mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M') = \mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M)$. Moreover, since $\{\ell_x, r_x\} \in \hat{M}'$, none of these agents can be worse off than in M', i.e., $\mathrm{vote}^{\mathcal{I}_A}_{\ell_x}(\hat{M}', M') \geq 0$ and $\mathrm{vote}^{\mathcal{I}_A}_{r_x}(\hat{M}', M') \geq 0$. Together, the claim follows for this case.

Second, assume that x is unmatched in M but matched in $\hat{M}$. Then, x is either unmatched in M' or matched with ℓ_x. In both cases, their partner in $\hat{M}'$ is preferred to their situation in M' and it holds that $\mathrm{vote}^{\mathcal{I}_A}_x(\hat{M}', M') = 1$. If $\{\ell_x, r_x\} \in M'$, then ℓ_x and r_x have the same partners in $\hat{M}'$ and M' and the claim follows. Otherwise, $\mathrm{vote}^{\mathcal{I}_A}_{r_x}(\hat{M}', M') = 1$ and $\mathrm{vote}^{\mathcal{I}_A}_{\ell_x}(\hat{M}', M') \geq -1$, and the claim is also true.

Third, assume that x is matched in M but unmatched in $\hat{M}$. Then, $\mathrm{vote}^{\mathcal{I}}_x(\hat{M}, M) = -1$. Moreover, ℓ_x cannot have been matched with

x in M' and it follows that $\text{vote}_{\ell_x}^{\mathcal{I}_A}(\hat{M}', M') = 1$. The claim follows since $\text{vote}_{x}^{\mathcal{I}_A}(\hat{M}', M') \geq -1$ and $\text{vote}_{r_x}^{\mathcal{I}_A}(\hat{M}', M') \geq -1$.

Finally, assume that x is unmatched in both M and $\hat{M}$. Then, $\text{vote}_{x}^{\mathcal{I}}(\hat{M}, M) = 0$. If $\{x, \ell_x\} \in M'$, then all agents in S_x have their identical partners in M' and $\hat{M}'$ and the claim follows. Otherwise, x is unmatched in M' and matched in $\hat{M}'$ and therefore $\text{vote}_{x}^{\mathcal{I}_A}(\hat{M}', M') = 1$. In addition, $\text{vote}_{\ell_x}^{\mathcal{I}_A}(\hat{M}', M') = 1$. The claim follows since $\text{vote}_{r_x}^{\mathcal{I}_A}(\hat{M}', M') \geq -1$. Since we have exhausted all cases, Observation ($\star\star$) follows.

Combining Observations ($\star$) and ($\star\star$), we obtain

$$\begin{aligned}\Delta^{\mathcal{I}_A}(\hat{M}', M') &= \sum_{x \in (W \cup F) \setminus C} \text{vote}_{x}^{\mathcal{I}_A}(\hat{M}', M') + \sum_{x \in C} \text{vote}_{S_x}^{\mathcal{I}_A}(\hat{M}', M') \\ &\geq \sum_{x \in (W \cup F) \setminus C} \text{vote}_{x}^{\mathcal{I}}(\hat{M}, M) + \sum_{x \in C} \text{vote}_{x}^{\mathcal{I}}(\hat{M}, M) \\ &= \Delta^{\mathcal{I}}(\hat{M}, M) > 0.\end{aligned}$$

This contradicts the popularity of M'. Hence, we have derived a contradiction and M is popular for $\mathcal{I}$.

It remains to show that M does not contain e and e'. We will show this fact for e. The proof for e' is completely analogous. Assume for contradiction that $e \in M$, which implies that $e \in M'$.

Define the matching $\hat{M}' = \{g \in M' : g \subseteq (W' \cup F') \setminus (S_a \cup S_b)\} \cup \{\{a, \ell_a\}, \{b, \ell_b\}\}$. In other words, $\hat{M}'$ differs from M' by dissolving the edge $\{a, b\}$ and potentially edges $\{\ell_a, r_a\}$ and $\{\ell_b, r_b\}$, and by creating edges of a and b with ℓ_a and ℓ_b, respectively. We will compare $\hat{M}'$ with M' in $\mathcal{I}_B$. Let $x \in \{a, b\}$. Then, $\text{vote}_{x}^{\mathcal{I}_B}(\hat{M}', M') = 1$ because they improved from their worst to their second-worst partner. Moreover, $\text{vote}_{\ell_x}^{\mathcal{I}_B}(\hat{M}', M') = 1$ because ℓ_x is matched with their most preferred matching partner in $\hat{M}'$ but not in M'. In addition, all agents $y \in (W' \cup F') \setminus (S_a \cup S_b)$ are matched to the same agent in both M' and $\hat{M}'$, or unmached in both. Hence, $\text{vote}_{y}^{\mathcal{I}_B}(\hat{M}', M') = 0$. Together, $\Delta^{\mathcal{I}_B}(\hat{M}', M') = \sum_{z \in S_a \cup S_b} \text{vote}_{z}^{\mathcal{I}_B}(\hat{M}', M') = 4 + \text{vote}_{r_a}^{\mathcal{I}_B}(\hat{M}', M') + \text{vote}_{r_b}^{\mathcal{I}_B}(\hat{M}', M') \geq 2 > 0$. Hence, M' is not popular for $\mathcal{I}_B$. This is a contradiction and hence $e \notin M'$, which concludes the proof. □

4.3 Unpopular Agents

We continue the consideration of instances of RobustPopularMatching with a common underlying graph, but from a different angle. In this section, we consider agents that are not matched by any popular matching. We refer to such an agent as an *unpopular agent*. All other agents are called *popular agents*. Given an instance $\mathcal{I}$ of MP, let $\mathcal{U}^{\mathcal{I}}$ denote the set of unpopular agents in $\mathcal{I}$. The consideration of unpopular agents leads to a class of instances of RobustPopularMatching that are trivially Yes-instances because popular matchings are maintained.

Proposition 4.10. *Consider an instance $(\mathcal{I}_A, \mathcal{I}_B)$ of RobustPopularMatching where only the preference orders of agents in $\mathcal{U}^{\mathcal{I}_A}$, i.e., of unpopular agents in $\mathcal{I}_A$, differ in the perturbed instance. Then, $(\mathcal{I}_A, \mathcal{I}_B)$ is a Yes-instance of RobustPopularMatching.*

Proof. Let $(\mathcal{I}_A, \mathcal{I}_B)$ be an instance of RobustPopularMatching where $\mathcal{I}_A$ and $\mathcal{I}_B$ only differ with respect to the preference orders of agents in $\mathcal{U}^{\mathcal{I}_A}$. Let M be a popular matching in $\mathcal{I}_A$. We claim that M is also popular for $\mathcal{I}_B$.

Let M' be any other matching. Let $x \in (W \cup F) \setminus \mathcal{U}^{\mathcal{I}_A}$ be a popular agent. Then, because the preferences of x are the same in both instances, $\text{vote}_{x}^{\mathcal{I}_B}(M', M) = \text{vote}_{x}^{\mathcal{I}_A}(M', M)$. Now, let $x \in \mathcal{U}^{\mathcal{I}_A}$. Since x is unmatched in M, x votes in favor of M' in both $\mathcal{I}_A$ and $\mathcal{I}_B$ if x is matched in M' and is indifferent between the two matchings if x remains unmatched. Hence, once again $\text{vote}_{x}^{\mathcal{I}_B}(M', M) = \text{vote}_{x}^{\mathcal{I}_A}(M', M)$. Together, $\Delta^{\mathcal{I}_B}(M', M) = \Delta^{\mathcal{I}_A}(M', M)$. Hence, the popularity of M in $\mathcal{I}_B$ follows from the popularity of M in $\mathcal{I}_A$. □

Put differently, the computation of robust matchings is not sensitive to perturbances of agents that do not matter to popularity in the first place. Notably, the set of unpopular agents can be computed efficiently: We can compute their complement, i.e., the set of popular agents, by simply checking an instance of PopularEdge for every available edge. Moreover, like for perturbations of one agent in Theorem 4.8, Proposition 4.10 extends to multiple instances if these only differ with respect to perturbances of the preferences of the unpopular agents in one of these instances.

4.4 Reduced Availability

We turn to the consideration of RobustPopularMatching for the case of alternated availability, i.e., the underlying graph may change while maintaining preference orders among common edges. In particular, we consider the special case where the underlying graph is complete. To this end, an instance $\mathcal{I}$ of MP is said to be *complete* if $G^{\mathcal{I}}$ is the complete bipartite graph on vertex set $W \cup F$, i.e., the edge set is the Cartesian product of the set of workers and firms $E^{\mathcal{I}} = W \times F$. Note that if one of the MP instances of a RobustPopularMatching instance is complete, then alternated availability is identical to reduced availability. Our first result is an efficient algorithm for this case.

Proposition 4.11. *RobustPopularMatching can be solved in polynomial time for input instances $(\mathcal{I}_A, \mathcal{I}_B)$ where $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by reducing availability and $\mathcal{I}_A$ is complete.*

Proof. We show how to solve the problem by solving a maximum weight popular matching problem. Consider an instance $\mathcal{I}$ for MP and assume that we are given a weight function $w : E^{\mathcal{I}} \to \mathbb{Q}$. The *weight* of a matching is defined as $w(M) := \sum_{e \in M} w(e)$. It is known that the problem of computing a matching of maximum weight among popular matchings can be solved in polynomial time for complete instances of MP [11].

Now, consider an instance $(\mathcal{I}_A, \mathcal{I}_B)$ for RobustPopularMatching where $\mathcal{I}_B$ evolves from $\mathcal{I}_A$ by reducing availability and $\mathcal{I}_A$ is complete. We define the weight function $w : E^{\mathcal{I}_A} \to \{-1, 0\}$ by $w(e) = 0$ if $e \in E^{\mathcal{I}_B}$ and $w(e) = -1$, otherwise.

We claim that $(\mathcal{I}_A, \mathcal{I}_B)$ is a Yes-instance of RobustPopularMatching if and only if the maximum weight popular matching in $\mathcal{I}_A$ with respect to w has a weight of 0.

First, assume that M is a popular matching for both $\mathcal{I}_A$ and $\mathcal{I}_B$. Then, $M \subseteq E^{\mathcal{I}_B}$ and M is a popular matching for $\mathcal{I}_A$ with $w(M) = 0$.

Conversely, if M is a popular matching for $\mathcal{I}_A$ with $w(M) = 0$. Then, $M \subseteq E^{\mathcal{I}_B}$. Moreover, any other matching for $\mathcal{I}_B$ is also a matching in $\mathcal{I}_A$ with the identical popularity margin. Hence, the popularity of M for $\mathcal{I}_B$ follows from the popularity of M for $\mathcal{I}_A$. □

Interestingly, we can still extend Proposition 4.11 to the case of multiple instances. As long as one instance is complete, all other instances may differ by arbitrary altered availability. For a proof, we can simply adjust the weight function in the proof of Proposition 4.11 to be 0 for edges present in *all* instances. However, the restriction that $\mathcal{I}_A$ is a complete instance is vital for Proposition 4.11. As we show next, we obtain a computational intractability if we drop this assumption. We provide the proof in the full version [8].

Proposition 4.12. *RobustPopularMatching is* NP*-complete for input instances* $(\mathcal{I}_A, \mathcal{I}_B)$ *where* $\mathcal{I}_B$ *evolves from* $\mathcal{I}_A$ *by reducing availability.*

Notably, by inspection of the proof, RobustPopularMatching is already NP-complete if the input instances only differ by reducing availability where two edges are removed. By contrast, if only one edge is removed, the problem is equivalent to computing a popular matching with a single forbidden edge. This problem can be solved in polynomial time [13].

4.5 Robust Popular Matchings in Related Models

We conclude our result section by discussing robustness of popularity in related models.

First, robustness of matchings can be defined for other models of popularity. As mentioned earlier, there exists the concept of strongly popular matchings, which have a strictly positive popularity margin against any other matching. Since strongly popular matchings are unique and can be computed in polynomial time [4], robust strongly popular matchings can also be computed in polynomial time, whenever they exist: One can simply check if strongly popular matchings exist in all input instances and compare them.

Second, one can consider popularity for mixed matchings, which are probability distributions over deterministic matchings, and popularity is then defined as popularity in expectation [21]. Popular mixed matchings correspond to the points of a tractable polytope for which feasible points can be identified in polynomial time. One can intersect the polytopes for multiple instances and still obtain a tractable polytope. This approach yields a polynomial time algorithm to solve RobustPopularMatching for mixed matchings and can even be applied for roommate games. In these games, the input graph is not required to be bipartite anymore and the linear programming method can still be applied [6]. Notably, this approach cannot be used to determine deterministic matchings. Even if the polytopes for all input instances are integral, the intersection of the polytopes may be nonempty but not contain integral points anymore. We discuss technical details concerning mixed popularity including such an example in the full version of the paper [8].

Finally, one can consider popularity for more general input instances. However, this quickly leads to intractabilities because the existence of popular matchings may not be guaranteed any more. For example, it is NP-hard to decide whether a popular matching exists if we consider roommate games [13] or if we have bipartite graphs but weak preferences [4]. These results immediately imply NP-hardness of RobustPopularMatching because one can simply duplicate the source instance, and a robust popular matching exists if and only if the source instance admits a popular matching.

5 CONCLUSION

We have initiated the study of robustness for popular matchings by considering the algorithmic question of determining a popular matching in the intersection of two given instances of matching under preferences. We investigate this problem for two restrictions. First, we assume that agents only perturb their preferences over a static set of available matching partners. When only one agent perturbs their preference order, we present a polynomial-time algorithm for solving RobustPopularMatching that is based on solving PopularEdge on suitably defined hybrid instances. By contrast, we encounter NP-completeness already for the case where only four agents perform a downward shift. Moreover, we identify a class of Yes-instances to RobustPopularMatching, where only unpopular agents perturb their preference orders.

In addition, we consider RobustPopularMatching for reduced availability. We encounter a complexity dichotomy based on preference completeness. If one input instance is complete, we can efficiently solve RobustPopularMatching by solving a maximum weight popular matching problem. However, if this is not the case, we once again obtain an NP-completeness.

We believe that our research paves the path for various exciting research directions, and we conclude by discussing some of these. First, an immediate open problem is to close the gap in the complexity of RobustPopularMatching between the feasibility for one agent and intractability for four agents changing their preference orders, i.e., determining the complexity of RobustPopularMatching if two or three agents perturb their preference orders. Another specific open problem concerns the complexity of RobustPopularMatching if only one side of the agents is allowed to change their preferences. As we have mentioned earlier, a polynomial-time algorithm exists for this problem when considering stable matchings [18]. However, our approach defining hybrid instances already has limitations if only two agents from the same side swap the preference order for a pair of other agents. Since popular matchings seem not to possess the lattice structure that was used for tackling stable matchings, we conjecture NP-hardness for popularity.

On a different note, it would be interesting to explore escape routes to our discovered hardness results. For this, one could try to efficiently find matchings offering a compromise between popularity in each of the input instances. For instance, one could attempt to find popular matchings in the second instance that have a large overlap with a given popular matching in the first instance. For complete instances, this can be done by finding a maximum weight popular matching problem similar to the approach for Proposition 4.11. In general, defining and investigating other notions of compromise matchings may lead to intriguing further discoveries.

ACKNOWLEDGMENTS

This work was supported in part by the AI Programme of The Alan Turing Institute and the NSF under grant CCF-2230414. We would like to thank Vijay Vazirani and Telikepalli Kavitha for valuable discussions and the anonymous reviewers for their comments.

REFERENCES

[1] D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. 2007. Popular matchings. *SIAM J. Comput.* 37, 4 (2007), 1030–1034.

[2] H. Aziz, P. Biró, T. Fleiner, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. 2022. Stable matching with uncertain pairwise preferences. *Theoretical Computer Science* 909 (2022), 1–11.

[3] H. Aziz, P. Biró, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. 2020. Stable matching with uncertain linear preferences. *Algorithmica* 82 (2020), 1410–1433.

[4] P. Biró, R. W. Irving, and D. F. Manlove. 2010. Popular Matchings in the Marriage and Roommates Problems. In *Proceedings of the 7th Italian Conference on Algorithms and Complexity (CIAC)*. 97–108.

[5] N. Boehmer, R. Bredereck, K. Heeger, and R. Niedermeier. 2021. Bribery and control in stable marriage. *Journal of Artificial Intelligence Research* 71 (2021), 993–1048.

[6] F. Brandt and M. Bullinger. 2022. Finding and Recognizing Popular Coalition Structures. *Journal of Artificial Intelligence Research* 74 (2022), 569–626.

[7] R. Bredereck, P. Faliszewski, A. Kaczmarczyk, R. Niedermeier, P. Skowron, and N. Talmon. 2021. Robustness among multiwinner voting rules. *Artificial Intelligence* 290 (2021), 103403.

[8] M. Bullinger, R. R. Gangam, and P. Shahkar. 2024. *Robust Popular Matchings*. Technical Report. https://arxiv.org/abs/2401.12653.

[9] Marquis de Condorcet. 1785. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale. Facsimile published in 1972 by Chelsea Publishing Company, New York.

[10] Á. Cseh. 2017. Popular Matchings. In *Trends in Computational Social Choice*, U. Endriss (Ed.). AI Access, Chapter 6.

[11] Á. Cseh and T. Kavitha. 2017. Popular edges and dominant matchings. *Mathematical Programming* 172, 1 (2017), 209–229.

[12] E. Elkind, P. Faliszewski, and A. Slinko. 2009. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT)*. 299–310.

[13] Y. Faenza, T. Kavitha, V. Powers, and X. Zhang. 2019. Popular Matchings and Limits to Tractability. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2790–2809.

[14] P. Faliszewski and J. Rothe. 2016. Control and Bribery in Voting. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, U. Endriss, J. Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Chapter 7.

[15] D. Gale and L. S. Shapley. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15.

[16] D. Gale and M. Sotomayor. 1985. Some remarks on the stable matching problem. *Discrete Applied Mathematics* 11, 3 (1985), 223–232.

[17] R. R. Gangam, T. Mai, N. Raju, and V. V. Vazirani. 2022. A Structural and Algorithmic Study of Stable Matching Lattices of "Nearby" Instances, with Applications. In *Proceedings of the 37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. 19:1–19:20.

[18] R. R. Gangam, T. Mai, N. Raju, and V. V. Vazirani. 2023. *The Stable Matching Lattice under Changed Preferences, and Associated Algorithms*. Technical Report. https://arxiv.org/abs/2304.02590.

[19] P. Gärdenfors. 1975. Match Making: Assignments based on bilateral preferences. *Behavioral Science* 20, 3 (1975), 166–173.

[20] S. Gupta, P. Misra, S. Saurabh, and M. Zehavi. 2019. Popular Matching In Roommates Setting is NP-hard. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2810–2822.

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

[22] T. Mai and V. V. Vazirani. 2018. Finding Stable Matchings that are Robust to Errors in the Input. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA)*. 60:1–60:11.

[23] D. F. Manlove. 2013. *Algorithmics of Matching Under Preferences*. World Scientific Publishing Company.

[24] S. Miyazaki and K. Okamoto. 2019. Jointly Stable Matchings. *Journal of Combinatorial Optimization* 38, 2 (2019), 646–665.

[25] A. E. Roth. 1984. The evolution of the labor market for medical interns and resident: a case study in game theory. *The Journal of Political Economy* 92 (1984), 991–1016.

[26] D. Shiryaev, L. Yu, and E. Elkind. 2013. On elections with robust winners. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 415–422.