# Matchings under Preferences: Strength of Stability and Tradeoffs

JIEHUA CHEN, TU Wien, Austria and University of Warsaw, Poland
PIOTR SKOWRON, University of Warsaw, Poland
MANUEL SORGE, TU Wien, Austria and University of Warsaw

We propose two solution concepts for matchings under preferences: *robustness* and *near stability*. The former strengthens while the latter relaxes the classical definition of stability by Gale and Shapley (1962). Informally speaking, robustness requires that a matching must be stable in the classical sense, even if the agents slightly change their preferences. Near stability, however, imposes that a matching must become stable (again, in the classical sense) provided the agents are willing to adjust their preferences a bit. Both of our concepts are quantitative; together they provide means for a fine-grained analysis of the stability of matchings. Moreover, our concepts allow the exploration of tradeoffs between stability and other criteria of social optimality, such as the egalitarian cost and the number of unmatched agents. We investigate the computational complexity of finding matchings that implement certain predefined tradeoffs. We provide a polynomial-time algorithm that, given agent preferences, returns a socially optimal robust matching (if it exists), and we prove that finding a socially optimal and nearly stable matching is computationally hard.

CCS Concepts: • **Theory of computation → Solution concepts in game theory**; **Algorithmic mechanism design**; **Fixed parameter tractability**; **W hierarchy**; **Problems, reductions and completeness;**

Additional Key Words and Phrases: Stable matchings, concepts of stability, NP-hardness, parameterized complexity analysis, exact algorithms, approximation algorithms

**ACM Reference format:**
Jiehua Chen, Piotr Skowron, and Manuel Sorge. 2021. Matchings under Preferences: Strength of Stability and Tradeoffs. *ACM Trans. Econ. Comput.* 9, 4, Article 20 (October 2021), 55 pages.
https://doi.org/10.1145/3485000

Piotr Skowron was supported by the Foundation for Polish Science within the Homing programme (Project title: "Normative Comparison of Multiwinner Election Rules") and by Poland's National Science Center grant UMO-2019/35/B/ST6/02215. Jiehua Chen and Manuel Sorge were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement numbers 677651 (JC) and 714704 (MS). Jiehua Chen was also supported by the WWTF research project (VRG18-012). Manuel Sorge was also supported by Alexander von Humboldt Foundation. Main work of Jiehua Chen and Manuel Sorge done while with University of Warsaw.


Authors' addresses: J. Chen, TU Wien, Wien, Austria, University of Warsaw, Warsaw, Poland; email: jiehua.chen2@gmail.com; P. Skowron, University of Warsaw, Warsaw, Poland; email: p.skowron@mimuw.edu.pl; M. Sorge, University of Warsaw, Warsaw, Poland; TU Wien, Wien, Austria; email: manuel.sorge@gmail.com.


2167-8375/2021/10-ART20 $15.00
https://doi.org/10.1145/3485000

## 1 INTRODUCTION

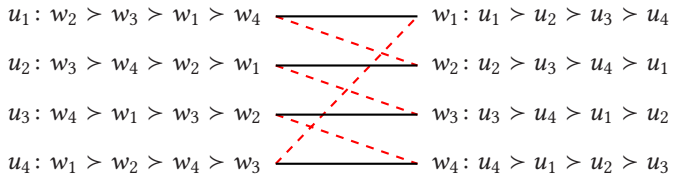
In the Stable Marriage problem [22], we are given two disjoint sets of agents, $U$ and $W$. Each agent from one set has a strict preference list that ranks a subset of the agents from the other set. The sets of agents and their preference lists are collectively called *preference profile* (sometimes abbreviated to just *profile*). The goal is to find a matching between $U$ and $W$ that does not contain a *blocking pair*, i.e., a pair of agents who are not matched with each other but prefer being matched with each other rather than being unmatched or being with their matched partners. A matching with no blocking pairs is called a *stable matching*.

The classical definition of stability is qualitative: A matching can be either stable or not, and there are no other states in between or beyond. In this article, by contrast, we take a quantitative approach. We propose and study two solution concepts: *robustness* and *near stability*, where the former strengthens and the latter relaxes the notion of stability. Intuitively, a robust matching is more than stable; it remains stable even if agents change their preferences slightly. In contrast, a nearly stable matching needs not be stable for the original profile, but it becomes so after some minor changes in the preferences. Below, we give more precise definitions of robust and nearly stable matchings and motivate their study through a number of observations.

*Robust matchings.* Our first main observation is that the preference lists provided by the agents do not always reflect their true preferences. This can happen, for instance, because the agents do not have full information about their potential partners, or because formulating accurate preferences is a hard task that requires substantial cognitive effort [40]. It is also typical that the agents change their preferences over time, for instance, in response to changes in their operating environment. Thus, a matching that is stable in the classical sense (with respect to the preferences expressed by the agents at the beginning) can in fact contain two or more agents who already have or will likely have incentives to drop their assigned partners and be matched with each other. In other words, there are situations where the classical definition of stability can turn out to be too weak. In a different setting, a third party may want to destabilize a matching by bribing certain agents to change their preferences. In that case, we are interested in stable matchings that defy such attacks.

For the above reasons, we introduce and study *$d$-robustness*, a strengthened notion of stability. A matching is *$d$-robust* for a given preference profile if this matching is stable and remains stable after performing an arbitrary sequence of up to $d$ swaps. Here, a *swap* is the reversal of two consecutive agents in a preference list. Intuitively, if a matching is $d$-robust for some reasonably large $d$, then it will not become unstable even if the agents specified slightly inaccurate preferences, nor will it become unstable even if the agents change their preferences by a little. Example 1.1 below illustrates the concept of robustness.

*Example 1.1.* Consider the profile $P$ for two disjoint sets of agents $U = \{u_1, u_2, u_3, u_4\}$ and $W = \{w_1, w_2, w_3, w_4\}$, where the preference lists are to the right of the corresponding agents; preferences are represented as horizontal lists where more preferred agents are put to the left of the less preferred ones.



This profile admits five stable matchings:

(1) The $U$-optimal stable matching $M_1 = \{\{u_1, w_2\}, \{u_2, w_3\}, \{u_3, w_4\}, \{u_4, w_1\}\}$ (indicated by the red dashed lines),
(2) the $W$-optimal stable matching $M_2 = \{\{u_1, w_1\}, \{u_2, w_2\}, \{u_3, w_3\}, \{u_4, w_4\}\}$ (indicated by the black solid lines),
(3) $M_3 = \{\{u_1, w_3\}, \{u_2, w_4\}, \{u_3, w_1\}, \{u_4, w_2\}\}$,
(4) $M_4 = \{\{u_1, w_1\}, \{u_2, w_4\}, \{u_3, w_3\}, \{u_4, w_2\}\}$, and
(5) $M_5 = \{\{u_1, w_3\}, \{u_2, w_2\}, \{u_3, w_1\}, \{u_4, w_4\}\}$.

See Section 3.1 for the notion of $X$-optimal stable matching with $X \in \{U, W\}$.
In terms of robustness, $M_2$ is superior to $M_1$, since $M_2$ is 1-robust but $M_1$ is not. To see this, we observe that, to make $M_2$ unstable, we need to perform at least one swap in the preference list of some agent $w$ in $W$ that involves its partner $M(w)$ and the agent ranked in the second position. However, the agent that is ranked in the second position does not prefer $w$ to its partner. In other words, a single swap does not suffice to make $M_2$ unstable. Stable matching $M_1$ is not 1-robust, since one can swap in the preference list of any agent $u$ from $U$ the two agents $M_1(u)$ and $w$ in the first and the second positions to obtain a profile where $\{u, w\}$ is a blocking pair for $M_1$. One can verify that no stable matching other than $M_2$ is 1-robust. ◇

*Nearly stable matchings.* Our second main observation is that there exist other factors, apart from the preferences, that can discourage the agents to break their relations with their matched partners. Such factors may include social pressure and additional costs incurred by changing the partner, for example. Thus, in some situations even weaker forms of stability may guarantee a sufficient level of resilience to agents changing their minds. We express this as the *local $d$-near stability* of a matching, which stipulates that there is a sequence of swaps such that the matching becomes stable, and in each agent's preference list, at most $d$ swaps are made.

This concept has an intuitive interpretation similar to the $\epsilon$-Nash-equilibrium [43, Section 2.6.6] in game theory: In a locally $d$-nearly stable matching no agent can improve its satisfaction by more than $d$ through rematching (see also the equivalent definitions in Section 2.2.3). This is analogous to $\epsilon$-Nash-equilibria, where no agent can improve their outcome by more than $\epsilon$. In this sense, local near stability also measures the strength of the incentive for two agents in a blocking pair to change their partners.[1]

Our third main observation is that, when there are constraints on other factors of the matching like social welfare (see below), it may not be possible to find a stable matching satisfying these constraints. Thus, it may be necessary to balance between the social welfare and the costs incurred by agents that want to switch partners. This cost is captured by the *global $d$-near stability* of a matching $M$, stating that there is a sequence of at most $d$ swaps in total such that $M$ becomes stable. To achieve the desired social welfare, we may thus provide proportionate compensation to the agents affected by the swaps.

Taking nearly stable matchings into consideration may indeed allow us to find a matching that is significantly better from the perspective of the society as a whole, than if we restricted ourselves to stable matchings only. We illustrate such scenario through the following example:

*Example 1.2.* Let $U = \{a_0, \ldots, a_{n-1}, x_1, \ldots, x_n\}$ and $W = \{b_0, \ldots, b_{n-1}, y_1, \ldots, y_n\}$, and consider the following preference profile $P$ of the agents; the index "$i + 1$" is taken modulo $n$.

[1] There are some differences between the two concepts, since we deal with ordinal preferences. Yet, our concepts generalize to cardinal utilities, where the similarities are more transparent.

$$a_0 : b_0 \succ b_1, \qquad b_0 : a_0 \succ a_{n-1},$$
$$a_i : b_i \succ b_{i+1}, \qquad b_i : a_{i-1} \succ x_1 \succ \cdots \succ x_n \succ a_i, \qquad \text{for all } i \in \{1, \ldots, n-1\},$$
$$x_i : y_i \succ b_1 \succ \cdots \succ b_{n-1}, \qquad y_i : x_i, \qquad \text{for all } i \in \{1, \ldots, n\}.$$

In every stable matching $M^s$ of $P$ agent $x_i$ must be matched with $y_i$ for all $i \in \{1, \ldots, n\}$, and $a_0$ with $b_0$. Consequently, $a_1$ needs to be matched with $b_1$ and, by an inductive argument, we can infer that, for each $i \in \{1, \ldots, n-1\}$, $a_i$ must be matched with $b_i$. Thus, if we look at the agents from $B = \{b_1, \ldots, b_n\}$, we observe that, except for $b_0$, each of them is matched with a partner ranked at the $(n+2)^{\text{th}}$ position. Yet, if we consider the profile obtained from $P$ by swapping $a_0$ and $a_{n-1}$ in the preference list of $b_0$, then $M = \{\{a_0, b_1\}, \{a_1, b_2\}, \ldots, \{a_{n-1}, b_0\}\} \cup \{\{x_i, y_i\} \mid 1 \leq i \leq n\}$ would be a stable matching. In this matching everyone is matched to one of her two most preferred agents. Clearly, when the satisfaction of an agent corresponds to how much she prefers her partner, this matching has a higher satisfaction among the agents than any stable matching. ◇

Intuitively, Example 1.2 shows that with a relatively small loss of stability, one can significantly improve the social cost of a matching $M$—in this example, this cost is defined as the sum of ranks that an agent $x$ has in the preference list of its matched partner $M(x)$. In the literature this measure is often referred to as the egalitarian cost of a matching [30]. We also consider another metric that counts the number of agents that are assigned a partner in a matching. We assume that the agents *do not* rank those from the opposite set that they would not agree to be matched to. In such a case a stable matching does not need to be *perfect*, i.e., it is possible that some agents will not be matched by any stable matching. The effect of stability on the number of matched agents is illustrated in Example 1.3.

*Example 1.3.* Consider a profile with 2 men, $a_1$ and $a_2$, and 2 women, $b_1$ and $b_2$, with preference lists:

$$a_1 : b_1, \qquad b_1 : a_2 \succ a_1,$$
$$a_2 : b_1 \succ b_2, \qquad b_2 : a_2.$$

For this profile, the only stable matching is $\{\{a_2, b_1\}\}$. However, if we swapped $b_1$ and $b_2$ in the preference list of $a_2$, then $\{\{a_1, b_1\}, \{a_2, b_2\}\}$ would be a stable matching, i.e., we would obtain a stable matching where more agents have partners. ◇

Let us explain the intuitive idea behind near stability based on Example 1.3. Consider matching $M = \{\{a_1, b_1\}, \{a_2, b_2\}\}$. We see that $b_1$ and $b_2$ are close to each other in the preferences list of $a_2$. Thus, intuitively, the incentive for agent $a_2$ to break its contract with $b_2$ and to form a new contract with $b_1$ is weak ($a_2$ would obtain a partner that it prefers only by one position according to its preference list). Thus, if breaking a contract comes with some cost, then $a_2$ might not decide to do this. Hence, one could consider matching $M$ "close to being stable." Formally, this is equivalent to saying that the matching would be stable if the preference list of $a_2$ changed by a single swap of adjacent agents. Further, imposing near stability, a weaker form of classic stability, allows to obtain matchings that are better with respect to other criteria, in this case a matching with more agents matched.

Example 1.2 and Example 1.3 suggest that there is a (possibly non-linear) tradeoff between stability and other criteria of social optimality. Our definition of near stability provides a formalism necessary to describe the tradeoffs; yet, to take advantage of them, one needs to be able to identify situations where a large improvement of social welfare is possible with a relatively small sacrifice of stability. We formalize this question as a computational problem (see Section 2.1 for formal definitions) and study its complexity.

Table 1. Summary of Our Results

| Social criteria | Robust (without ties) | Robust (with ties) | Globally Nearly Stable (without ties) | Locally Nearly Stable (without ties) |
|---|---|---|---|---|
| No further restrictions | P [Thm 3.22] | In NP, NP-h ($d = 1$) [Prop 4.4, Thm 4.5] | Always exists even for $d_G = d_L = 0$ and can be found in $O(n^2)$ time [22, 26] | |
| Perfect matching | P [Cor 3.24] | In NP, NP-h ($d = 0$) [37] | In XP, W[1]-h for $d_G$ [Prop 5.4, Thm 5.5]<br>W[1]-h for $n_u$ [Cor 5.6]<br>In FPT for $n_m$ [Prop 5.7]<br>*No* poly-approximation [Thm 5.3] | NP-h ($d_L = 1$) [Thm 5.3]<br>W[1]-h for $n_u$ [Cor 5.6]<br>*No* poly-approximation [Thm 5.3] |
| Egalitarian cost $\eta$ | P [Thm 3.26] | In NP, NP-h ($d = 0$) [37] | In XP, W[1]-h for $d_G$ [Prop 5.4, Thm 5.5]<br>W[1]-h for $n_u$ [Cor 5.6]<br>*No* poly-approximation [Thm 5.3] | NP-h ($d_L = 1$) [Thm 5.3]<br>W[1]-h for $n_u$ [Cor 5.6]<br>*No* poly-approximation [Thm 5.3] |

Herein, $d$ denotes the number of swaps for robust matchings, $d_L$ (respectively, $d_G$) denotes the number of swaps for local near stability (respectively, global near stability), $\eta$ denotes the egalitarian cost of the desired matching, and $n_m$ (respectively, $n_u$) denotes the number of matched (respectively, unmatched) agents in any stable matching of the initial profile without ties.

### 1.1 Our Contributions

We introduce the concepts of robustness and near stability, and explore the tradeoff between stability and the egalitarian cost, and between stability and the number of matched agents. We provide a polynomial-time algorithm that, given a preference profile and a number $d$, decides whether a $d$-robust matching exists and computes one if it exists (Theorem 3.22). We achieve this by providing a polynomial-size characterization of the preference profiles (Section 3.2) that are close to the input preference profile and by heavily exploiting the structural properties of so-called rotations adherent to a preference profile [26]. Moreover, our algorithm can be extended to find a $d$-robust matching with minimum egalitarian cost if there exists any $d$-robust matching (Theorem 3.26). However, when ties are present, we show that deciding the existence of a $d$-robust matching is NP-complete and remains NP-hard even if $d = 1$ (Theorem 4.5).

In contrast to the polynomial-time algorithms for robust matchings, we show that the problem of finding a matching that implements a certain tradeoff between the near stability and the egalitarian cost, or between the near stability and the property of being perfect for the matching is NP-complete, and it is NP-hard to approximate (Theorem 5.3). Motivated by this general hardness result, we study the parameterized complexity, mainly with respect to the parameter number of allowed swaps. For details on parameterized complexity, we refer to the books of Cygan et al. [15], Downey and Fellows [16], Flum and Grohe [21], and Niedermeier [42]. For this article, it suffices to know that

— a computational problem in *XP* for a parameter $k$ if it can be solved in $O(n^{f(k)})$ time, where $n$ denotes the input size and $f$ is a computable function depending only on $k$,
— a computational problem in *FPT* for a parameter $k$ if it can be solved in $f(k) \cdot n^{O(1)}$ time, and
— a computational problem is *W[1]-hard* for a parameter $k$ means that unless the unlikely complexity-theoretic collapse "FPT = W[1]" happens, the corresponding problem *cannot* be solved in $f(k) \cdot n^{O(1)}$ time.

See Table 1 for a summary of our results. We mostly obtain further hardness results. While for local near stability the problem remains NP-complete even if only a single swap per agent is allowed (Theorem 5.3), for global near stability we obtain a polynomial-time algorithm if the total number $d_G$ of allowed swaps is a constant (Proposition 5.4). The exponent in the running time depends on $d_G$, however, and this dependency cannot be removed unless the unlikely complexity-theoretic collapse "FPT = W[1]" happens (Theorem 5.5). We also study the complexity in the cases where there are small numbers of unmatched or matched agents in a classically stable matching of the input profile.

### 1.2 Related Work

For an overview on the Stable Marriage and related problems, we refer to the books of Knuth [34], Gusfield and Irving [26], and Manlove [38].

*Robustness.* First, we review work related to our concept of robustness. As we mentioned in the beginning of this section, one of the observations that motivates our study of robust matchings is that the preferences of the agents may be uncertain. In this regard, Aziz et al. [3], Miyazaki and Okamoto [41], and Chen et al. [14] study a variant of Stable Marriage where there is a collection of "possible" preference profiles given as input, and they look for a matching that is stable in each of the given profiles (the corresponding computational problem is NP-hard even for constant number of input profiles). Our work starts with the assumption that the preferences provided by the agents are a good approximation of their true preferences. Thus, our robustness concept respects every profile that is close to the preferences provided by the agents. This makes a crucial difference—finding a robust matching if one exists, according to our definition, is solvable in polynomial time.

Our robustness concept is related to the works of Mai and Vazirani [35]. They introduced a probabilistic model, where there are additional *erroneous* preference profiles given in the input, each differing from the original one by a *single* preference list in which a single agent's position is shifted forward. The given erroneous profiles are equipped with a probability distribution and the goal is to find a stable matching for the input profile that remains stable with largest probability in a randomly chosen erroneous profile. While they do not assume the difference of an erroneous profile to the input profile to be small, their assumption on the structure of erroneous profiles implies that there are only $O(n^3)$ profiles in the probability distribution. In contrast, in our definition of robustness, we require that the sought matching must be stable in *every* profile that is close to the original one, but that can differ from the original profile by more than one preference list. Furthermore, we do not assume that the distribution of the profiles is given, but rather infer the "relevant" (close) profiles directly from the original profile. Our approach, based on the concept of distances, induces a quantitative measure of the strength of stability; we further extend it in the converse direction by considering matchings that are nearly stable, getting a full set of tools that allows one to reason about the strength of stability for any matching.

Mai and Vazirani proved that their probabilistic problem can be solved in polynomial time. Their techniques rely on the fact that, for each erroneous profile $P'$, one can in polynomial time find two relevant rotations $\pi$ and $\rho$ of the input profile $P$ such that each closed subset $S$ of rotations (corresponding to a stable matching of the input profile) that includes $\pi$ must also include $\rho$ as long as $S$ corresponds to a stable matching of $P$ and $P'$. In our model, this is not the case; for each preference profile that differs by $d$ swaps there may be more than two relevant rotations that are interdependent. Thus, this approach is not directly applicable in our scenario. Nevertheless, we can characterize all close preference profiles by polynomially many close and relevant profiles. Each of the relevant profiles is represented by at most two inter-related rotations (e.g., the inclusion of one rotation leads to the inclusion another rotation). Based on this, and partly inspired by the techniques of Mai and Vazirani [35], we can extend the rotation graph for the original profile and provide a polynomial-time algorithm for our robustness model. In this regard, our algorithmic techniques can be considered as a generalization of the ones by Mai and Vazirani [35]. Moreover, our structural characterization (see Section 3.2) can be adapted to also solve the fully robustness problem in their follow-up paper [36] where there are additionally polynomially many erroneous profiles, each differing from the original one by a single preference list, and the goal is to find a stable matching that stays stable in these profiles. Note that, in the general case, where we allow arbitrary changes between profiles, it is impossible to obtain a polynomial-time algorithm unless

P = NP: Miyazaki and Okamoto [41] and Chen et al. [14] showed that finding a matching that is stable for even only two preference profiles is NP-hard. The reason for the jump in complexity can be understood as follows: For two arbitrary preference profiles there may be more than two rotations such that their relations in the rotation digraph are not immediately clear.

Kanade et al. [31] considered a probabilistic model similar to the one by Mai and Vazirani [35], where at each timestep, a randomly selected agent may change her preference list by swapping two randomly selected adjacent agents. The authors focused on algorithms that in expectation generate as few blocking pairs as possible. Our robustness approach differs in two aspects: (1) we consider a fully deterministic setting, and (2) we measure the robustness of a matching by computing the distance to the closest preferences profile where the matching becomes unstable, whereas Kanade et al. [31] measure it by counting the expected number of blocking pairs that would appear if the agents' preferences changed randomly.

Finally, let us mention a relation between robustness and strategy-proofness. We say that a matching algorithm is *strategy-proof* (see References [26, Chapter 4], [46, Section 1.7], [38, Section 2.9]) if no agent can obtain a better partner by misreporting her preferences; it is known that there exists no strategy-proof matching mechanism. Nevertheless, robustness implies a very weak form of strategy-proofness, where the set of agents' strategies is limited—the agents are willing to report only those rankings that are not significantly different from their true preferences. Even more closely related, robustness implies resilience to certain forms of bribery—the problem of bribery, originally defined for single-winner elections [20], can be naturally adapted to matchings.

*Near stability.* Now, we turn to work that is more related to our near stability concept. Another interpretation of a locally $d$-nearly stable matching is that in each blocking pair there is an agent whose rank improvement by switching partners would be at most $d$. Drummond and Boutilier [17] use this rank improvement approach to study Stable Marriage problem under *partially ordered* preferences. They introduced the notion of an *$r$-maximally stable* matching, i.e., a matching such that for each linear completion of the input profile and for each unmatched pair at least one agent in the pair ranks the other higher than its matched partner by at most $r$ positions. When restricting the input to linear preferences, as is our focus here, $r$-maximal stability is equivalent to local r-near stability for each $r \geq 0$. We prove this formally in Section 2.2.3. Here, in contrast, we do not deal with partial preferences, but instead we want to achieve a given social welfare in addition to $r$-maximal stability.

Pini et al. [44] and Anshelevich et al. [2] studied a concept called (additive) $\alpha$-stability that measures the degree of instability for utility-based preferences. For ordinal preferences their concept is equivalent to our local $\alpha$-near stability. Anshelevich et al. [2] studied the tradeoff between the total utility of a matching and its $\alpha$-stability for restricted structures of utility scores (which cannot model ordinal preferences). Pini et al. [44] showed that an optimal $\alpha$-stable matching, subject to a lexicographic aggregation of certain optimality criteria, can be done in polynomial time and they considered manipulation issues.

*Other related notions.* Finally, we review further related work, not necessarily directly related to our notions of robustness or near stability. Recently, Menon and Larson [39] proposed a different robustness concept to deal with uncertain preferences—the authors assume that each agent has preferences with ties on the agents of the opposite set and look for a perfect matching to minimize the maximum number of blocking pairs among all linear extensions of the input preferences. In contrast to our approach, however, these blocking pairs may represent an arbitrarily large rank improvement, i.e., an arbitrarily large number of swaps needed to make the matching stable. Finding a solution as above is equivalent to finding a *perfect* matching with minimum number of so-called *super-blocking pairs*, a concept introduced by Irving [28] to cope with preferences with ties, i.e.,

weak orders (also see Reference [26]). Menon and Larson [39] mainly obtained inapproximability results.

Genc et al. [24, 25] provide yet another view on robustness in the context of stable matchings. They define an $(x, y)$-supermatch as a stable matching that satisfies the following property: If any $x$ pairs break up, then it is possible to rematch the agents in these pairs so that the new matching is again stable; further, this re-matching must be done by breaking at most $y$ other pairs. Hence, an $(x, y)$-supermatch may not be robust in our sense, but it needs to be easy to repair. Very recently, Bredereck et al. [7] study a dynamic variant of the stable matching problem, where the agents are already matched but some of the agents change their preferences and these changes are given, and the goal is to rematch as few agents as possible to obtain a stable matching. They study cases where the agent preferences may be incomplete and contain ties and obtain mostly hardness results.

In the second part of this article, we study tradeoffs between the stability (of various strength) and optimality criteria such as the egalitarian cost and the number of unmatched agents. This is related to the studies on the price of stability in matching markets [5]. Other studied optimality criteria include the Pareto efficiency concept. A matching is Pareto-efficient for one side, say $U$, if no other matching can make an agent from $U$ better off without making some other agent from $U$ worse off. Since Pareto efficiency and stability are incompatible [19], meaning that there may be instances where no stable matching is Pareto-efficient, even for only one side, Ergin [19], Cantala and Pàpai [10], Alcalde and Romero-Medina [1], Che and Tercieux [12], Kesten [33], Ehlers and Morill [18], and Troyan et al. [48] studied the tradeoff between stability and Pareto efficiency (for one side of the market). More precisely, Cantala and Pàpai [10], Alcalde and Romero-Medina [1], and Che and Tercieux [12] consider other near stability notions that are inspired by the idea of bargaining—according to these notions for each claim of an alternative matching one should be able to formulate a respective counter-claim of another alternative matching. Under this weaker form of stability, there exists matchings that are both nearly stable (in their sense) and Pareto-efficient, for only one side of the market. Kesten [33] considers a richer model, where agents can provide additional information that would not affect their assignment, but that would allow to improve the quality of the matching overall. Finally, Troyan et al. [48] study a probabilistic framework, where agents' preferences are drawn randomly.

Concepts similar to our robustness have been also studied in other contexts, for instance for single-winner [47] and multi-winner elections [8].

## 2 BASIC DEFINITIONS, NOTATIONS, AND OUR STABILITY CONCEPTS

For each natural number $t$ by $[t]$ we denote the set $\{1, 2, \ldots, t\}$. Throughout, we will use $U$ and $W$ to denote two disjoint $n$-element sets of agents.

A *preference profile* $P = ((\succ_u^P)_{u \in U}, (\succ_w^P)_{w \in W})$ is a collection of the *preference lists* of the agents from $U$ and $W$. Here, for each agent $u \in U$, the notation $\succ_u^P$ denotes a linear order on a subset $W'$ of $W$ that represents the ranking of agent $u$ over all agents from $W'$ in profile $P$. The agents in $W'$ are also called *acceptable* to $u$. The agents *not ranked* by $u$ are those in $W \setminus W'$, that is, those that $u$ does not agree to be matched to; we also call them *unacceptable*. If $w \succ_u^P w'$, then we say that $w$ is *preferred* to $w'$ by $u$ in $P$. Analogously, for each agent $w \in W$, $\succ_w$ represents a linear order on (a subset of) $U$ that represents the ranking of $w$ in profile $P$ and we likewise use the notions of preference list, preferred, (un-)acceptable, and (not) ranked. We assume that the acceptability relation among the agents is *symmetric*, i.e., for each two agents $u$ and $w$ it holds that $u$ is acceptable to $w$ if and only if $w$ is acceptable to $u$. We say that $P$ has *complete* preferences if each agent finds all agents from the opposite set acceptable.

Given an agent $x$ with her preference list $\succ_x$ and given an agent $y$ from the opposite set, the *rank* $\mathrm{rk}_x(y, \succ_x)$ of $y$ in the preference list of $x$ is equal to the number of agents that are preferred

to $y$ by $x$. If $y$ is unacceptable to $x$, then the rank $\mathsf{rk}_x(y, \succ_x)$ is defined as the number of agents acceptable to $x$. We usually omit the symbol $\succ_x$ in $\mathsf{rk}_x(y, \succ_x)$ and write only $\mathsf{rk}_x(y)$ whenever the preference list of $x$ is clear from the context. For instance, the rank of $y_3$ in the preference list $\succ_x : y_1 \succ y_3 \succ y_2$ is one. We say that *$x$ ranks $y$ higher than $z$* if $\mathsf{rk}_x(y) < \mathsf{rk}_x(z)$.

Throughout, except in Section 4, by "$x \succeq y$" for two agents $x$ and $y$, we mean "$x = y$ or $x \succ y$."

To a preference profile $P = ((\succ_u^P)_{u \in U}, (\succ_w^P)_{w \in W})$, we assign an *acceptability graph* $G$, which is a bipartite graph on $U \uplus W$ such that two agents are connected by an edge if each finds the other acceptable. Without loss of generality, $G$ does not contain isolated vertices, meaning that each agent has at least one agent that it finds acceptable.

*Blocking pairs and stable matchings.* Given two disjoint sets of agents, $U$ and $W$, a *matching* $M$ is a set of pairwise disjoint pairs, each pair containing one agent from $U$ and one agent from $W$, i.e., $M \subseteq \{\{u, w\} \mid u \in U \wedge w \in W\}$ and for each two distinct pairs $p, p' \in M$ it holds that $p \cap p' = \emptyset$. Given a pair $\{u, w\}$ with $u \in U$ and $w \in W$, if it holds that $\{u, w\} \in M$, then we use $M(u)$ to refer to $w$ and $M(w)$ to refer to $u$, and we say that $u$ and $w$ are their respective *partners* under $M$; otherwise, we say that $\{u, w\}$ is an *unmatched pair* under $M$. If an agent $x$ is *not* assigned any partner by $M$, then we say that $x$ is *unmatched by $M$* and we put $M(x) = \emptyset$. We also call $\emptyset$ an *bottom agent*. If $M(x) = \emptyset$, then we define the *rank* $\mathsf{rk}_x(M(x))$ as the number of agents acceptable to $x$ to model that $x$ prefers to be matched to any acceptable agent over being unmatched.

We say that a pair $\{u, w\}$ is *blocking* (or *a blocking pair of*) $M$ if the following holds:

(1) $u$ and $w$ find each other acceptable but are not matched together,
(2) $u$ is either unmatched by $M$ or $\mathsf{rk}_u(w) < \mathsf{rk}_u(M(u))$, and
(3) $w$ is either unmatched by $M$ or $\mathsf{rk}_w(u) < \mathsf{rk}_w(M(w))$.

Finally, we say that a matching $M$ is *stable* if it does not admit a blocking pair. Example 1.1 in the introduction illustrates the concept of stable matchings.

We use $\mathcal{SM}(P)$ to denote the set of all stable matchings for a preference profile $P$. Given a matching $M$, we use $\mathcal{BP}(P, M)$ to denote the set of all unmatched pairs that are blocking $M$ in profile $P$. Obviously, for each stable matching $M \in \mathcal{SM}(P)$, it holds that $\mathcal{BP}(P, M) = \emptyset$.

### 2.1 Our Spectrum of Stability Notions and Central Problems

Let us now define our concepts of robustness and near stability, informally introduced in Section 1.

First, we need the notion of *swaps* and *swap distances*.

*Definition 2.1 (Swaps and Swap Distances).* Given an agent $z$ and two agents $x$ and $y$ that are ordered consecutive in the preference list $\succ_z$ of $z$, the *swap* of $x$ and $y$ by $z$ denotes the operation of switching the relative order of $x$ and $y$ in $\succ_z$. We use $(z, \{x, y\})$ to denote such a swap.

For two preference lists $\succ$ and $\succ'$, the *swap distance between $\succ$ and $\succ'$* is defined as follows:

$$\tau(\succ, \succ') := \begin{cases} |\{\{x, y\} \mid x \succ y \wedge y \succ' x\}|, & \text{if } \succeq_i \text{ and } \succeq_j \text{ have the } \textit{same} \text{ acceptable set,} \\ \infty, & \text{otherwise.} \end{cases}$$

For two preference profiles $P_1$ and $P_2$ with $P_1 = ((\succ_u^{P_1})_{u \in U}, (\succ_w^{P_1})_{w \in W})$ and $P_2 = ((\succ_u^{P_2})_{u \in U}, (\succ_w^{P_2})_{w \in W})$, the *swap distance between $P_1$ and $P_2$* is defined as follows:

$$\tau(P_1, P_2) := \sum_{x \in U \cup W} \tau\left(\succ_x^{P_1}, \succ_x^{P_2}\right).$$

Intuitively, the swap distance $\tau(\succ, \succ')$ equals the minimum number of swaps that are required to turn $\succ$ into $\succ'$.

*Definition 2.2 (Robustness).* For a given preference profile $P$, we say that a matching $M$ is *$d$-robust* if for each profile $P'$ with $\tau(P, P') \leq d$ it holds that $M$ is stable in $P'$.

Note that our robustness concept is monotone—each $d$-robust matching is also $d'$-robust for $0 \leq d' \leq d$. We are interested in the computational question of finding the largest integer $d$ such that there is a $d$-robust matching. This can be phrased as a decision problem as follows:

Robust Matching

***Input:*** A preference profile $P$ with agent sets $U$ and $W$ of size $n$ each, and a non-negative integer $d \in \mathbb{N}$.

***Question:*** Is there a $d$-robust matching for $P$?

Next, we define near stability. Here, we introduce two definitions—global near stability and local near stability—that differ in the scope of admissible changes to the original preference profile.

*Definition 2.3 (Near stability).* For a given preference profile $P$, we say that a matching $M$ is *globally $d$-nearly stable* if there exists a profile $P'$ with $\tau(P, P') \leq d$ such that $M$ is stable in $P'$. We say that $M$ is *locally $d$-nearly stable* if there exists a profile $P'$ with $\tau(\succ_x^P, \succ_x^{P'}) \leq d$ for each agent $x \in U \cup W$ such that $M$ is stable in $P'$.

Since near stability is a more permissive concept than stability as defined by Gale and Shapley [22], it is straightforward to verify that a globally $d$-nearly stable (or locally $d$-nearly stable) matching always exists for $d \geq 0$. Here, our main focus is to explore the tradeoffs between the strength of stability and other criteria of social optimality. We say that a matching $M$ is *perfect* if each agent has a partner under $M$. The *egalitarian cost* of $M$ in a profile $P = (\succ_x)_{x \in U \cup W}$ is

$$\eta(M) \coloneqq \sum_{x \in U \cup W} \mathrm{rk}_x(M(x), \succ_x).$$

*Example 2.4.* The profile given in Example 1.2 has a unique stable matching $M^s$, where each agent $b_i$, $1 \leq i \leq n-1$, obtains partner $a_i$ with rank $n$. One can verify that the egalitarian cost $\eta(M^s)$ of $M^s$ is $\Omega(n^2)$. The second matching $M$ (see the example) has cost $O(n)$, which is stable for the profile, which differs from the original one by only one swap. ◇

This leads to the following computational problems, abbreviated as Global-Near+Perf, Local-Near+Perf, Global-Near+Egal, and Local-Near+Egal.

Globally (or Locally) Nearly Stable Perfect Matching

***Input:*** A preference profile $P$ with agent sets $U$ and $W$ of size $n$ each, and a non-negative integer $d \in \mathbb{N}$.

***Question:*** Is there a globally $d$-nearly stable (or locally $d$-nearly stable) stable matching for $P$ that is perfect?

Globally (or Locally) Nearly Stable Egalitarian Matching

***Input:*** A preference profile $P$ with agent sets $U$ and $W$ of size $n$ each, and two non-negative integers $d, \eta \in \mathbb{N}$.

***Question:*** Is there a globally $d$-nearly stable (or locally $d$-nearly stable) stable matching for $P$ that has egalitarian cost at most $\eta$ in $P$?

For preferences without ties (i.e., every agent has a strict preference list), we use the following fundamental result from the literature:

Theorem 2.5 (Follows from [26, Theorem 1.4.2]). *For incomplete preferences without ties, the agent set can be partitioned into two disjoint subsets $R$ and $S$ such that every stable matching matches every agent from $R$ and none of the agents from $S$. For agent set of size $2n$, this partition can be computed in $O(n^2)$ time.*

## 2.2 Structural Properties of Robust and Nearly Stable Matchings

Before we proceed further, we provide some structural results concerning Definition 2.2 and 2.3. First, we give two observations regarding robustness. These are not necessary for the considerations about algorithms later on, but serve to strengthen the intuition about profiles that allow for robust matchings and, we feel, are interesting in their own right. Further below, we consider the tradeoff between near stability and perfectness of matchings and give alternative characterizations of locally nearly stable matchings.

### 2.2.1 *Structure of Matchings and Profiles.*

Proposition 2.6. *If $d \geq n$ and there exists one agent who finds at least two other agents acceptable, then no stable matching is $d$-robust.*

Proof. Let $M$ be an arbitrary stable matching. To show that no stable matching is $d$-robust, it suffices to show that performing at most $n$ swaps can make an unmatched pair a blocking pair of $M$. To this end, let $x$ be an agent who finds at least two other agents acceptable. Further, let $y$ be an agent that is acceptable to $x$ so that $\{x, y\}$ satisfies the following: If $x$ is unmatched under $M$ or if $\mathsf{rk}_x(M(x)) \geq 1$, then $y$ is the most preferred agent of $x$ (that is, $\mathsf{rk}_x(y) = 0$); otherwise, $y$ is the second-most preferred agent of $x$ (that is, $\mathsf{rk}_x(y) = 1$). Now, use at most one swap to make agent $y$ the most preferred agent of $x$, and at most $n - 1$ swaps to make agent $x$ the most preferred agent of $y$. This results in $\{x, y\}$ being a blocking pair of $M$. Hence, $M$ is not $d$-robust, as $d \geq n$. □

The next lemma characterizes preference profiles with "maximum" robust matchings. We need two more notions: A preference profile is *position-wise distinct* if no agent has the same rank in the preference lists of two different agents. For instance, the preference profiles in Example 1.1 and Example 1.3 are position-wise distinct, while the one in Example 1.2 is not, since $b_1$ has rank one in the preference lists of both $a_0$ and $x_i$, $i \in [n]$. Note that position-wise distinct preference profiles are closely related to the so-called *Latin squares* in combinatorics in the sense that in every complete and position-wise distinct preference profile, the preference lists on each side form a Latin square. A matching is *top-choice* if each agent is matched to her most preferred partner.

Proposition 2.7. *Every $(n - 1)$-robust matching is top-choice and every profile allowing for an $(n - 1)$-robust matching is position-wise distinct.*

Proof. Let $P$ be a preference profile on two $n$-element sets $U$ and $W$, and let $M$ be an $(n - 1)$-robust matching of $P$.

We first show that $M$ is top-choice. This is clear if each agent finds only one other agent acceptable. Otherwise, there is at least one unmatched pair of agents. Observe that for each unmatched pair $\{x, y\}$ of agents it must hold that

$$\mathsf{rk}_x(y) + \mathsf{rk}_y(x) \geq n, \tag{1}$$

as otherwise, we can perform at most $n - 1$ swaps, $\mathsf{rk}_x(y)$ swaps in $x$'s preference list and $\mathsf{rk}_y(x)$ swaps in $y$'s preference list, to make $x$ and $y$ become each other's most preferred agents. This results in $\{x, y\}$ being a blocking pair of $M$—a contradiction to $M$ being $(n - 1)$-robust.

To show that $M$ is top-choice, let us consider an arbitrary agent $x$. Let $y$ be her most preferred agent, i.e., $\mathsf{rk}_x(y) = 0$. Since the rank of each agent is at most $n-1$, it follows that $\mathsf{rk}_x(y)+\mathsf{rk}_y(x) \leq n - 1$. This implies that $\{x, y\}$ is matched under $M$ as otherwise property (1) would have been violated and $M$ would not be $(n - 1)$-robust.

It remains to prove that $P$ is position-wise distinct. We first consider the case when $n = 2$ and then the case when $n \geq 3$; the case of $n = 1$ is trivial.

$$\begin{aligned} u&: \quad M(u) \succ w \succ \cdots, \\ M(w)&: \quad w \succ \cdots \succ w' \succ \cdots, \\ u'&: \quad M(u') \succ w' \succ \cdots, \end{aligned} \qquad \begin{aligned} M(u)&: \quad u \succ \cdots, \\ w&: \quad M(w) \succ \cdots \succ u, \\ w'&: \quad M(w') \succ \cdots \succ u'. \end{aligned}$$

Fig. 1. Illustration of the preference lists of the agents to show that each agent from $U$ finds at least two agents acceptable in the proof of Proposition 2.7.

**Case 1:** $n = 2$**.** Since $M$ is top-choice, we infer that for each two agents their most preferred partners are different. Since $n = 2$, this means that each of these partners is put once in the top position and once in a position that is different than the top one. That is, $P$ is position-wise distinct.
**Case 2:** $n \geq 3$**.** We first claim that each agent from $U$ finds at least two agents acceptable. To this end, consider an arbitrary agent $u$ who finds at least two agents acceptable; note that if such an agent does not exist, then the profile is obviously position-wise distinct because $M$ is top-choice. Let $w$ denote the agent that is at the second position in the preference list of $u$, i.e., $\mathsf{rk}_u(w) = 1$. Since $M$ is top-choice, it follows that $\{u, w\}$ is an unmatched pair. By property (1), we have $\mathsf{rk}_w(u) \geq n-1$ and indeed $\mathsf{rk}_w(u) = n-1$, since $n-1$ is the largest-possible rank. This implies that $w$ has complete preferences and finds all agents from $U$ acceptable. By the symmetry of acceptability, $w$ appears in the preference list of every agent from $U$. By the top-choice property of $M$, we infer that every agent $x$ from $U \setminus \{M(w)\}$ finds at least two agents acceptable: her partner $M(x)$ and agent $w$.

We claim that indeed $M(w)$ also finds at least two other agents acceptable. Since $n \geq 3$, there is a third agent $u' \in U \setminus \{M(w), u\}$ who finds $w$ acceptable. Let $w' \in W$ be an agent with $\mathsf{rk}_{u'}(w') = 1$. Again by property (1), this implies that $\mathsf{rk}_{w'}(u') = n-1$, i.e., $w'$ finds every agent from $U$ acceptable, including $M(w)$. By the symmetry of acceptability, $M(w)$ also finds $w'$ acceptable. Since $u \neq u'$ and $\mathsf{rk}_w(u) = n-1$, we infer that $w' \neq w$, because $\mathsf{rk}_{w'}(u') = n-1$. This implies that $M(w)$ also finds at least two agents acceptable, namely, $w$ and $w'$. See Figure 1 for an illustration.

We have just shown that each agent from $U$ finds at least two agents acceptable. Next, we show that the agents that are ranked at the second positions by the agents from $U$ are distinct among each other. Towards a contradiction assume there are three agents, $u, u' \in U$ and $w \in W$, such that $\mathsf{rk}_u(w) = \mathsf{rk}_{u'}(w) = 1$. Using property (1), it follows that $\mathsf{rk}_w(u) = \mathsf{rk}_w(u') = n-1$ (recall that the maximum value of a rank is $n-1$). Since no two distinct agents can have the same rank in the same preference list, we get a contradiction. In other words, each agent $u$ from $U$ has a distinct agent $w$ with rank one, i.e.,

$$\text{for each agent } w \in W \text{ there exists a unique agent } u \text{ with } \mathsf{rk}_u(w) = 1. \tag{2}$$

Finally, we show that each agent has complete preferences. Using properties (1) and (2), we infer that the agent $w$ that is ranked in the second position by $u$ satisfies $\mathsf{rk}_w(u) = n-1$, implying that she has complete preferences. By the symmetry of acceptability, each agent $u \in U$ must also have complete preferences.

We are now ready to prove that $P$ is position-wise distinct. Since $n \geq 3$, there exists at least one unmatched pair of agents; recall that the preference list of each agent is complete. We will show the stronger statement that, for each unmatched pair $\{u, w\}$ with $u \in U$ and $w \in W$ and for each $z \in [n-1]$ it holds that

$$\mathsf{rk}_u(w) = z \text{ if and only if } \mathsf{rk}_w(u) = n - z. \tag{3}$$

To see that Equation (3) implies that $P$ is position-wise distinct, suppose, for the sake of contradiction, that there are three distinct agents $x_1, x_2, y$ and an integer $z$ such that $\mathsf{rk}_{x_1}(y) = \mathsf{rk}_{x_2}(y) = z$.

Since $M$ is top-choice, $z > 0$, thus $y$ is not matched neither to $x_1$ nor to $x_2$. By Equation (3) $\mathsf{rk}_y(x_1) = \mathsf{rk}_y(x_2) = n - z$, a contradiction.

We show Equation (3) via induction on the rank index $z := \mathsf{rk}_u(w)$, starting with the base case $z = 1$. To this end, let $\{u, w\}$ be an unmatched pair. To show the "only if" part of Equation (3), assume that $z = \mathsf{rk}_u(w) = 1$. By Equation (1), it follows that $\mathsf{rk}_w(u) \geq n - 1$. Since the rank of each agent is at most $n - 1$, it follows that $\mathsf{rk}_w(u) = n - 1 = n - z$.

For the "if" part of the base case suppose, towards a contradiction, that there is an unmatched pair $\{u, w\}$ with $\mathsf{rk}_w(u) = n - 1$ but $\mathsf{rk}_u(w) \neq 1$. By Equation (1), it follows that $\mathsf{rk}_u(w) > 1$. By Equation (2), there exists another agent $u' \in U \setminus \{u\}$ with $\mathsf{rk}_{u'}(w) = 1$. However, then by the "if" part of the base case, it follows that $\mathsf{rk}_w(u') = n-1$, a contradiction. Thus, Equation (3) holds when $z = 1$.

For the induction assumption, assume that Equation (3) holds for every index $z' \leq z - 1$. For the "only if" part, consider an unmatched pair $\{u, w\}$ with $\mathsf{rk}_u(w) = z$. By Equation (1), it follows that $\mathsf{rk}_w(u) \geq n - z$. Suppose, for the sake of contradiction, that $\mathsf{rk}_w(u) = n - z'$ with $z' < z$. By the "if" part of the induction assumption, we infer that $\mathsf{rk}_u(w) = z' < z$, a contradiction.

The "if" part of the induction step follows analogously. □

*2.2.2 Tradeoffs.* Now, we discuss the tradeoffs formalized in the problems regarding near stability and social optimality. As mentioned in Example 1.2 even a single swap in the preference profile can improve the egalitarian cost of the stable matching by $\Omega(n^2)$. However, this is not the case when the social optimality is measured by the number of agents who will have a partner in the matching.

Theorem 2.8. *Let $P_1$ and $P_2$ be two preference profiles with $\tau(P_1, P_2) = 1$. Let $S_1$ and $S_2$ denote the set of agents that are unmatched by any stable matching of $P_1$ and of $P_2$, respectively. Then, $|(S_1 \setminus S_2) \cup (S_2 \setminus S_1)| \leq 2$.*

Proof. Without loss of generality, assume that profile $P_2$ is obtained from $P_1$ by swapping agents $w_1$ and $w_2$ in the preference list of agent $u_1$ so that $u_1$ prefers $w_1$ to $w_2$ in $P_1$ and $w_2$ to $w_1$ in $P_2$.

Since all stable matchings in a fixed preference profile match the same set of agents (see Theorem 2.5), to show the statement, it suffices to show that $P_1$ and $P_2$ admit stable matchings $M_1$ and $M_2$, respectively, such that the following is satisfied: Let $S_1$ and $S_2$ denote the set of agents that are unmatched under $M_1$ and $M_2$, respectively. Then, $|(S_1 \setminus S_2) \cup (S_2 \setminus S_1)| \leq 2$.

To achieve this, we start with an arbitrary but fixed stable matching, $M_1$, of $P_1$. And we will show how to modify $M_1$ to obtain a stable matching of $P_2$ such that at most two more different agents become unmatched. By symmetry, this will prove the theorem.

Observe that if $M_1$ is not stable for profile $P_2$, then $M(u_1) = w_1$ and $\{u_1, w_2\}$ is the only blocking pair. This follows from the fact that $P_1$ and $P_2$ differ only by one single swap of the preference list of agent $u_1$. If $\{u_1, w_2\}$ is blocking $M$ in $P_2$, then we modify $M_1$ in the following way: We break the pairs $\{u_1, M_1(u_1)\}$ and $\{M_1(w_2), w_2\}$ (if the corresponding pair exists) and we add $\{u_1, w_2\}$ to $M_1$. Now, there are at most two new unmatched agents: $M_1(w_2) \in U$ and $M_1(u_1) \in W$. Further, if we remove these agents from the consideration, then the matching would be stable.

We now proceed as follows: We will perform a sequence of changes to $M_1$. After each change, we will keep in a *penalty box* at most two unmatched agents, at most one from $U$ and at most one from $W$, starting with $M_1(w_2)$ and $M_1(u_1)$ (insofar they exist). Further, each change will keep satisfying the following invariant: If we remove the agents contained in the penalty box, then the matching would be stable in the resulting profile. Let us now describe how we perform the changes. In the first sequence of changes, we will keep replacing a $U$-agent from the penalty box with another $U$-agent. If at some point the penalty box would not contain an agent from $U$, then

we stop and move to the second sequence of replacements, this time replacing a $W$-agent in the penalty box. Let $M$ be the matching at the current iteration. We take out an agent $u \in U$ from the penalty box (again, if the penalty box does not contain an agent from $U$, then we move to the second sequence of changes where we replace the agents from $W$ in the penalty box; if the penalty box is empty, then we stop). Agent $u$ might be involved in a number of blocking pairs—if it is not, then we simply remove this agent from the penalty box and move to the second sequence of changes. If $u$ is involved in at least one blocking pair, then we take $u$'s most preferred agent $w \in W$ such that $\{u, w\}$ is a blocking pair of the current matching $M$—we remove $\{M(w), w\}$ (if it exists) from the matching $M$ and add $\{u, w\}$ to $M$. Finally, we add $M(w)$ (if it exists) to the penalty box. Clearly, $u$ cannot be involved in any blocking pair, thus, any blocking pair must involve an agent from the penalty box; hence, the invariant is indeed satisfied.

Each such a change replaces a $U$-agent from the penalty box with another $U$-agent or removes a $U$-agent from the penalty box. Further each replacement improves one of the $W$-agents by giving her a more preferred partner. Thus, our procedure must stop at some point. When this is the case, we remove the $U$-agent from the penalty box.

Then, we perform the second analogous procedure, but each time replacing a $W$-agent in the penalty box with another $W$-agent, until the penalty box is empty or until it contains only an agent that is not involved in any blocking pair. By an analogous argument, such changes keep the invariant satisfied and the procedure finally stops.

Clearly, when the procedure stops, there are no blocking pairs. Further, the resulting matching $M_2$ has at most two more agents without partners than $M_1$. This follows from the fact that we removed at most one agent from $U$ and at most one agent from $W$, not matching them with a partner. □

Repeated application of Theorem 2.8 yields that, to increase the number of matched agents by $\ell \in \mathbb{N}$ in a given stable matching of some profile, we have to allow for at least $\ell/2$ swaps. In other words, if a stable matching leaves $s$ agents unmatched, then there is a globally $d$-nearly stable *perfect* matching only if $d \geq s/2$.

*2.2.3 Alternative Characterizations.* Next, let us discuss the relation between local $d$-near stability and $r$-maximal stability. A matching $M$ is *$r$-maximally stable* [17][2] if for each unmatched pair $\{u, v\} \notin M$, it holds that $\min\{\mathsf{rk}_u(M(u)) - \mathsf{rk}_u(v), \mathsf{rk}_v(M(v)) - \mathsf{rk}_v(u)\} \leq r$. At first glance, this notion looks quite different from local $d$-near stability; we show below that in fact they are equivalent for perfect matchings. A difference is that an $r$-maximal stable matching may have a pair of agents that are both unmatched and yet are acceptable to each other, as long as these two agents rank each other in one of the $d+1$ last places. This cannot occur for a locally $r$-nearly stable matching, because such a pair of agents would induce a blocking pair and no set of swaps would remove the blocking pair.

Moreover, local $\mathsf{d_L}$-near stability is equivalent to the following measure of the weight of a blocking pair: We say that a matching $M$ is *$\mathsf{d_L}$-nearly bp stable* if for each blocking pair $b \in \mathcal{BP}(P, M)$, there exists a profile $P'_b$ such that $b \notin \mathcal{BP}(P'_b, M)$ and $\tau(P, P'_b) \leq \mathsf{d_L}$.

We start by showing that, for perfect matchings, $r$-maximal stability implies local $r$-near stability.

Proposition 2.9. *Let $P$ be a preference profile without ties, and $\mathsf{d_L}$ a non-negative integer. Let $M$ be a perfect matching for $P$. If $M$ is $\mathsf{d_L}$-maximally stable, then it is locally $\mathsf{d_L}$-nearly stable.*

[2]Drummond and Boutilier [17] were concerned with perfect matchings, whereas the definition here is extended to the general case.

Proof. Construct a directed graph $G$ on the set $V$ of agents as follows: For each blocking pair $\{u, v\} \in \mathcal{BP}(P, M)$ find the agent, say $u$, such that

$$\mathsf{rk}_u(M(u)) - \mathsf{rk}_u(v) \leq \mathsf{rk}_v(M(v)) - \mathsf{rk}_v(u)$$

and add the arc $(v, u)$ to $G$ (that is, add an arc directed towards $u$).

Intuitively, since $M$ is perfect, we have that $M(u)$ exists, and to stop $\{u, v\}$ from blocking $M$ it suffices to "shift" $M(u)$ forward to just before $v$ in $u$'s preference list and this costs at most as many swaps than doing the analogous shift of $M(v)$ in $v$'s list.

To obtain a modified profile $P'$ in which $M$ is stable, define, for each agent $u$, a set of swaps in $u$'s preference list as follows: Let $B_u$ denote the set consisting of all agents $v$ such that $(v, u)$ is an arc in $G$. If $B_u$ is empty, then no swaps are performed in $u$'s preference list. Otherwise, pick $w := \operatorname{argmin}_{v \in B_u} \mathsf{rk}_u(v)$, i.e., $w$ is the highest ranked agent from $B_u$. Since $M$ is $\mathsf{d_L}$-maximally stable, we have

$$\min\{\mathsf{rk}_u(M(u)) - \mathsf{rk}_u(w), \mathsf{rk}_w(M(w)) - \mathsf{rk}_w(u)\} \leq \mathsf{d_L}.$$

Since $(w, u)$ is an arc in $G$, we have $\mathsf{rk}_u(M(u)) - \mathsf{rk}_u(w) \leq \mathsf{d_L}$. Since $M$ is perfect, $M(u)$ exists. Swap $M(u)$ in $u$'s preference list with the agent directly preceding $M(u)$ until $M(u) \succ_u^{P'} w$ holds in the resulting profile $P'$. In this way, for each agent, we have made at most $\mathsf{d_L}$ swaps to obtain $P'$.

Note that throughout the swapping process no new blocking pairs are introduced, that is, $\mathcal{BP}(P', M) \subseteq \mathcal{BP}(P, M)$, because in each step only a matched agent improves her rank in its matched partner's preference list. Moreover, for each blocking pair $\{u, v\} \in \mathcal{BP}(P, M)$, we have either $M(u) \prec_u^{P'} v$ or $M(v) \prec_v^{P'} u$ by construction. Thus, $M$ is stable with respect to $P'$, showing that $M$ is locally $\mathsf{d_L}$-nearly stable. □

Now, we turn to the reverse direction in matchings that do not need to be perfect and to the relations to near bp stability.

Proposition 2.10. *Let $P$ be a preference profile without ties, $M$ be a matching for $P$, and $\mathsf{d_L}$ a non-negative integer.*

*(i) If $M$ is locally $\mathsf{d_L}$-nearly stable, then it is $\mathsf{d_L}$-nearly bp stable.*
*(ii) If $M$ is $\mathsf{d_L}$-nearly bp stable, then it is $\mathsf{d_L}$-maximally stable.*
*(iii) If $M$ is $\mathsf{d_L}$-nearly bp stable, then it is locally $\mathsf{d_L}$-nearly stable.*

Proof. (i): Let $P'$ be a profile as promised by local $\mathsf{d_L}$-near stability. To obtain $P'$ from $P$, for each blocking pair $\{u, v\} \in \mathcal{BP}(P, M)$ either $M(v)$ has been swapped before $u$ in $v$'s preference list or $M(u)$ has been swapped before $v$ in $u$'s preference list. In either case, at most $\mathsf{d_L}$ swaps were used. Restricting the swaps made to obtain $P'$ to only those swaps of the case that applied yields a profile $P'_{\{u,v\}}$ as required by $\mathsf{d_L}$-nearly bp stability.

(ii): Let $\{u, v\}$ be an unmatched pair under $M$. If $\{u, v\} \notin \mathcal{BP}(P, M)$, then

$$\min\{\mathsf{rk}_u(M(u)) - \mathsf{rk}_u(v), \mathsf{rk}_v(M(v)) - \mathsf{rk}_v(u)\} \leq 0 \leq \mathsf{d_L},$$

as required for such pairs by $\mathsf{d_L}$-maximal stability. Otherwise, $\{u, v\} \in \mathcal{BP}(P, M)$. By $\mathsf{d_L}$-nearly bp stability, there exists a profile $P'$ with $\tau(P, P') \leq \mathsf{d_L}$ such that $\{u, v\} \notin \mathcal{BP}(P', M)$. This means that $M(u) \succ_u^{P'} v$ or $M(v) \succ_v^{P'} u$ holds. Since $\{u, v\} \in \mathcal{BP}(P, M)$, meaning that both $v \succ_u^P M(u)$ and $u \succ_v^P M(v)$ hold, and since $\tau(P, P') \leq \mathsf{d_L}$, we have $\mathsf{rk}_u(M(u)) - \mathsf{rk}_u(v) \leq \mathsf{d_L}$ or $\mathsf{rk}_v(M(v)) - \mathsf{rk}_v(u) \leq \mathsf{d_L}$. In other words, $M$ is $\mathsf{d_L}$-maximally stable.

(iii): The proof is similar to Proposition 2.9. We first claim that for each blocking pair $b \in \mathcal{BP}(P, M)$ there is an agent $u \in b$ such that $u$ is matched and, letting $v$ be the other agent in $b$, we have

$$\mathsf{rk}_u(M(u)) - \mathsf{rk}_u(v) \leq \mathsf{d_L}.$$

Indeed, otherwise, no set of $\mathsf{d_L}$ swaps will result in a profile in which $b$ does not block $M$.

Construct a directed graph $G$ on the set $V$ of agents as follows: For each blocking pair $\{u, v\} \in \mathcal{BP}(P, M)$ find the agent, say $u$, such that $u$ is matched and such that either $v$ is not matched, or

$$\mathrm{rk}_u(M(u)) - \mathrm{rk}_u(v) \leq \mathrm{rk}_v(M(v)) - \mathrm{rk}_v(u).$$

Add the arc $(v, u)$ to $G$ (that is, add an arc directed towards $u$).

To obtain a modified profile $P'$ in which $M$ is stable, define, for each agent $u$, a set of swaps in $u$'s preference list as follows: Let $B_u$ denote the set consisting of all agents $v$ such that $(v, u)$ is an arc in $G$. If $B_u$ is empty, then no swaps in $u$'s preference list are performed. Otherwise, $u$ is matched. Pick $w \coloneqq \mathrm{argmin}_{v \in B_u} \mathrm{rk}_u(v)$, i.e., $w$ is the highest ranked agent from $B_u$. By the above observation on blocking pairs, we have that

$$\min\{\mathrm{rk}_u(M(u)) - \mathrm{rk}_u(w), \mathrm{rk}_w(M(w)) - \mathrm{rk}_w(u)\} \leq \mathsf{d_L}.$$

Since $(w, u)$ is an arc in $G$, we have $\mathrm{rk}_u(M(u)) - \mathrm{rk}_u(w) \leq \mathsf{d_L}$. Swap $M(u)$ in $u$'s preference list with the agent directly preceding $M(u)$ until $M(u) \succ_u^{P'} w$ holds in the resulting profile $P'$. In this way, in each agent's preference list, we have made at most $\mathsf{d_L}$ swaps to obtain $P'$.

Note that throughout the swapping process no new blocking pairs are introduced, that is, $\mathcal{BP}(P', M) \subseteq \mathcal{BP}(P, M)$, because in each step only a matched agent improves her rank in its matched partner's preference list. Moreover, for each blocking pair $\{u, v\} \in \mathcal{BP}(P, M)$, we have either $M(u) \prec_u^{P'} v$ or $M(v) \prec_v^{P'} u$ by construction. Thus, $M$ is stable with respect to $P'$, showing that $M$ is locally $\mathsf{d_L}$-nearly stable. □

Combining Proposition 2.9 and 2.10 in particular yields that all three stability notions are equivalent for perfect matchings.

## 3 A POLYNOMIAL-TIME ALGORITHM FOR FINDING ROBUST MATCHINGS

In this section, we present a polynomial-time algorithm for the Robust Matching problem. First, in Section 3.1, we provide a brief overview of tools and results from the literature that we will use in our algorithm. Crucially, we recall rotations, the rotation poset, and their relation to stable matchings. Informally, *rotations* are cyclic shifts of agents in a stable matching, so that another stable matching is obtained. The set of rotations is *partially ordered* by the relation of whether one rotation needs to be carried out for the other to be applicable. This structure helps later on to find specific (i.e., robust) stable matchings: Indeed, each stable matching corresponds to a so-called *closed* subset of rotations, i.e., a subset $S$ of rotations such that all predecessors of rotations in $S$ are also in $S$ [26]. We will later on modify the rotation poset and augment it with further information so that, roughly, robust matchings correspond to closed subsets of rotations, for this modified rotation poset.

The results described in the subsequent sections, starting from Section 3.2, are our original contributions. First, in Section 3.2, we show that, instead of all $O(n^d)$ possible profiles that differ by at most $d$ swaps from the input, we can concentrate only on $O(n^4)$ *relevant* preference profiles, and we study their properties. Intuitively, it is enough to look at each of the $O(n^2)$ pairs of agents that are matched in some stable matching for the input profile and the $O(n^2)$ sets of swaps that make them into a blocking pair for some matching. Second, in Section 3.3, we show how these relevant profiles relate to rotations: For each relevant profile, we will identify at most two special rotations. These special rotations either need to be present in a closed subset of rotations corresponding to a robust matching, or they cannot be present, or if one of the two rotations is present, then the other needs to be present as well. Finally, in Section 3.4, we present algorithms tying together all the structural properties we have observed before and derive efficient running time guarantees.

### 3.1 Preliminaries

We remark that the results below were originally stated for *complete* preferences by Gusfield and Irving [26] and Gale and Shapley [22]. Nevertheless, since all stable matchings match the same set of agents (Theorem 2.5), we can verify that they also hold when the preferences may be incomplete.

Theorem 3.1 ([26, Theorem 3.4.3]). *For each preference profile, after $O(n^4)$ preprocessing time, determining whether two pairs $p_1, p_2$ belong to the same stable matching can be done in $O(1)$ time.*

*Optimal Stable Matchings.* As already observed in the literature, the set of all stable matchings for a given preference profile forms a lattice—a specific partially ordered set (poset)—that is useful in designing algorithms for finding special kinds of stable matchings (see Example 3.8 and Figure 2 for an example). The maximum and minimum elements are so-called optimal stable matchings: Consider a preference profile $P$ with two sets, $U$ and $W$, of agents and consider two matchings $M$ and $M'$. We say that an agent $a \in U \cup W$ *prefers $M$ to $M'$*, denoted as $M \succ_a M'$, if $\mathrm{rk}_a(M(a)) < \mathrm{rk}_a(M'(a))$. Similarly, agent $a$ *weakly prefers $M$ to $M'$*, denoted as $M \succeq_a M'$, if $M(a) = M'(a)$ or $\mathrm{rk}_a(M(a)) < \mathrm{rk}_a(M'(a))$. Accordingly, we say that $M$ is a *$U$-optimal* (respectively, *$W$-optimal*) stable matching if it is stable and there is *no* other stable matching $M'$ different from $M$ such that each agent from $U$ (respectively, from $W$) prefers $M'$ to $M$.

It is well-known that $U$-optimal and $W$-optimal stable matchings are unique. The concepts of $U$-optimal and $W$-optimal stable matchings are already illustrated in Example 1.1.

Theorem 3.2 below shows that, when comparing two stable matchings, an improvement of an agent $u \in U$ always comes at the cost of some other agent from $W$.

Theorem 3.2 ([26, Theorem 1.3.1, Section 1.4.2]). *Let $M_1$ and $M_2$ be two stable matchings of the same preference profile with (possibly) incomplete preferences, and let $u$ and $w$ be two agents such that $M_1(u) = w$ but $M_2(u) \neq w$. Then, $M_1 \succ_u M_2$ if and only if $M_2 \succ_w M_1$.*

Finally, we recall that the famous Gale/Shapley algorithm always finds the $U$-optimal (or, depending on the variant of the algorithm used, the $W$-optimal) stable matching.

Theorem 3.3 ([22], [26, Section 1.4.2]). *The $U$-optimal and the $W$-optimal stable matchings of a preference profile with $2n$ agents always exist and can be found in $O(n^2)$ time.*

*Rotations and the Rotation Poset.* We now review a fundamental object, *rotations*, and some well-known structural properties of stable matchings. These concepts will play an instrumental role in our analysis in the subsequent sections. For more details, we refer to the exposition by Gusfield and Irving [26].

*Definition 3.4 (Successor agent, rotations, and rotation elimination).* Let $P$ be a preference profile with two disjoint sets of agents, $U$ and $W$, and with (possibly) incomplete preferences. Given a stable matching $M \in \mathcal{SM}(P)$, for each agent $u \in U$, we define its *successor* $\mathrm{succ}_M(u)$ as the *first* (after $M(u)$) agent $w$ on the preference list of $u$ such that $w$ is matched under $M$ and prefers $u$ to its partner $M(w)$. An ordered sequence $\rho = ((u_0, w_0), (u_1, w_1), \ldots, (u_{r-1}, w_{r-1}))$ of pairs is called a *rotation* if there exists a stable matching $M \in \mathcal{SM}(P)$ such that for each $i \in \{0, 1, \ldots, r-1\}$, we have $(u_i, w_i) \in U \times W$, $M(u_i) = w_i$, and $\mathrm{succ}_M(u_i) = w_{i+1}$ (index $i+1$ taken modulo $r$). Accordingly, we say rotation $\rho$ is *exposed* in $M$.

We use the notation $M/\rho$ to refer to the matching resulting from $M$ by replacing each pair $\{u_i, w_i\}$ with $\{u_i, w_{i+1}\}$. Formally,

$$M/\rho = M \setminus \{\{u_i, w_i\} \mid 0 \leq i \leq r-1\} \cup \{\{u_i, w_{i+1}\} \mid 0 \leq i \leq r-1\},$$

where "$i+1$" is taken modulo $n$. The transformation of $M$ to $M/\rho$ is called the *elimination of $\rho$ from $M$*.

If an agent $u$ has successor $\text{succ}_M(u)$, then their preference lists can be illustrated as follows:

$$u\colon \ \ldots M(u)\ldots \text{succ}_M(u)\ldots$$
$$\text{succ}_M(u)\colon \ \ldots u\ldots M(\text{succ}_M(u))\ldots.$$

Eliminating a rotation from a stable matching results in another stable matching [26]. The concepts from Definition 3.4 are illustrated in the example below.

*Example 3.5.* Consider the profile in Example 1.1. Relative to $M_1$, agent $w_3$ is the first agent among all agents in the preference list of $u_1$ that prefer $u_1$ to their respective partners. Indeed, $w_1 = M_1(u_1)$ and agent $w_2$ prefers its partner $M_1(w_2) = u_2$ to $u_1$. Thus, $\text{succ}_{M_1}(u_1) = w_3$. The sequence $((u_1, w_2), (u_2, w_3), (u_3, w_4), (u_4, w_1))$ is the only rotation exposed in $M_1$. ◇

Interestingly, while a given profile with $O(n)$ agents may admit exponentially ($\Omega(2^n)$) many different stable matchings [29, 32], the number of rotations is polynomial ($O(n^2)$) [26, Corollary 3.2.1]. Indeed, the set of all rotations gives a compact representation of the set of all possible stable matchings for a given preference profile. To determine robustness efficiently, we will use this representation intensely.

The next structural result concerns the properties of a stable matching after eliminating a rotation.

Theorem 3.6 ([26, Theorem 2.5.6, Lemma 3.2.1, Lemma 3.2.2]). *Consider a preference profile $P$ with two disjoint sets of agents, $U$ and $W$, and with (possibly) incomplete preferences. For each two agents $u \in U$ and $w \in W$ the following holds (recall that $x \succeq y$ means that $x = y$ or $x \succ y$):*

*(i) $\{u, w\}$ is in a stable matching if and only if either it is in the $W$-optimal stable matching or $(u, w)$ belongs to some rotation.*

*(ii) There is at most one rotation $\rho$ with $\rho = ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ such that for some $i \in \{0, \ldots, r-1\}$ it holds that $u = u_i$ and $w_i \succeq_u w \succ_u w_{i+1}$.*

*(iii) There is at most one rotation $\rho$ with $\rho = ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ such that for some $i \in \{0, \ldots, r-1\}$ it holds that $u = u_i$ and $w = w_{i+1}$.*

*(iv) There is at most one rotation $\rho$ with $\rho = ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ such that for some $i \in \{0, \ldots, r-1\}$ it holds that $w = w_i$ and $u_{i-1} \succ_w u \succeq_w u_i$.*

Now, we are ready to introduce the notion of the rotation poset of a given preference profile $P$. As we will see later on, each stable matching can be obtained by performing a number of eliminations of rotations on the $U$-optimal stable matching. When starting from $U$ some rotations can be exposed only after some others have been already eliminated. This induces a partial order on rotations and defines the rotation poset.

*Definition 3.7 (Predecessors of Rotations, the Rotation Poset, and the Rotation Digraph).* Let $\pi$ and $\rho$ be two rotations for a preference profile $P$. We say that $\pi$ is a *predecessor* of $\rho$, written as $\pi \lhd^P \rho$, if no stable matching in which $\rho$ is exposed can be obtained from the $U$-optimal stable matching by a sequence of eliminations of rotations without eliminating $\pi$ first. The reflexive closure of the relation $\lhd^P$, denoted as $\trianglelefteq^P$, defines a partial order on the set of all rotations.

Let $R$ be the set consisting of all rotations for $P$. The pair $(R, \trianglelefteq^P)$ is called the *rotation poset* for $P$. We use *a closed subset* as a shorthand for "a subset of $R$ that is closed under precedence relation $\trianglelefteq^P$."

An alternative representation of the rotation poset $\trianglelefteq(P)$ is through an acyclic directed graph, called the *rotation digraph of $P$* and written as $G(P)$, whose vertex set is the set of rotations of $P$, and there is a direct arc $(\pi, \rho)$ from rotation $\pi$ to rotation $\rho$ if and only if $\pi$ precedes $\rho$ (i.e., $\pi \lhd \rho$) and there is no other rotation $\sigma$ such that $\pi \lhd^P \sigma \lhd^P \rho$.

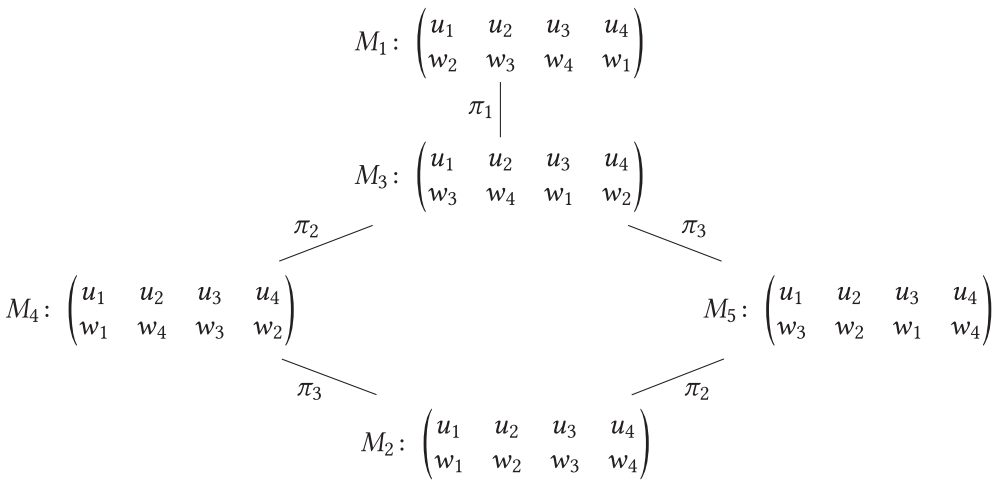



Fig. 2. The lattice structure for the stable matchings of $P$ discussed in Example 3.8.

The following example illustrates the rotation poset of profile given in Example 1.1:

*Example 3.8.* Let us consider the profile $P$ given in Example 1.1 again. As we mentioned in Example 3.5, rotation $\pi_1 = ((u_1, w_2), (u_2, w_3), (u_3, w_4), (u_4, w_1))$ is the only rotation exposed in the $U$-optimal stable matching $M_1$. After eliminating $\pi_1$ from $M_1$, we obtain the stable matching $M_3 = M_1/\pi_1$. One can also verify that the sequences $\pi_2 = ((u_1, w_3), (u_3, w_1))$ and $\pi_3 = ((u_2, w_4), (u_4, w_2))$ are the only two rotations exposed in $M_3$. After eliminating $\pi_2$ from $M_3$, we obtain the stable matching $M_4 = M_3/\pi_2$. After eliminating $\pi_3$ from $M_3$, we obtain the stable matching $M_5 = M_3/\pi_3$. After eliminating rotation $\pi_3$ from $M_4$ or eliminating the rotation $\pi_2$ from $M_5$, we obtain the $W$-optimal stable matching $M_2$.

Since $\pi_1$ is only exposed in $M_1$ and since $\pi_2$ and $\pi_3$ are only exposed after the elimination of $\pi_1$, we infer that $\pi_1$ is the direct predecessor of both $\pi_2$ and $\pi_3$.

The diagram in Figure 2 depicts the lattice structure of the stable matchings for $P$, in terms of dominance with respect to the satisfaction of the agents from $U$. Herein, the matchings are depicted as matrices such that each pair in a matching is represented by a column in the corresponding matrix. ◇

Finally, let us describe a central result from the literature that relates rotations and stable pairs.

Theorem 3.9 ([26, Theorem 2.5.7, Lemma 3.3.2]). *Let $R$ denote the set of all rotations of a preference profile $P$, and let $G(P)$ denote the rotation digraph of $P$.*

*(i) A matching $M$ is a stable matching of $P$ if and only if there is a closed subset of rotations $R' \subseteq R$ with respect to the precedence relation $\lhd^P$ such that $M$ can be generated by taking the $U$-optimal stable matching and by eliminating the rotations in $R'$ in an order consistent with $\lhd^P$.*

*(ii) The rotation set $R$ and the rotation digraph $G(P)$ can be computed in $O(n^2)$ time.*

### 3.2 Profile Characterization

For a given profile $P$ with $O(n)$ agents and a given swap distance bound $d = O(n)$ there are exponentially many profiles that are within swap distance of $d$ to $P$. In this section, we show that we do not need to consider all of them to find a $d$-robust matching. We will define a set of $O(n^4)$ "relevant" profiles such that each matching $M$ that is not $d$-robust in $P$ is unstable in at least one of the relevant profiles. Each relevant profile is characterized by a distinct pair of shifts. Herein,

briefly put, a *shift* is a set of swaps that are performed on the input profile $P$ and that all involve swapping the same agent forward in a single preference list. Intuitively, if there exists a profile $P'$ witnessing that a certain matching $M$ is *not* $d$-robust, then $P'$ harbors a blocking pair $b$ for $M$ and $P'$. This blocking pair can be represented by a profile that only includes a pair of shifts that also result in the blocking pair $b$. Such a pair of shifts can be equivalently characterized by a certain tuple of four agents involved in these shifts. We will call such tuples *stable quadruple*. We then show that stable quadruples are closely related to certain rotations. This will give us the tools that are essential for constructing a polynomial-time algorithm.

*Definition 3.10 (Stable quadruples and swap sets).* Let $P = ((\succ_u^P)_{u\in U}, (\succ_w^P)_{w\in W})$ be a preference profile for the two agent sets $U$ and $W$. A *stable quadruple* (with respect to $P$) is a quadruple $(u^*, w^*, u, w)$ of four distinct agents with $u^*, u \in U$ and $w^*, w \in W$ such that (i) $u^*$ and $w^*$ are acceptable to each other and (ii) there exists a stable matching for $P$ that contains both $\{u^*, w\}$ and $\{u, w^*\}$.

For each stable quadruple $q = (u^*, w^*, u, w)$ of $P$, we define the *swap set* associated with $P$ and $q$, denoted as $\mathsf{swaps}(P, q)$, as follows:

$$\mathsf{swaps}(P,q) := \left\{(u^*, \{y, w^*\}) \mid y \in W : w \succeq_{u^*}^P y \succ_{u^*}^P w^*\right\} \cup \left\{(w^*, \{u^*, x\}) \mid x \in U : u \succeq_{w^*}^P x \succ_{w^*}^P u^*\right\},$$

where the notation $x \succeq y$ means either $x = y$ or $x \succ y$.

Note that the swap set $\mathsf{swaps}(P, q)$ is the smallest set of swaps that involve the following two kinds of shifts in the preference lists of $u^*$ and $w^*$:

(i) The first kind of shifts puts agent $w^*$ forward until she is right in front of $w$ in the preference list of $u^*$, and
(ii) the second kind of shifts puts agent $u^*$ forward until she is right in front of $u$ in the preference list of $w^*$.

If $w^*$ (respectively, $u^*$) is already in front of $w$ (respectively, $u$), then no swap in the corresponding preference list is needed.

Intuitively, if $\{u^*, w\}$ and $\{u, w^*\}$ are in a stable matching $M$, then the above two shifts will result in $M$ being unstable. If the shifts incorporate at most $d$ swaps, then the matching cannot be $d$-robust. Hence, stable quadruples with at most $d$ swaps in their swap sets form an obstruction that needs to be avoided.

*Definition 3.11 (Profile associated with a stable quadruple).* Let $P = ((\succ_u^P)_{u\in U}, (\succ_w^P)_{w\in W})$ be a preference profile for the two agent sets $U$ and $W$, and let $q = (u^*, w^*, u, w)$ be a stable quadruple of $P$. Then, define $\mathsf{slist}(\succ_{u^*}^P, q)$ as the preference list resulting from starting with $\succ_{u^*}^P$ and performing the swaps from $\mathsf{swaps}(P, q)$ that involve the preference list of $u^*$. Analogously, define $\mathsf{slist}(\succ_{w^*}^P, q)$ as the preference list resulting from starting with $\succ_{w^*}^P$ and performing the swaps from $\mathsf{swaps}(P, q)$ that involve the preference list $\succ_{w^*}^P$. Now, define the preference profile associated with quadruple $q$, denoted as $\mathsf{Prof}[\mathsf{swaps}(P, q)]$, as follows:

$$\mathsf{Prof}[\mathsf{swaps}(P,q)] := \left(\left(\succ_x^P\right)_{x\in U\setminus\{u^*\}} + \mathsf{slist}\left(\succ_{u^*}^P, q\right), \left(\succ_y^P\right)_{y\in W\setminus\{w^*\}} + \mathsf{slist}\left(\succ_{w^*}, q\right)\right).$$

Briefly, $\mathsf{Prof}[\mathsf{swaps}(P, q)]$ is the preference profile resulting from $P$ by replacing the preference lists of $u^*$ and $w^*$ with $\mathsf{slist}(\succ_{u^*}^P, q)$ and $\mathsf{slist}(\succ_{w^*}^P, q)$, respectively.

*Example 3.12.* For an illustration, let us consider the profile given in Example 1.1 on page 2, denoted as $P = ((\succ_{u_i}^P)_{u_i\in U}, (\succ_{w_i}^P)_{w_i\in W})$, and the following stable quadruple $q = (u_3, w_2, u_4, w_1)$; note that $\{u_3, w_1\}$ and $\{u_4, w_2\}$ belong to the same stable matching $M_3$ (see Example 3.8 on

page 19). The swap set $\mathsf{swaps}(P, q)$ consists of two swaps; both involve changing $u_3$'s preference list: $\mathsf{swaps}(P, q) = \{(u_3, \{w_2, w_3\}), (u_3, \{w_2, w_1\})\}$.

If we perform the swaps given in $\mathsf{swaps}(P, q)$ on $\succ^P_{u_3}$ and on the preference profile, then we obtain that

$$\mathsf{slist}\left(\succ^P_{u_3}, q\right) = \{u_3 \colon w_4 \succ w_2 \succ w_1 \succ w_3\}, \text{ and}$$
$$\mathsf{Prof}[\mathsf{swaps}(P, q)] = \left(\left(\succ^P_{u_1}, \succ^P_{u_2}, \mathsf{slist}\left(\succ^P_{u_3}, q\right), \succ^P_{u_4}\right), \left(\succ^P_{w_1}, \succ^P_{w_2}, \succ^P_{w_3}, \succ^P_{w_4}\right)\right).$$

Finally, we observe that in $\mathsf{Prof}[\mathsf{swaps}(P, q)]$, $u_3$ prefers $w_2$ to $w_1$, and $w_2$ prefers $u_3$ to $u_4$, that is, $M_3$ is not stable in this profile. ◇

We now gather some helpful and straightforward observations about stable quadruples. A stable quadruple $q$ and the corresponding profile $\mathsf{Prof}[\mathsf{swaps}(P, q)]$ satisfy the following properties:

Observation 3.13. *Let $P$ be a preference profile over the two agent sets $U$ and $W$, let $q$ be a stable quadruple with $q = (u^*, w^*, u, w)$ and let $Q = \mathsf{Prof}[\mathsf{swaps}(P, q)]$ denote the preference profile after performing the swaps in the set* $\mathsf{swaps}(P, q)$.

*(i) Each agent $x \in U \cup W \setminus \{u^*, w^*\}$ other than $u^*$ and $w^*$ has $\succ^Q_x = \succ^P_x$.*

*(ii) If $w^* \succ^P_{u^*} w$, then $\succ^Q_{u^*} = \succ^P_{u^*}$.*
*Otherwise, for each two distinct agents $y, z \in W \setminus \{w^*\}$ the following holds:*
*(a) $y \succ^Q_{u^*} z$ if and only if $y \succ^P_{u^*} z$,*
*(b) $y \succ^Q_{u^*} w^*$ if and only if $y \succ^P_{u^*} w$,*
*(c) $w^* \succ^Q_{u^*} y$ if and only if $w \succeq^P_{u^*} y$.*

*(iii) If $u^* \succ^P_{w^*} u$, then $\succ^Q_{w^*} = \succ^P_{w^*}$.*
*Otherwise, for each two distinct agents $y, z \in U \setminus \{u^*\}$ the following holds:*
*(a) $y \succ^Q_{w^*} z$ if and only if $y \succ^P_{w^*} z$,*
*(b) $y \succ^Q_{w^*} u^*$ if and only if $y \succ^P_{w^*} u$,*
*(c) $u^* \succ^Q_{w^*} y$ if and only if $u \succeq^P_{w^*} y$.*

*(iv) In $Q$, agent $u^*$ prefers $w^*$ to $w$, and agent $w^*$ prefers $u^*$ to $u$.*

Informally, we will argue that, to find a $d$-robust matching, it suffices to focus on profiles associated with stable quadruples with swap sets of size at most $d$. We will show that for each such stable quadruple $q = (u^*, w^*, u, w)$ in profile $\mathsf{Prof}[\mathsf{swaps}(P, q)]$, we only need to ensure that $\{u^*, w^*\}$ is not a blocking pair. Before that, Lemmas 3.14 and 3.15 below formalize our intuition that $\{u^*, w^*\}$ is the only possible blocking pair in $\mathsf{Prof}[\mathsf{swaps}(P, q)]$.

Lemma 3.14. *Consider a preference profile $P$ and a stable matching $M$ for $P$. Let $q = (u^*, w^*, u, w)$ be a stable quadruple. Then, the pair $\{u^*, w^*\}$ is the only possible blocking pair of $M$ in* $\mathsf{Prof}[\mathsf{swaps}(P, q)]$.

Proof. Let $Q = \mathsf{Prof}[\mathsf{swaps}(P, q)]$. Suppose, towards a contradiction, that $M$ admits a blocking pair $\{x, y\}$ with $x \in U$ and $y \in W$ in profile $Q$ and that $\{x, y\} \neq \{u^*, w^*\}$. Since $Q$ differs from $P$ only in the preference lists of $u^*$ and $w^*$ and since $M$ is stable in $P$, it follows that either $x = u^*$ or $y = w^*$. If $x = u^*$, implying that $\{u^*, y\}$ is blocking $M$ in $Q$, then it holds that $y \succ^Q_{u^*} M(u^*)$ and $u^* \succ^Q_y M(y)$. However, since $\{x, y\} \neq \{u^*, w^*\}$ it follows that $y \neq w^*$, and that $y \succ^P_{u^*} M(u^*)$ (see Observation 3.13(ii)) and $u^* \succ^P_y M(y)$ (see Observation 3.13(i)). This implies that $\{u^*, y\}$ is also blocking $M$ in $P$, a contradiction to $M$ being stable in $P$. Analogously, we will also derive a contradiction for the case of $x \neq u^*$ and $y = w^*$. □

Lemma 3.15. *Let $P_1$ and $P_2$ be two preference profiles for the same two disjoint sets $U$ and $W$, and let $M \in \mathcal{SM}(P_1)$ be a stable matching of $P_1$. Let $\{u^*, w^*\} \in \mathcal{BP}(P_2, M)$ be a blocking pair for $P_2$ with $u^* \in U$ and $w^* \in W$. Define $q = (u^*, w^*, M(w^*), M(u^*))$. The following holds:*
*(i)* $\mathcal{BP}(\mathrm{Prof}[\mathrm{swaps}(P_1, q)], M) = \{\{u^*, w^*\}\}$.
*(ii)* $|\mathrm{swaps}(P_1, q)| \leq \tau(P_1, P_2)$.

Proof. To show the first statement, assume that $M$ is not stable in $P_2$ and let $\{u^*, w^*\}$ be a blocking pair of $M$ for $P_2$. Set $Q = \mathrm{Prof}[\mathrm{swaps}(P_1, q)]$.

By Observation 3.13(iv), we immediately get that $\{u^*, w^*\}$ is blocking $M$ in profile $Q$. The fact that $\{u^*, w^*\}$ is the only blocking pair follows from Lemma 3.14.

Now let us consider the second statement. By the definition of swap sets on $q$, we have:

$$|\mathrm{swaps}(P_1, q)| = \max\left(\mathrm{rk}_{u^*}\left(w^*, \succ_{u^*}^{P_1}\right) - \mathrm{rk}_{u^*}\left(M(u^*), \succ_{u^*}^{P_1}\right), 0\right) + \max\left(\mathrm{rk}_{w^*}\left(u^*, \succ_{w^*}^{P_1}\right) - \mathrm{rk}_{w^*}\left(M(w^*), \succ_{w^*}^{P_1}\right), 0\right).$$

Since $\{u^*, w^*\}$ is blocking $M$ in $P_2$ but $M$ is stable for $P_1$, it holds that

$$w^* \succ_{u^*}^{P_2} M(u^*) \text{ and } u^* \succ_{w^*}^{P_2} M(w^*), \text{ while } M(u^*) \succ_{u^*}^{P_1} w^* \text{ or } M(w^*) \succ_{w^*}^{P_1} u^*.$$

Thus,

$$\begin{aligned}\tau(P_1, P_2) &\geq \max\left(\mathrm{rk}_{u^*}\left(w^*, \succ_{u^*}^{P_1}\right) - \mathrm{rk}_{u^*}\left(M(u^*), \succ_{u^*}^{P_1}\right), 0\right) \\ &\quad + \max\left(\mathrm{rk}_{w^*}\left(u^*, \succ_{w^*}^{P_1}\right) - \mathrm{rk}_{w^*}\left(M(w^*), \succ_{w^*}^{P_1}\right), 0\right) \\ &= |\mathrm{swaps}(P_1, q)|,\end{aligned}$$

proving the statement. □

Finally, the following result summarizes the intuition that stable quadruples with small swap sets are the only obstructions to $d$-robust matchings and that we only need to focus their associated profiles when searching for a $d$-robust matching.

Lemma 3.16. *Let $P_0$ be a preference profile for two disjoint sets of agents, $U$ and $W$, and let $d \in \mathbb{N}$ be a non-negative integer. A matching $M$ is $d$-robust for profile $P_0$ if and only if for each stable quadruple $q = (u^*, w^*, u, w)$ of $P_0$ such that $|\mathrm{swaps}(P_0, q)| \leq d$, matching $M$ is also stable in* $\mathrm{Prof}[\mathrm{swaps}(P_0, q)]$.

Proof. The "only if" direction is straightforward, because $M$ is stable in each profile $P$ with $\tau(P_0, P) \leq d$ and for each stable quadruple $q = (u^*, w^*, u, w)$ such that $|\mathrm{swaps}(P_0, q)| \leq d$ it holds that $\tau(P_0, \mathrm{Prof}[\mathrm{swaps}(P_0, q)]) = |\mathrm{swaps}(P_0, q)| \leq d$.

For the "if" direction, assume that there is a matching, called $M$, such that for each stable quadruple $q = (u^*, w^*, u, w)$ with $|\mathrm{swaps}(P_0, q)| \leq d$, matching $M$ is stable in $\mathrm{Prof}[\mathrm{swaps}(P_0, q)]$. Suppose, for the sake of contradiction, that there is a preference profile $P$ with $\tau(P_0, P) \leq d$ such that $M$ is *not* stable in $P$. Let $\{x, y\}$ be a blocking pair of $M$ in $P$ with $x \in U$ and $y \in W$. Now let us consider the quadruple $q' = (x, y, M(y), M(x))$. Note that $q'$ is a stable quadruple with respect to $P_0$, since $M$ is stable for $P_0$. Since $\{x, y\} \in \mathcal{BP}(P, M)$, by Lemma 3.15(i), it follows that $\mathcal{BP}(\mathrm{Prof}[\mathrm{swaps}(P_0, q)], M) = \{\{x, y\}\}$ and, by Lemma 3.15(ii), $|\mathrm{swaps}(P_0, q)| \leq \tau(P_0, P) \leq d$—a contradiction to our assumption. □

As a corollary to Lemma 3.16, we obtain a polynomial-time algorithm to check whether a given matching is $d$-robust. Observe that it is not obvious from the definition of $d$-robustness that this can be done in polynomial time, because the trivial way would be to try out all the possible swap sets of size at most $d$, leading to a running time of $\Omega(n^{2d})$.

**Case (a1): $u^*$ prefers $w^*$ to $w$,** i.e., $w^* \succ^{P_0}_{u^*} w$. Then, for $u^*$ : $w_i \succeq w^* \succ w_{i+1}$. ($\pi(q)$)

**Case (a2): $u^*$ prefers $w$ to $w^*$,** i.e., $w \succ^{P_0}_{u^*} w^*$. Then, for $u^*$ : $w_i \succ w = w_{i+1} \succ w^*$. ($\pi(q)$)

**Case (b1): $w^*$ prefers $u^*$ to $u$,** i.e., $u^* \succ^{P_0}_{w^*} u$. Then, for $w^*$ : $u_{i-1} \succeq u^* \succ u_i$. ($\rho(q)$)

**Case (b2): $w^*$ prefers $u$ to $u^*$,** i.e., $u \succ^{P_0}_{w^*} u^*$. Then, for $w^*$ : $u_{i-1} \succ u = u_i \succ u^*$. ($\rho(q)$)

Fig. 3. Illustration of the definitions of $\pi(q)$ (upper two cases) and $\rho(q)$ (lower two cases) associated with a stable quadruple $q$; see Definition 3.17.

### 3.3 Relation between Stable Quadruples and Rotations

Before we state our central results, we need one more element: In this subsection, we define two specific rotations corresponding to a stable quadruple and we investigate their properties pertaining to robustness. The results stated in this subsection might look quite technical, yet we deliberately chose these particular formulations, as we believe they make the analysis of our algorithm transparent.

*Definition 3.17 ($\pi(q)$ and $\rho(q)$ for a stable quadruple $q$).* Let $P_0 = (\succ_x^{P_0})_{x \in U \cup W}$ be a preference profile with two sets of agents, $U$ and $W$, and consider a stable quadruple $q = (u^*, w^*, u, w)$.

We use the notation $\pi(q)$ to refer to a rotation $\pi := ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ with $u^* = u_i$ (for some $i \in \{0, \ldots, r-1\}$) that fulfills the following conditions:
(a1) If $w^* \succ^{P_0}_{u^*} w$, then $w^* = w_i$ or $w_i \succ^{P_0}_{u^*} w^* \succ^{P_0}_{u^*} w_{i+1}$.
(a2) If $w \succ^{P_0}_{u^*} w^*$, then $w = w_{i+1}$.

We use the notation $\rho(q)$ to refer to a rotation $\rho := ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ with $w^* = w_i$ (for some $i \in \{0, \ldots, r-1\}$) that fulfills the following conditions:
(b1) If $u^* \succ^{P_0}_{w^*} u$, then $u^* = u_{i-1}$ or $u_{i-1} \succ^{P_0}_{w^*} u^* \succ^{P_0}_{w^*} u_i$.
(b2) If $u \succ^{P_0}_{w^*} u^*$, then $u = u_i$.

Figure 3 illustrates the two specific rotations; recall that for two agents $x$ and $y$, the expression "$x \succeq y$" means that $x = y$ or $x \succ y$.

*General ideas behind $\pi(q)$ and $\rho(q)$.* Rotations $\pi(q)$ and $\rho(q)$ can be informally described as follows: Consider the preference profile $Q = \mathrm{Prof}[\mathrm{swaps}(P_0, q)]$. Rotation $\pi(q)$ is the first rotation (according to the precedence relation on rotations) that moves the partner of $u^*$ from $w^*$ or from an agent who is more preferred than $w^*$ to an agent that is less preferred than $w^*$, where the preference relation is according to profile $Q$. Similarly, rotation $\rho(q)$ is the first rotation that moves the partner of $w^*$ from an agent who is less preferred than $u^*$ to $u^*$ or to an agent that is more preferred than $u^*$, where the preference relation is according to profile $Q$. However, in Definition 3.17, we deliberately do *not* refer to profile $Q$ and define the rotations $\pi(q)$ and $\rho(q)$ solely based on $P_0$ to make the subsequent formal analysis and the algorithm as clear as possible.

Roughly speaking, eliminating rotation $\pi(q)$ could make a stable matching of the original profile not stable anymore in the new profile $Q$—indeed, this is the "first" rotation whose elimination

causes $u^*$ to prefer $w^*$ over its matched partner. To make sure that $\{u^*, w^*\}$ is not blocking the constructed matching in $Q$, we need to enforce that, whenever the matching includes $\pi(q)$, agent $w^*$ must obtain a partner who she prefers over $u^*$. This is achieved by also eliminating $\rho(q)$ from the matching. In other words, when selecting rotations that should form a robust matching, adding $\rho(q)$ fixes some potential issues that arise as a result of eliminating $\pi(q)$ from the matching. This intuition is formalized in the subsequent lemmas and theorems. While the main idea is intuitive, the formal analysis is complex, since we need to take care of a few technical nuances.

Note that, by our definition, neither $\pi(q)$ nor $\rho(q)$ needs to exist. However, if they exist, then they are unique. We will prove this statement in Lemma 3.19, below. Before that, however, let us go through an example that illustrates the definitions of $\pi(q)$ and $\rho(q)$.

*Example 3.18.* Recall that in Example 3.8, we have derived the rotation poset of the profile given in Example 1.1, and $q = (u_3, w_2, u_4, w_1)$ is the stable quadruple discussed in Example 3.12. In the notation of Definition 3.17, $(u_3, w_2, u_4, w_1) = (u^*, w^*, u, w)$.

Since $u_3$ prefers $w_1$ to $w_2$ (Case (a2) as illustrated in Figure 3), to define $\pi(q)$, we are searching for a rotation that includes $(u_3, x)$ for some agent $x \in W$ such that after the elimination of this rotation $u_3$ receives agent $w_1$ as a partner. Rotation $\pi_1$ is the only rotation that fulfills this condition. Thus, $\pi(q) := \pi_1$. Let $P'$ be the profile resulting from performing the two swaps given in $\mathrm{swaps}(P, q)$. One can verify that in $P'$ agent $u_3$ prefers $w_2$ to $w_1$. Thus, in the same profile $P'$,

(i) either $u_3$ prefers $w_2$ to the partner assigned by a stable matching whose corresponding closed subset of rotations includes $\pi(q)$, or
(ii) the partner of $u_3$ is $w^* = w_2$.

Indeed, after eliminating any successor rotation of $\pi(q)$ either we still assign $w_2$ to $u_3$ or change the partner of $u_3$ to one that is less preferred than $w_1$ in the initial profile, including agent $w^*$, which is more preferred than $w_1$ in $P'$. This means that such kind of stable matching may be blocked by $\{u_3, w_2\}$ in $P'$.

As for $\rho(q)$, since $w_2$ prefers $u_3$ to $u_4$ (Case (b1) as illustrated in Figure 3), we are searching for a rotation such that after eliminating this rotation $w_2$ receives either $u_3$ or a partner that he prefers to $u_3$. This is satisfied by $\pi_3$:

Observe that, after the elimination of this rotation, $w_2$ obtains $u_2$, which is her most preferred agent, i.e., an agent that is strictly better than $u_3$. Again, one can verify that in $P'$ agent $w_2$ still prefers $u_2$ to $u_3$. Thus, $w_2$ prefers its partner, assigned by a stable matching whose corresponding closed subset includes $\rho(q)$, to $u_3$. This means that such kind of stable matching cannot be blocked by $\{u_3, w_2\}$.

For a comparison, let us consider another stable quadruple $q' = (u_1, w_2, u_4, w_1)$; again, in the notation of Definition 3.17 $(u_1, w_2, u_4, w_1) = (u^*, w^*, u, w)$. Since $u_1$ prefers $w_2$ to $w_1$ (Case (a1) as illustrated in Figure 3), we are searching for a rotation that includes $(u_1, x)$ such that

$$\text{either } x = w_2, \tag{4}$$

$$\text{or } u_1 \text{ prefers } x \text{ over } w_2 \text{ in } P \text{ and will obtain a partner that } u_1 \text{ prefers } w_2 \text{ to.} \tag{5}$$

Since $\pi_1$ includes $(u_1, w_2)$, satisfying (4), we have that $\pi(q') = \pi_1$.

As for $\rho(q')$, since $w_2$ prefers $u_4$ to $u_1$, we need to find a rotation that includes $(u_4, w_2)$. Since $\pi_3$ includes $(u_4, w_2)$, we set $\rho(q') := \pi_3$. ◇

*Technical results about $\pi(q)$ and $\rho(q)$.* Rotations $\pi(q)$ and $\rho(q)$ are critical concepts that will be used by our algorithm for finding robust matchings. The next two lemmas provide tools that allow us to use these concepts conveniently. We start by showing that $\pi(q)$ and $\rho(q)$ are unique.

Lemma 3.19. *Let $q = (u^*, w^*, u, w)$ be a stable quadruple of a preference profile $P$. The following holds:*

*(i) If $\pi(q)$ exists, then it is unique.*
*(ii) If $\rho(q)$ exists, then it is unique.*

Proof. For the first statement, assume that rotation $\pi(q)$ exists with $\pi(q) = ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ and $u^* = u_i$. We distinguish between two cases.

**Case (i):** $w^* \succ_{u^*} w$**.** Then, $w^* = w_i$ or $w_i \succ^P_{u^*} w^* \succ^P_{u^*} w_{i+1}$ by definition of $\pi(q)$. In either case, Theorem 3.6(ii) guarantees that $\pi(q)$ is unique.

**Case (ii):** $w \succ_{u^*} w^*$**.** By the definition of $\pi(q)$, we have that $w = w_{i+1}$. By Theorem 3.6 (iii), rotation $\pi(q)$ is unique.

Now, we turn to the second statement. Assume that rotation $\rho(q)$ exists with $\rho(q) = ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ and $w^* = w_i$. Again, we distinguish between two cases:

**Case (i):** $u^* \succ_{w^*} u$**.** This implies that $u^* = u_{i-1}$ or $u_{i-1} \succ^P_{w^*} u^* \succ^P_{w^*} u_i$ by definition of $\rho(q)$. If $u^* = u_{i-1}$, then the uniqueness is guaranteed by Theorem 3.6(iii). If $u_{i-1} \succ^P_{w^*} u^* \succ^P_{w^*} u_i$, then the uniqueness follows from Theorem 3.6(iv).

**Case (ii):** $u \succ_{w^*} u^*$**.** By the definition of $\rho(q)$, we have that $u = u_i$. By Theorem 3.6 (iv), rotation $\rho(q)$ is unique. □

The following result is a centerpiece of the algorithm, specifying exactly which constraints need to be fulfilled by a closed subset of rotations that corresponds to a robust matching:

Lemma 3.20. *Let $P_0$ be a profile and $q = (u^*, w^*, u, w)$ be a stable quadruple of $P_0$. Let $Q = \mathrm{Prof}[\mathrm{swaps}(P_0, q)]$ denote the profile after we perform the swaps in $\mathrm{swaps}(P_0, q)$ on $P_0$. The following holds:*

*(i) If $\pi(q)$ does not exist, then the following holds:*
  *(a) If $w^* \succ^{P_0}_{u^*} w$, then each stable matching $N \in \mathcal{SM}(P_0)$ has $w^* \succ^Q_{u^*} N(u^*)$.*
  *(b) If $w \succ^{P_0}_{u^*} w^*$, then each stable matching $N \in \mathcal{SM}(P_0)$ has either $N(u^*) = w^*$ or $w^* \succ^Q_{u^*} N(u^*)$.*
*(ii) If $\rho(q)$ does not exist, then the following holds:*
  *(a) If $u^* \succ^{P_0}_{w^*} u$, then each stable matching $N \in \mathcal{SM}(P_0)$ has $u^* \succ^Q_{w^*} N(w^*)$.*
  *(b) If $u \succ^{P_0}_{w^*} u^*$, then each stable matching $N \in \mathcal{SM}(P_0)$ has either $N(u^*) = w^*$ or $u^* \succ^Q_{w^*} N(w^*)$.*
*(iii) If neither $\pi(q)$ nor $\rho(q)$ exist, then $\mathcal{SM}(P_0) \cap \mathcal{SM}(Q) = \emptyset$.*

*Let $S$ be a closed subset of rotations for $P_0$ and let $M$ be the corresponding stable matching.*

*(iv) If $\pi(q)$ does not exist and $\rho(q)$ exists but $\rho(q) \notin S$, then $M \notin \mathcal{SM}(Q)$.*
*(v) If $\rho(q)$ exists and $\rho(q) \in S$, then $M \in \mathcal{SM}(Q)$.*
*(vi) If $\pi(q)$ exists and $\pi(q) \notin S$, then $M \in \mathcal{SM}(Q)$.*
*(vii) If $\pi(q)$ exists and $\pi(q) \in S$ and either $\rho(q)$ does not exist or $\rho(q)$ exists but $\rho(q) \notin S$, then $M \notin \mathcal{SM}(Q)$.*

Proof. Let $P_0, q = (u^*, w^*, u, w), Q$ be as defined above. Since $q$ is a stable quadruple, by definition, there exists a stable matching in $P_0$, say $M'$, such that $M'(u^*) = w$ and $M'(u) = w^*$.

**Statement** (i)**.** Assume that $\pi(q)$ does not exist.

First, let us consider the case when $w^* \succ_{u^*}^{P_0} w$. By the definition of $M'$ above, this means that

$$w^* \succ_{u^*}^{P_0} M'(u^*). \tag{6}$$

We first claim that no stable matching in $P_0$ assigns $u^*$ to $w^*$. By Theorem 3.6(i), it is equivalent to claim that the $W$-optimal stable matching does not assign $u^*$ to $w^*$ and no rotation contains $(u^*, w^*)$. First, Property (6) implies that $w^*$ prefers $M'(w^*)$ to $u^*$ in $P_0$ as otherwise $\{u^*, w^*\}$ will be blocking $M'$ in $P_0$. This also means that $\{u^*, w^*\}$ is not in the $W$-optimal stable matching. Second, since $\pi(q)$ does not exist, by Definition 3.17 (Case (a1)), $(u^*, w^*)$ is not in any rotation.

Thus, to show Statement (i), we only need to show that *no* stable matching $N \in \mathcal{SM}(P_0)$ has $N(u^*) \succ_{u^*}^{Q} w^*$. Towards a contradiction, suppose that there exists such a stable matching $N$. By Observation 3.13(ii) and since $w^* \succ_{u^*}^{P_0} w$, it follows that $\succ_{u^*}^{P_0} = \succ_{u^*}^{Q}$, and so $N(u^*) \succ_{u^*}^{P_0} w^*$, implying that $N(u^*) \succ_{u^*}^{P_0} w^* \succ_{u^*}^{P_0} M'(u^*)$ because of Property (6). Thus, there are two stable matchings for $P_0$, where in one of them $u^*$ obtains a partner (namely, $N(u^*)$) who is more preferred than $w^*$, and in the other $u^*$ obtains a partner (namely, $M'(u^*)$) who is less preferred than $w^*$. By Theorem 3.9(i) there is a sequence $S$ of rotations such that starting with the $U$-optimal stable matching for $P_0$ and successively eliminating the rotations in $S$ results in a matching in which $u^*$ obtains the partner $M'(u^*)$. Since $N$ certifies that the partner of $u^*$ in the $U$-optimal matching is strictly preferred by $u^*$ to $w^*$, sequence $S$ includes a rotation that moves $u^*$'s partner from being strictly preferred to $w^*$ to being strictly less desired than $w^*$. This is a contradiction to the fact that $\pi(q)$ does not exist.

Now consider the case when $w \succ_{u^*}^{P_0} w^*$ (Figure 3 Case (a2)). Since $M'(u^*) = w$ belongs to some stable matching and $\pi(q)$ does not exist, we infer that every stable matching $N \in \mathcal{SM}(P_0)$ has either $N(u^*) = w$ or $w \succeq_{u^*}^{P_0} N(u^*)$. By Observation 3.13(ii)(c), it follows that every stable matching $N \in \mathcal{SM}(P_0)$ has either $N(u^*) = w^*$ or $w^* \succ_{u^*}^{Q} N(u^*)$.

**Statement** (ii)**.** Assume that $\rho(q)$ does not exist. Again, we consider two cases, starting with $u^* \succ_{w^*}^{P_0} u$ (see Figure 3 Case (b1)). Since $\rho(q)$ does not exist, we can infer that $\{u^*, w^*\}$ does not belong to any stable matching of $P_0$. Indeed, if a stable matching containing $\{u^*, w^*\}$ existed, then there would be a rotation that changes the partner of $w^*$ from $M'(w^*)$, which is less preferred than $u^*$ to $u^*$ with respect to profile $P_0$.

Thus, to show the desired statement, we only need to show that *no* stable matching $N \in \mathcal{SM}(P_0)$ has $N(w^*) \succ_{w^*}^{Q} u^*$. Towards a contradiction, suppose that there exists such a stable matching $N$ with $N(w^*) \succ_{w^*}^{Q} u^*$. By Observation 3.13(iii) and since $u^* \succ_{w^*}^{P_0} w$, it follows that $\succ_{w^*}^{P_0} = \succ_{w^*}^{Q}$ and so $N(w^*) \succ_{w^*}^{P_0} u^*$, implying that $N(w^*) \succ_{w^*}^{P_0} u^* \succ_{w^*}^{P_0} M'(w^*)$, because $u^* \succ_{w^*}^{P_0} u$ and $M'(w^*) = u$. In other words, there exist two stable matchings where $w^*$ is matched to a partner that is less preferred than $u^*$ and a partner that is more preferred than $u^*$, respectively. However, this is a contradiction to the assumption that $\pi(q)$ does not exist.

Now, let us move to the case when $u \succ_{w^*}^{P_0} u^*$ (see Figure 3 Case (b2)). Recall that $\{u, w^*\}$ belongs to the stable matching $M'$. Since $\rho(q)$ does not exist by Theorem 3.6(i), we infer that $\{u, w^*\}$ in the $W$-optimal stable matching. In other words, for every stable matching $N \in \mathcal{SM}(P_0)$, we have either $N(w^*) = u$ or $u \succ_{w^*}^{P_0} N(w^*)$. By Observation 3.13(iii)(c), it follows that every stable matching $N \in \mathcal{SM}(P_0)$ has either $N(w^*) = u^*$ or $u^* \succ_{w^*}^{Q} N(w^*)$.

**Statement** (iii)**.** Assume that neither $\pi(q)$ nor $\rho(q)$ exists. Again, we distinguish between two cases.

**Case (1):** $w \succ_{u^*}^{P_0} w^*$**.** Since $\pi(q)$ does not exist, by Statement (i), it follows that every stable matching $N \in \mathcal{SM}(P_0)$ has either $N(u^*) = w^*$ or $w^* \succ_{u^*}^{Q} N(u^*)$. Consider an arbitrary

stable matching $N \in \mathcal{SM}(P_0)$, and first let us analyze the case when $N(u^*) = w^*$. Recall that $M'$ was defined as a stable matching of $P_0$ with $M'(u^*) = w$ and $M'(u) = w^*$. By our assumption, $u^*$ prefers $M'$ to $N$ in $P_0$; thus, by Theorem 3.2, we get that $w^*$ must prefer $N$ to $M'$ in $P_0$, i.e., it must hold that $u^* \succ_{w^*}^{P_0} u$. By Statement (ii), that "$u^* \succ_{w^*}^{P_0} u$" and the assumption that "rotation $\rho(q)$ does not exist" imply that $u^* \succ_{w^*}^{Q} N(w^*)$. This contradicts our assumption that $N(u^*) = w^*$. Now, let us move to the second alternative, when $w^* \succ_{u^*}^{Q} N(u^*)$. We know that $\rho(q)$ does not exist. Thus, by Statement (ii), we get that $u^* \succ_{w^*}^{Q} N(w^*)$ because of the following:

- Either $u^* \succ_{w^*}^{P_0} u$, whence by Statement (ii), we have $u^* \succ_{w^*}^{Q} N(w^*)$,
- or $u \succ_{w^*}^{P_0} u^*$ and by Statement (ii), we have that $u^* \succ_{w^*}^{Q} N(w^*)$ or that $N(u^*) = w^*$ (the latter case has just been handled; either way, we have that $u^* \succ_{w^*}^{Q} N(w^*)$.

Yet, this implies that $\{u^*, w^*\}$ is blocking $N$ in $Q$. Thus, $N \notin \mathcal{SM}(Q)$.

**Case (2): $w^* \succ_{u^*}^{P_0} w$.** Now, consider an arbitrary stable matching $N \in \mathcal{SM}(P_0)$. We aim to show that $N$ is not stable in $Q$. Since $\pi(q)$ does not exist, from Statement (i) it follows that

$$w^* \succ_{u^*}^{Q} N(u^*). \tag{7}$$

Thus, to show that $N$ is not stable in $Q$ it suffices to show that $\{u^*, w^*\}$ is blocking $N$ in $Q$, i.e., $w^*$ prefers $u^*$ to $N(w^*)$ in $Q$. If $u^* \succ_{w^*}^{P_0} u$, then, since $\rho(q)$ does not exist, by the first case in Statement (ii), we will obtain that $u^* \succ_{w^*}^{Q} N(w^*)$. Otherwise, by the second case in Statement (ii), it follows that $u^* \succ_{w^*}^{Q} N(w^*)$, because by Property (7), we have that $N(u^*) \neq w^*$.

Summarizing, we have shown that *no* stable matching of $P_0$ is stable for $Q$.

Throughout the remainder of the proof, let $S$ and $M$ be as defined in the lemma.

**Statement (iv).** Assume that $\pi(q)$ does not exist and $\rho(q)$ exists but $\rho(q) \notin S$. Since $\pi(q)$ does not exist, by Statement (i), for every stable matching $N \in \mathcal{SM}(P_0)$ it holds that $N(u^*) = w^*$ or $w^* \succ_{u^*}^{Q} N(u^*)$. This includes $M$, meaning that $M(u^*) = w^*$ or $w^* \succ_{u^*}^{Q} M(u^*)$.

We consider these two cases separately.

**Case (1): $M(u^*) = w^*$.** By Statement (i), it follows that $w \succ_{u^*}^{P_0} w^*$. Recall that $M'$ is a stable matching of $P_0$ with $N(u^*) = w$ and $N(u) = w^*$. In $P_0$, since $u^*$ prefers $M'$ to $M$, by Theorem 3.2, it must be the case that $w^*$ prefers $M$ to $M'$, i.e., $u^* \succ_{w^*}^{P_0} u$. Thus, the rotation $\rho(q)$ (which, by our assumption, is guaranteed to exist) operates as follows: It changes the partner of $w^*$ from an agent that is less preferred than $u^*$ to $u^*$ or to an agent that is more preferred than $u^*$ (regarding $P_0$). Since $\rho(q) \notin S$, we infer that in matching $M$ agent $w^*$ obtains a partner that is less preferred than $u^*$, i.e., $u^* \succ_{w^*}^{P_0} M(w^*)$. This leads to a contradiction with $M(u^*) = w^*$.

**Case (2): $w^* \succ_{u^*}^{Q} M(u^*)$.** Towards a contradiction, suppose that $M$ is also stable for $Q$. This implies that $M(w^*) \succ_{w^*}^{Q} u^*$. By Observation 3.13(iii)(b), we have that $M(w^*) \succ_{w^*}^{P_0} u$. If $u \succ_{w^*}^{P_0} u^*$, then the rotation $\rho(q)$ changes the partner of $w^*$ from $u$ to some agent that is more preferred than $u$ (regarding the preferences in $P_0$). Since, according to $M$, agent $w^*$ already has a partner that is more preferred than $u$, we infer that $\rho(q)$ is the predecessor of some rotation in $S$, meaning that itself $\rho(q) \in S$ by the closedness of $S$—a contradiction. If $u^* \succ_{w^*}^{P_0} u$, then $\succ_{w^*}^{P_0} = \succ_{w^*}^{Q}$ by Observation 3.13(iii), and so we get that $M(w^*) \succ_{w^*}^{P_0} u^* \succ_{w^*}^{P_0} u$, because $M(w^*) \succ_{w^*}^{Q} u^*$. In this case, rotation $\rho(q)$ changes the partner of $w^*$ from an agent that is less preferred than $u^*$ to $u^*$ or an agent who is more preferred than $u^*$. However,

since in $M$ agent $w^*$ already has a partner who is preferred over $u^*$, we again infer that $\rho(q) \in S$—a contradiction.

Summarizing, we conclude that $M \notin \mathcal{SM}(Q)$.

**Statement** (v)**.** Let us assume that $\rho(q)$ exists and $\rho(q) \in S$.

By Lemma 3.14, except $\{u^*, w^*\}$, no other unmatched pair with respect to $M$ could be blocking $M$ in $Q$. In the following, we claim that $\{u^*, w^*\}$ is not blocking $M$ in $Q$, implying that $M$ is stable for $Q$. We distinguish between two cases.

If $u^* \succ_{w^*}^{P_0} u$, then by the definition of $\rho(q)$ (Case (b1)) and since $\rho(q) \in S$, it follows that $M(w^*) = u^*$ or $M(w^*) \succ_{w^*}^{P_0} u^*$. Moreover, by Observation 3.13(iii), we have that $\succ_{w^*}^{Q} = \succ_{w^*}^{Q}$, implying $M(w^*) = u^*$ or $M(w^*) \succ_{w^*}^{Q} u^*$. Thus, $\{u^*, w^*\}$ cannot be blocking $M$ in $Q$.

If $u \succ_{w^*}^{P_0} u^*$, then by the definition of $\rho(q)$ (Case (b2)) and since $\rho(q) \in S$, it follows that $M(w^*) \succ_{w^*}^{P_0} u$. Moreover, by Observation 3.13(iii)(b), we obtain that $M(w^*) \succ_{w^*}^{Q} u^*$. Again, $\{u^*, w^*\}$ cannot be blocking $M$ in $Q$.

**Statement** (vi)**.** Let us assume that $\pi(q)$ exists and $\pi(q) \notin S$. Similarly to Statement (v), by Lemma 3.14, except $\{u^*, w^*\}$, no other unmatched pair with respect to $M$ could be blocking $M$ in $Q$. In the following, we claim that $\{u^*, w^*\}$ is not blocking $M$ in $Q$, which implies that $M$ is stable for $Q$. We distinguish between two cases.

If $w^* \succ_{u^*}^{P_0} w$, then by the definition of $\pi(q)$ (Case (a1)) and since $\pi(q) \notin S$, it follows that $M(u^*) = w^*$ or $M(u^*) \succ_{u^*}^{P_0} w^*$. Moreover, by Observation 3.13(ii), we have that $\succ_{u^*}^{Q} = \succ_{u^*}^{Q}$, implying $M(u^*) = w^*$ or $M(u^*) \succ_{u^*}^{Q} w^*$. Thus, $\{u^*, w^*\}$ cannot be blocking $M$ in $Q$.

If $w \succ_{u^*}^{P_0} w^*$, then by the definition of $\pi(q)$ (Case (a2)) and since $\pi(q) \notin S$, it follows that $M(u^*) \succ_{u^*}^{P_0} w$. Moreover, by Observation 3.13(ii)(b) , we obtain that $M(u^*) \succ_{u^*}^{Q} w^*$. Again, $\{u^*, w^*\}$ cannot be blocking $M$ in $Q$.

**Statement** (vii)**.** Assume that $\pi(q)$ exists and $\pi(q) \in S$ and either $\rho(q)$ does not exist or it exists but $\rho(q) \notin S$.

Suppose, for the sake of contradiction, that $M$ is stable for $Q$. We distinguish between three cases, in each case obtaining a contradiction.

**Case 1: $w^* \succ_{u^*}^{P_0} w$.** By the definition of $\pi(q)$ and since $\pi(q) \in S$, referencing Theorem 3.9(i), it follows that $w^* \succ_{u^*}^{P_0} M(u^*)$. Thus, $w^* \succ_{u^*}^{Q} M(u^*)$, because $\succ_{u^*}^{P_0} = \succ_{u^*}^{Q}$ (by Observation 3.13(ii)). In particular, this implies that $\{u^*, w^*\}$ is an unmatched pair in $M$. By assumption that $M$ is stable for $Q$, we must have that

$$M(w^*) \succ_{w^*}^{Q} u^*. \tag{8}$$

If $u^* \succ_{w^*}^{P_0} u$, then by Observation 3.13(iii), we have $M(w^*) \succ_{w^*}^{P_0} u^* \succ_{w^*}^{P_0} u$. Recall that $M'$ is stable for $P_0$ with $M'(w^*) = u$. This means that there are two stable matchings, where $w^*$ obtains a partner (namely, $u = M'(w^*)$) who is less preferred than $u^*$, and a partner (namely, $M(w^*)$) who is more preferred than $u^*$. This implies that $\rho(q)$ exists and that $\rho(q) \in S$—a contradiction.

If $u \succ_{w^*}^{P_0} u^*$, then by Observation 3.13(iii) and by (8), we have that $M(w^*) \succ_{w^*}^{P_0} u \succ_{w^*}^{P_0} \succ u^*$. Again, since $M'$ is stable for $P_0$ with $M'(w^*) = u$ there are two stable matchings, where $w^*$ obtains partner $u = M'(w^*)$, and a partner (namely, $M(w^*)$) who is more preferred than $u$. This implies that $\rho(q)$ exists such that $\rho(q) \in S$—a contradiction.

**Case 2: $w \succ_{u^*}^{P_0} w^*$ and $M(u^*) \neq w^*$.** If we can show that $w^* \succ_{u^*}^{Q} M(u^*)$, then we can use the same reasoning as we did for the first case to show the same contradiction. Thus, it suffices to prove that $w^* \succ_{u^*}^{Q} M(u^*)$.

By the definition of $\pi(q)$ and since $\pi(q) \in S$, referencing Theorem 3.9(i), it follows that $M(u^*) = w$ or $w \succ^{P_0}_{u^*} M(u^*)$, and thus $w^* \succ^{Q}_{u^*} M(u^*)$, because $M(u^*) \neq w^*$ (by assumption in this case) and $w^* \succ^{Q}_{u^*} w$. This finishes the proof for the second case.

**Case 3: $w \succ^{P_0}_{u^*} w^*$ and $M(u^*) = w^*$.** Recall that $M'$ is also a stable matching for $P_0$ with $M'(u^*) = w$. Thus, $P_0$ admits two different stable matchings $M$ and $M'$, where $M(u^*) = w^*$, $M'(u^*) = w$, and $M'(w^*) = u$. By the precondition that $w \succ^{P_0}_{u^*} w^*$ and by Theorem 3.2, we must have that $u^* \succ^{P_0}_{w^*} u$ (i.e., $w^*$ prefers $M$ to $M'$ in $P_0$). In particular, this means that there must be a rotation that changes the partner of $w^*$ from $u = M'(w^*)$, which is less preferred than $u^*$, to agent $u^*$. Thus, $\rho(q)$ exists and must be in $S$—a contradiction. □

### 3.4 Polynomial-time Algorithms for Robust Matchings

In this section, we first present an $O(n^4)$-time algorithm for finding a robust matching if it exists (see Algorithm 1 and Example 3.23). Then, we use a **Linear Programming (LP)** formulation to show that perfect robust matchings and robust matchings with minimum egalitarian cost can be found in polynomial time if they exist. Both approaches crucially rely on

(i) the one-to-one correspondence between the stable matchings and the closed subsets of the rotation poset [26, Section 3.7],
(ii) the implications between the presence of the two rotations $\pi(q)$ and $\rho(q)$ of stable quadruples $q$ derived from Lemma 3.20, and
(iii) the fact that all stable quadruples can be computed in $O(n^4)$ time.

The proof for (iii) is roughly based on iterating over all possible rotations and building a lookup table that stores for all pairs $(x, y) \in U \times W$ of agents a constant number of rotations that make the partner of $x$ less preferred to $y$ or more preferred to $y$. Then, for each stable quadruple $q$, we can look up its corresponding rotations $\pi(q)$ and $\rho(q)$ in the table. We state this observation for reference below.

Proposition 3.21. *Determining all stable quadruples $q$ and their respective rotations $\pi(q)$ and $\rho(q)$ as defined in Definition 3.17 can be done in $O(n^4)$ time.*

Proof. By Theorem 3.1 and by Theorem 3.9(ii), all $O(n^4)$ stable quadruples can be found in $O(n^4)$ time; also see Definition 3.10 for the definition of stable quadruples.

For each stable quadruple $q$, we show how to determine whether $\pi(q)$ and $\rho(q)$ exist and find them if they exist, in $O(1)$ time for a given stable quadruple $q$. We build in $O(n^4)$ time a size-$O(n^2)$ lookup table $T$ to store for each ordered pair $(x, y) \in U \times W$ the following up to six rotations:

(1) Let $\sigma_1(x, y)$ denote the rotation that changes the partner of $x$ from someone who is more preferred than $y$ to $y$.
Formally, $\sigma_1(x, y) := ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ such that $x = u_i$ and $y = w_{i+1}$ for some $i \in \{0, \ldots, r-1\}$. Note that the uniqueness of this rotation is guaranteed by Theorem 3.6(iii).
(2) Let $\sigma_2(x, y)$ denote the rotation that changes the partner of $x$ from someone who is more preferred than $y$ to someone who is less preferred than $y$.
Formally, $\sigma_2(x, y) := ((u_0, w_0), \ldots, (u_{r-1}, w_{r-1}))$ such that $x = u_i$ and $w_i \succ_x y \succ_x w_{i+1}$ for some $i \in \{0, \ldots, r-1\}$. Note that the uniqueness of this rotation is guaranteed by Theorem 3.6(ii).
(3) Let $\sigma_3(x, y)$ denote the rotation that changes the partner of $x$ from $y$ to someone who is less preferred than $y$.

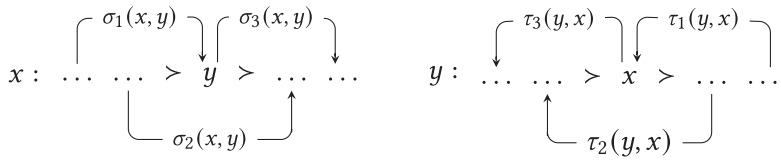



Fig. 4. The six rotations defined in the proof of Proposition 3.21.

Formally, $\sigma_3(x,y) := ((u_0,w_0),\ldots,(u_{r-1},w_{r-1}))$ such that $x = u_i$ and $y = w_i$ for some $i \in \{0,\ldots,r-1\}$. Note that the uniqueness of this rotation is guaranteed by Theorem 3.6(ii). Moreover, the existence of $\sigma_2(x,y)$ precludes the existence of $\sigma_1(x,y)$ and $\sigma_3(x,y)$, because $\sigma_2(x,y)$ implies that $\{x,y\}$ is not in any stable matching while $\sigma_1(x,y)$ or $\sigma_3(x,y)$ implies that $\{x,y\}$ is some stable matching.

(4) Let $\tau_1(y,x)$ denote the rotation that changes the partner of $y$ from someone who is less preferred than $x$ to $x$.

Formally, $\tau_1(y,x) := ((u_0,w_0),\ldots,(u_{r-1},w_{r-1}))$ such that $y = w_i$ and $x = u_{i-1}$ for some $i \in \{0,\ldots,r-1\}$. Note that the uniqueness of this rotation is guaranteed by Theorem 3.6(iii).

(5) Let $\tau_2(y,x)$ denote the rotation that changes the partner of $y$ from someone who is less preferred than $x$ to someone who is more preferred than $x$.

Formally, $\tau_2(y,x) := ((u_0,w_0),\ldots,(u_{r-1},w_{r-1}))$ such that $y = w_i$ and $u_{i-1} \succ_y x \succ_y u_i$ for some $i \in \{0,\ldots,r-1\}$. Note that the uniqueness of this rotation is guaranteed by Theorem 3.6(iv).

(6) Let $\tau_3(y,x)$ denote the rotation that changes the partner of $y$ from $x$ to someone who is more preferred than $x$.

Formally, $\tau_3(y,x) := ((u_0,w_0),\ldots,(u_{r-1},w_{r-1}))$ such that $y = w_i$ and $x = u_i$ for some $i \in \{0,\ldots,r-1\}$. Note that the uniqueness of this rotation is guaranteed by Theorem 3.6(iv). Moreover, the existence of $\tau_2(y,x)$ precludes the existence of $\tau_1(y,x)$ and $\tau_3(y,x)$, because $\tau_2(y,x)$ implies that $\{x,y\}$ is not in any stable matching while $\tau_1(y,x)$ or $\tau_3(y,x)$ implies that $\{x,y\}$ is some stable matching.

Figure 4 illustrates the six rotations we have just defined.

Now, we continue with the determination of $\rho(q)$ and $\pi(q)$. Let $q = (u^*,w^*,u,w)$.

$$\pi(q) := \begin{cases} \sigma_3(u^*,w^*), & \text{if } w^* \succ_{u^*} w \text{ and } \sigma_3(u^*,w^*) \text{ exists,} \\ \sigma_2(u^*,w^*), & \text{if } w^* \succ_{u^*} w \text{ and } \sigma_2(u^*,w^*) \text{ exists,} \\ \sigma_1(u^*,w), & \text{if } w \succ_{u^*} w^* \text{ and } \sigma_1(u^*,w) \text{ exists,} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

$$\rho(q) := \begin{cases} \tau_1(w^*,u^*), & \text{if } u^* \succ_{w^*} u \text{ and } \tau_1(w^*,u^*) \text{ exists,} \\ \tau_2(w^*,u^*), & \text{if } u^* \succ_{w^*} u \text{ and } \tau_2(w^*,u^*) \text{ exists,} \\ \tau_3(w^*,u), & \text{if } u \succ_{u^*} u^* \text{ and } \tau_3(w^*,u) \text{ exists,} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

One can verify that the above construction corresponds to Definition 3.17. Since there are $O(n^2)$ rotations and $n^2$ ordered pairs, the whole table, containing $O(n^2)$ entries, can be determined in $O(n^4)$ time. (Note that, for a given rotation $\rho$, we can first find the agents who are affected by the rotation in $O(n)$ time, and, for each of the affected agents $z$, find in $O(n)$ time all the pairs $(z,w)$ such that $\rho$ needs to be added to the table entry of $(z,w)$.) After computing the table, we can determine in constant time the two rotations $\pi(q)$ and $\rho(q)$ for each $q$ from the $O(n^4)$ stable quadruples by looking up into the table. In total, the running time is $O(n^4)$. □

**ALGORITHM 1:** Computing $d$-robust matchings.

**Input**: A preference profile $P$ with agent sets $U$ and $W$, and an integer $d \in \mathbb{N}$.
**Output**: A $d$-robust matching for $P$ or $\perp$ if none exists.

1 Compute the rotation digraph $G(P)$.
2 $G_1(P) \leftarrow G(P)$
3 $D \leftarrow \emptyset$
4 $A \leftarrow \emptyset$
5 **foreach** *stable quadruple* $q$ *with* $|\text{swaps}(P, q)| \leq d$ **do**
6 Compute $\pi(q)$ and $\rho(q)$ if they exist using Prop. 3.21.
7 **if** $\nexists \pi(q)$ *and* $\nexists \rho(q)$ **then return** $\perp$
8 **if** $\exists \pi(q)$ *and* $\exists \rho(q)$ **then** $G_1(P) \leftarrow G_1(P) + (\rho(q), \pi(q))$
9 **if** $\exists \pi(q)$ *and* $\nexists \rho(q)\}$ **then** $D \leftarrow D \cup \{\pi(q)\}$
10 **if** $\nexists \pi(q)$ *and* $\exists \rho(q)$ **then** $A \leftarrow A \cup \{\rho(q)\}$
11 $G_2(P) \leftarrow G_1(P) - D - \{v \in V(G_1(P)) \mid \exists$ a dipath in $G_1(P)$ from a vertex in $D$ to $v\}$
12 **if** $A \not\subseteq V(G_2(P))$ **then return** $\perp$
13 $T \leftarrow A \cup \{v \in V(G_2(P)) \mid \exists$ a dipath in $G_2(P)$ from $v$ to some vertex in $A\}$
`// Set T now corresponds to a closed set of rotations (see Theorem 3.22).`
14 **return** the matching corresponding to the closed set $T$ of rotations

Now, we prove our main result for the Robust Matching problem by providing a polynomial-time algorithm. A pseudo-code description is given in Algorithm 1. Intuitively, it works as follows: Let $P$ be the profile in the given instance. We work with the rotation digraph $G(P)$ (see Definition 3.7 and Theorem 3.9). Recall that each stable matching for $P$ corresponds to a closed subset of the rotations in the rotation poset, i.e., a vertex subset $S$ of $G(P)$ such that no vertex not in $S$ has a sink in $S$. The aim is to find a closed subset of rotations in $G(P)$ that corresponds to a $d$-robust matching. In Section 3.2, we have seen that it is enough to avoid the obstructions given by stable quadruples that have swap sets with at most $d$ swaps (Lemma 3.16). Lemma 3.20 then showed how to avoid these obstructions. This lemma characterizes the constraints that a closed set of rotations needs to satisfy to be $d$-robust. Algorithm 1 now incorporates some of these constraints into $G(P)$ by adding additional arcs, which means that closed subsets in $G(P)$ are more constrained. Then, the algorithm checks for a closed subset of rotations in the resulting graph that satisfies all remaining constraints, such as that certain rotations need to be contained in the subset of rotations.

More precisely, the algorithm proceeds as follows: Algorithm 1 first adds arcs to $G(P)$ in lines 2 to 10 that model implications between rotations contained in $d$-robust matchings given in Lemma 3.20. Then, it removes rotations from $G(P)$ that cannot occur in $d$-robust matchings according to Lemma 3.20 (lines 9 to 11). Finally, in line 13 it checks whether there is a closed subset of rotations that contains the required rotations for $d$-robust matchings as specified by Lemma 3.20.

Theorem 3.22. *Given an instance of* Robust Matching *with* $2n$ *agents, in* $O(n^4)$ *time, we can either find a* $d$*-robust matching or correctly report that no such matching exists.*

Proof. As mentioned, a pseudocode description is given in Algorithm 1. We need to prove the correctness and then analyze the running time. Let $P$ be the preference profile of the given instance. Below, when referring to $G_1(P)$, we mean the graph $G_1(P)$ obtained from $G(P)$ after line 10 and by $G_2(P)$, we mean the graph obtained after line 11. Call a vertex subset $S$ in a directed graph $G$ *closed* if there is no arc $(x, y)$ in $G$ such that $x \notin S$ and $y \in S$.

We claim that if Algorithm 1 returns something different from $\perp$, then it is a $d$-robust matching. We first show that the set $T$ computed in line 13 is a closed subset of rotations in the rotation

poset: Clearly, $T$ is closed in $G_2(P)$. Since $G_2(P)$ is obtained from $G_1(P)$ by removing vertices together with all of their successors, $T$ is closed in $G_1(P)$ as well. Since $G_1(P)$ is obtained from $G(P)$ by adding arcs, $T$ is closed in $G(P)$, implying the claim. By Theorem 3.9, there is a stable matching $M$ for $P$ associated with $T$.

Since $T$ is a closed subset, Lemma 3.20 (iv) to (vii) apply. We now verify that, for each stable quadruple $q$ with $|\text{swaps}(P,q)| \leq d$, we have $M \in \mathcal{SM}(\text{Prof}[\text{swaps}(P,q)])$ (recall that $\text{Prof}[\text{swaps}(P,q)]$ is the profile obtained from $P$ by performing the swaps in $\text{swaps}(P,q)$). By Lemma 3.16 it then follows that $M$ is $d$-robust.

Let $q$ be a stable quadruple of $P$ such that $|\text{swaps}(P,q)| \leq d$. Assume that $\rho(q)$ exists. If $\pi(q)$ does not exist, then $\rho(q) \in T$ by line 10 and line 13, and thus $M \in \mathcal{SM}(\text{Prof}[\text{swaps}(P,q)])$ by Lemma 3.20 (v). If $\pi(q)$ exists, then, since $T$ is closed and by line 8, either $\pi(q) \notin T$, giving $M \in \mathcal{SM}(\text{Prof}[\text{swaps}(P,q)])$ by Lemma 3.20 (vi), or $\rho(q) \in T$, giving $M \in \mathcal{SM}(\text{Prof}[\text{swaps}(P,q)])$ by Lemma 3.20 (v).

Now assume that $\rho(q)$ does not exist. Then, $\pi(q)$ exists, because otherwise, we would have returned $\perp$ in line 7. By line 9, we then have $\pi(q) \in D$ and by line 11 $\pi(q) \notin V(G_2(P))$. Thus, $\pi(q) \notin T$ by construction of $T$ on line 13. This gives $M \in \mathcal{SM}(\text{Prof}[\text{swaps}(P,q)])$ by Lemma 3.20 (vi). Thus, indeed the returned matching is $d$-robust.

It remains to show that a $d$-robust matching is returned if there is a $d$-robust matching $M$ for $P$. By the above, it suffices to show that $\perp$ is not returned in lines 7 and 12. By Lemma 3.16, $M \in \mathcal{SM}(\text{Prof}[\text{swaps}(P,q)])$ for each stable quadruple $q$ with $|\text{swaps}(P,q)| \leq d$. Thus, by Lemma 3.20 (iii) at least one of $\rho(q)$ and $\pi(q)$ exists, meaning that $\perp$ cannot be returned in line 7. If $\perp$ was returned due to line 12, then there is a stable quadruple $q$ with $|\text{swaps}(P,q)| \leq d$ such that $\pi(q)$ does not exist and $\rho(q)$ exists and, furthermore, $\rho(q) \in V(G_1(P)) \setminus V(G_2(P))$. Let $S$ be the closed subset of rotations in $G(P)$ associated with $M$. By Lemma 3.20 (iv), $\rho(q) \in S$. Since $\rho(q) \notin V(G_2(P))$, by lines 9 and 11, there is a stable quadruple $q'$ with $|\text{swaps}(P,q')| \leq d$ such that $\pi(q') \in D$ and there is a path (possibly of length zero) from $\pi(q')$ to $\rho(q)$ in $G_1(P)$. Since $\rho(q) \in S$, thus also $\pi(q') \in S$. Since $\pi(q') \in D$, by line 9, $\rho(q')$ does not exist. Thus, by Lemma 3.20 (vii) $M \notin \mathcal{SM}(\text{Prof}[\text{swaps}(P,q')])$, a contradiction to $M$ being $d$-robust. Thus, indeed a $d$-robust matching is returned if there is one.

The running time of $O(n^4)$ can be obtained as follows: By Theorem 3.9, the rotation digraph in line 1 can be computed in $O(n^2)$ time. Lines 5 and 6 can be carried out in $O(n^4)$ time by Proposition 3.21. Thus, clearly, lines 2 to 10 can be carried out in $O(n^4)$ time. Lines 9–11 can be done in $O(n^4)$ time, because $G(P)$ contains $O(n^2)$ vertices. Analogously, lines 10–14 take $O(n^4)$ time. □

*Example 3.23.* To illustrate Algorithm 1, consider the profile $P$ given in Example 1.1 and let $d = 1$. $P$ admits three rotations, $\pi_1 = ((u_1, w_2), (u_2, w_3), (u_3, w_4), (u_4, w_1))$, $\pi_2 = ((u_1, w_3), (u_3, w_1))$, and $\pi_3 = ((u_2, w_4), (u_4, w_2))$; see also Figure 2 for the rotations and the matchings they are exposed in. There are 10 stable quadruples for $d = 1$. The corresponding $\pi(q)$ and $\rho(q)$ are summarized in Table 2.

The digraphs $G(P)$ and $G_1(P) = G_2(P)$ constructed in Algorithm 1 are depicted in Figure 5. One can verify that after carrying out lines 1 to 10, we have $A = \{\pi_1\}$ (see rows 4, 8, 9 in the table) and $D = \emptyset$. Set $T = \{\pi_1, \pi_2, \pi_3\}$ is the only closed set in $G_2$ that includes $\pi_1$, which corresponds to $M_2$. Indeed, our algorithm will return $M_2$ (see Example 1.1) as the only 1-robust matching. ◇

Since all stable matchings match the same set of agents, to check whether there is a perfect $d$-robust matching it suffices to check whether the $U$-optimal stable matching is perfect. Thus, we obtain the following:

Corollary 3.24. *Given an instance of Robust Matching with $2n$ agents, in $O(n^4)$ time we can either find a $d$-robust perfect matching or correctly report that no such matching exists.*

Table 2. Summary of the Stable Quadruples and Their Two Specific Rotations for Example 3.23

| Stable quadruple $q$ with $\lvert\text{swaps}(P,q)\rvert \leq 1$ | $\pi(q)$ | $\rho(q)$ |
|---|---|---|
| $(u_1, w_1, u_3, w_3)$ | $\pi_1$ | $\pi_2$ |
| $(u_1, w_2, u_4, w_1)$ | $\pi_1$ | $\pi_3$ |
| $(u_1, w_2, u_4, w_3)$ | $\pi_1$ | $\pi_3$ |
| $(u_1, w_3, u_2, w_2)$ | no | $\pi_1$ |
| $(u_1, w_4, u_2, w_1)$ | $\pi_2$ | $\pi_3$ |
| $(u_2, w_1, u_3, w_2)$ | $\pi_3$ | $\pi_2$ |
| $(u_2, w_2, u_4, w_4)$ | $\pi_1$ | $\pi_3$ |
| $(u_2, w_4, u_3, w_3)$ | no | $\pi_1$ |
| $(u_3, w_1, u_4, w_4)$ | no | $\pi_1$ |
| $(u_3, w_2, u_4, w_3)$ | $\pi_2$ | $\pi_3$ |


Fig. 5. The two digraphs discussed in Example 3.23.

Now, we turn to the problem variant where we look for a $d$-robust matching with minimum egalitarian cost. Our polynomial-time algorithm for this variant builds on a **Linear Programming (LP)** formulation that finds a stable matching. This LP formulation in turn is based on the one-to-one correspondence between the stable matchings and the closed subsets of the rotation poset [26, Section 3.7]. A crucial property of this formulation is that its constraint matrix is totally unimodular. Hence, each extreme point of the polytope defined by this formulation is integral.

The LP formulation is as follows: Let $P_0$ be a preference profile with two disjoint sets, $U$ and $W$, each containing $n$ agents. Let $R(P_0)$ be the set of rotations for $P_0$ and let $G(P_0)$ with arc set $E(P_0)$ be the rotation digraph of $P_0$ representing the precedence relation $\lhd^{P_0}$; by Theorem 3.9(ii), both the rotation set $R(P_0)$ and the rotation digraph $G(P_0)$ can be computed in $O(n^2)$ time. For each rotation $\rho \in R(P_0)$, we introduce a variable $x_\rho$ with box constraints $0 \leq x_\rho \leq 1$, where $x_\rho = 1$ will correspond to adding $\rho$ to the solution subset, while $x_\rho = 0$ means that $\rho$ will not be taken into the subset. By Gusfield and Irving [26, Section 3.7], the constraint matrix of the constraints

$$x_\rho - x_\pi \leq 0, \qquad \forall \pi, \rho \in R(P_0) \text{ with } (\pi, \rho) \in E(P_0), \tag{LP1}$$

$$0 \leq x_\rho \leq 1, \qquad \forall \rho \in R(P_0), \tag{LP2}$$

is totally unimodular, and thus there is a solution in which each variable takes either value zero or one. In this way, the set $S = \{\rho \mid x_\rho = 1\}$, defined by including exactly those rotations whose variable values are set to one, is closed under the rotation poset and thus defines a stable matching.

Before we state our main result for the Robust Matching problem, we recall a condition that ensures that an LP formulation gives an integral solution.

Proposition 3.25 ([9]). *If $A \in \{-1, 0, +1\}^{\hat{n} \times \hat{m}}$ and $b \in \mathbb{Z}^{\hat{n}}$ such that each row in $A$ has at most one $+1$ and at most one $-1$, then $A$ is totally unimodular, and every extreme point of the system $Ax \leq b$, $x \in \mathbb{N}_0^{\hat{n}}$ is integral.*

Theorem 3.26. *Determining whether a $d$-robust matching exists and finding one with minimum egalitarian cost if it exists can be done in polynomial time.*

Proof. Following Lemma 3.20, we will add some additional constraints to the LP given by (LP1) and (LP2), which results in an LP whose constraint matrix remains totally unimodular (see Proposition 3.25). First, similarly to the proof of Theorem 3.22, we compute for each stable quadruple $q$ with $|\text{swaps}(P_0, q)| \leq d$ the two specific rotations $\pi(q)$ and $\rho(q)$ as defined in Definition 3.17.

As already discussed, $\pi(q)$ and $\rho(q)$ may not exist. If they exist, then by Lemma 3.19 they are unique. Moreover, by Lemma 3.20(iii), we may assume that at least one of $\pi(q)$ and $\rho(q)$ exist, as otherwise $\mathcal{SM}(P) \cap \mathcal{SM}(\text{Prof}[\text{swaps}(P, q)]) = \emptyset$, implying that $P$ does not admit a $d$-robust matching. We distinguish between three cases, in each case describing how to add some constraints to the LP defined above.

**Case (1): Both $\pi(q)$ and $\rho(q)$ exist.** Add the constraint

$$x_{\pi(q)} - x_{\rho(q)} \leq 0. \tag{LP3.1}$$

By Lemma 3.20, Statements (v), (vi), and (vii), the stable matching defined according to a closed subset is stable in $\text{Prof}[\text{swaps}(P_0, q)]$ if and only if $x_{\rho(q)} = 1$ or $x_{\pi(q)} = 0$.

**Case (2): $\pi(q)$ exists but $\rho(q)$ does not.** Add the constraint

$$x_{\pi(q)} = 0. \tag{LP3.2}$$

The above constraint is justified by Lemma 3.20(vii).

**Case (3): $\pi(q)$ does not exist but $\rho(q)$ exists.** Add the constraint

$$x_{\rho(q)} = 1. \tag{LP3.3}$$

This constraint is justified by Lemma 3.20, Statements (iv) and (v).

Note that in each of the three cases, we add to the constraint matrix a row that has at most one $+1$, at most one $-1$, and the remaining values are all 0s. Thus, we can infer by Proposition 3.25 that the resulting constraint matrix is still totally unimodular and all primal solutions to our problem are integral. Since the matrix has $O(n^4)$ rows and $O(n^2)$ columns, solving the thus constructed LP can be done in polynomial time.

Finally, observe that finding a $d$-robust matching, if one exists, with minimum egalitarian cost can be done in polynomial time by the following: For each rotation $\rho$, we can compute how adding $\rho$ to a stable matching changes its egalitarian score (see Gusfield and Irving [26, p. 128, bottom] for details) Then, it is sufficient to add an appropriate optimization objective to the LP constructed above. □

## 4 ROBUSTNESS AND PREFERENCES WITH TIES: NP-HARDNESS

In this section, we consider the case when the preference list $\succeq$ of each agent is a weak order (i.e., transitive and complete, but not necessarily asymmetric binary relation) on the set of the agents who she finds acceptable. For instance, the following binary relation $\succeq_i$ with

$$\succeq_i = \{(1, 1), (2, 2), (3, 3), (1, 2), (2, 1), (1, 3), (2, 3)\} \tag{9}$$

is a weak order on $\{1, 2, 3\}$. The expression "$x \succeq_i y$" means that $i$ weakly prefers $x$ over $y$ (i.e., $x$ is better or as good as $y$). We use $\succ_i$ to denote the asymmetric part (i.e., $x \succeq_i y$ and $\neg(y \succeq_i x)$) and $\sim_i$ to denote the symmetric part of $\succeq_i$ (i.e., $x \succeq_i y$ and $y \succeq_i x$). We say that two agents $x, y$ are *tied* in

the preference list $\succeq_i$ if $x \sim_i y$. In this case, $x$ and $y$ form a *tie* and the preference list $\succeq_i$ (and any preference profile it is contained in) contains *ties*. Since a preference list $\succeq_i$ can be decomposed into an asymmetric part $\succ_i$ and a symmetric part $\sim_i$, in the following when we illustrate a preference list, we only describe the $\succ_i$ part and the "relevant" $\sim_i$ part. For instance, the preference list $\succeq_i$ as described in Equation (9) will be depicted as follows:

$$\succeq_i = (\{1, 2\}) \succ_i 3.$$

For preference lists with ties, we need to adapt the notion of swap distances from Section 2.1.

*Definition 4.1 (Swap distances for preferences with ties).* Let $\succeq_1$ and $\succeq_2$ be two preference lists on (subsets of) of an agent set $V$, possibly with ties. The *swap distance* $\tau(\succeq_i, \succeq_j)$ between $\succeq_i$ and $\succeq_j$ is defined as follows:

$$\mathcal{D}(\succeq_i, \succeq_j) \coloneqq \{\{x, y\} \subseteq V \mid x \succ_i y \wedge y \succ_j x\},$$
$$\mathcal{T}(\succeq_i, \succeq_j) \coloneqq \{\{x, y\} \subseteq V \mid x \sim_i y \wedge (x, y) \notin \sim_j\},$$

$$\tau(\succeq_i, \succeq_j) \coloneqq \begin{cases} \infty, & \text{if } \succeq_i \text{ and } \succeq_j \text{ have } \textit{different} \text{ acceptable sets,} \\ |\mathcal{D}(\succeq_i, \succeq_j)| + |\mathcal{T}(\succeq_i, \succeq_j)| + |\mathcal{T}(\succeq_j, \succeq_i)|, & \text{otherwise.} \end{cases}$$

For two preference profiles $P_1$ and $P_2$ with $P_1 = ((\succeq_u^{P_1})_{u \in U}, (\succeq_w^{P_1})_{w \in W})$ and $P_2 = ((\succeq_u^{P_2})_{u \in U}, (\succeq_w^{P_2})_{w \in W})$, the *swap distance between $P_1$ and $P_2$* is defined as follows:

$$\tau(P_1, P_2) \coloneqq \sum_{x \in U \cup W} \tau\left(\succeq_x^{P_1}, \succeq_x^{P_2}\right).$$

For technical reasons, for two distinct agents $x$ and $y$ (with $x$ possibly being a bottom agent), we define the *swap distance from $y$ to $x$ in $\succeq$* as the number of agents that are ranked between $x$ and $y$:

$$\tau(x, y, \succeq) \coloneqq \begin{cases} |\{z \in V \setminus \{y\} \mid x \succeq z \succeq y\}|, & \text{if } x \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases}$$

Briefly put, the swap distance between $\succeq_i$ and $\succeq_j$ is the smallest Kendall $\tau$ distance between all possible linear extensions of $\succeq_i$ and of $\succeq_j$. The swap distance from $y$ to $x$ denotes the minimum number of pairs required to make $y$ (strictly) preferred to $x$ in $\succeq$. Naturally, if $y \succ x$ or $x$ is a bottom agent, then $\tau(x, y, \succeq) = 0$.

*Example 4.2.* Consider the following two preference lists $\succeq_1$ and $\succeq_2$ with

$$e \succ_1 (\{a, b, c\}) \succ_1 d \text{ and } e \succ_2 d \succ_2 c \succ_2 (\{a, b\}).$$

In words, in preference list $\succeq_1$, agent $e$ is ranked first, all three agents from $\{a, b, c\}$ are tied at the second position, and $d$ is ranked last. The swap distance between $\succeq_1$ and $\succeq_2$ is five due to the following five pairs:

(i) $\mathcal{D}(\succeq_1, \succeq_2) = \{\{a, d\}, \{b, d\}, \{c, d\}\}$,
(ii) $\mathcal{T}(\succeq_1, \succeq_2) = \{\{a, c\}, \{b, c\}\}$, and $\mathcal{T}(\succeq_2, \succeq_1) = \emptyset$.

Alternatively, using the notion of Kendall $\tau$ distance, let $\succ_1^*$ and $\succ_2^*$ denote the linear extensions of $\succeq_1$ and $\succeq_2$, respectively. Then, we can verify that the smallest number of pairs differently ordered between $\succ_1^*$ and $\succ_2^*$ is five, by defining $\succ_1^* \coloneqq e \succ a \succ b \succ c \succ d$ and $\succ_2^* \coloneqq e \succ d \succ c \succ a \succ b$.

In $\succeq_1$, the swap distance from $a$ to $c$ (or from $c$ to $a$) is two, while the swap distance from $d$ to $c$ is three but the swap distance from $c$ to $d$ is zero, i.e., $\tau(c, a, \succeq_1) = \tau(a, c, \succeq_1) = 2$, $\tau(c, d, \succeq_1) = 3$, and $\tau(d, c, \succeq_1) = 0$. ◇

We observe the following for the notion of swap distances:

Lemma 4.3. *Let $\succeq$ and $\succeq'$ be two preference lists on the same acceptable set $V$. The following hold:*
*(1) $\mathcal{D}(\succeq, \succeq')$, $\mathcal{T}(\succeq, \succeq')$, and $\mathcal{T}(\succeq', \succeq)$ are pairwisely disjoint.*
*(2) $\tau(\succeq, \succeq') = \tau(\succeq', \succeq)$.*
*(3) For each two distinct agents $x, y \in V$ with $y \succ' x$, it holds that $\tau(\succeq, \succeq') \geq \tau(x, y, \succeq)$.*

Proof. Let $V, \succeq, \succeq'$ be as defined. For notational convenience, let $A := \mathcal{D}(\succeq, \succeq')$, $B := \mathcal{T}(\succeq, \succeq')$, and $C := \mathcal{T}(\succeq', \succeq)$.

For Statement (1), let us consider an arbitrary pair $\{x, y\} \subseteq V$ with $x \neq y$. Since $\succeq$ and $\succeq'$ are two weak orders on $V$ and since $\{x, y\}$ is an unordered pair, there are nine cases for $\{x, y\}$:

(i) $x \succ y$ **and** $x \succ' y$**.** Then, $\{x, y\}$ belongs to neither $A$, nor $B$, nor $C$.
(ii) $y \succ x$ **and** $y \succ' x$**.** This case is the same as Case (i).
(iii) $x \sim y$ **and** $x \sim' y$**.** This case is the same as Case (i).
(iv) $x \succ y$ **and** $y \succ' x$**.** Then, $\{x, y\} \in A$, but $\{x, y\} \notin B$ and $\{x, y\} \notin C$.
(v) $y \succ x$ **and** $x \succ' y$**.** This case is the same as Case (iv).
(vi) $x \succ y$ **and** $x \sim' y$**.** Then, $\{x, y\} \notin A$, $\{x, y\} \notin B$, but $\{x, y\} \in C$.
(vii) $y \succ x$ **and** $x \sim' y$**.** This case is the same as Case (vi).
(viii) $x \sim y$ **and** $x \succ' y$**.** Then, $\{x, y\} \notin A$, $\{x, y\} \in B$, but $\{x, y\} \notin C$.
(ix) $x \sim y$ **and** $y \succ' x$**.** This case is the same as Case (viii).

Altogether, we showed that $\{x, y\}$ belongs to at most one set from $A, B, C$. Hence, the three sets $A, B, C$ are pairwisely disjoint.

Statement (2) follows from Statement (1) and the fact that $\mathcal{D}(\succeq, \succeq') = \mathcal{D}(\succeq', \succeq)$.

For Statement (3), assume that $y \succ' x$. We distinguish between two cases. If $y \succ x$, then $\tau(x, y, \succeq) = 0$. We immediately have that $\tau(\succeq, \succeq') \geq \tau(x, y, \succeq)$, since $\tau(\succeq, \succeq') \geq 0$.

It remains to consider the case when $x \succ y$. Consider an arbitrary agent $z \in V \setminus \{y\}$ with $x \succeq z \succeq y$ and $z \neq y$. By the transitivity of $\succeq'$, if $z \succeq' y$, then $z \succ' x$, and hence $\{x, z\} \in A \cup B$. If $y \succ' z$, then $\{z, y\} \in A \cup B$. Hence, $\tau(\succeq, \succeq') \geq |A \cup B| \geq \tau(x, y, \succeq)$. □

The NP-containment of Robust Matching follows from Lemma 4.3.

Proposition 4.4. Robust Matching *with ties is in NP.*

Proof. Let $I = (P_0, d)$ be an instance of Robust Matching with ties with $P_0 = ((\succeq_u^{P_0})_{u \in U}, (\succeq_w^{P_0})_{w \in W})$. To show NP-containment, it suffices to show that checking whether a given matching is $d$-robust can be done in polynomial time. To this end, let $M$ be a matching of $U \cup W$. For notational convenience, for each two agents $x, y$ acceptable to each other, we say in this proof that $y$ *prefers* $x$ to her partner $M(y)$ in $\succ$, written as $x \succ_y M(y)$, if $M(y) = \emptyset$ or $x \succ_y M(y)$.

To check whether $M$ is $d$-robust, for each acceptable and unmatched pair $\{u, w\}$ with $u \in U$ and $w \in W$, we check whether $\tau(M(u), w, \succeq_u^{P_0}) + \tau(M(w), u, \succeq_w^{P_0}) \leq d$. If this holds for some pair, we reject and otherwise accept.

To show the correctness it suffices to prove that $M$ is $d$-robust if and only if for each acceptable and unmatched pair $\{u, w\}$ with $u \in U$ and $w \in W$ it holds that $\tau(M(u), w, \succeq_u^{P_0}) + \tau(M(w), u, \succeq_w^{P_0}) > d$.

The "only if" direction is straightforward: Assume that $M$ is $d$-robust. Suppose, for the sake of contradiction, that there exists an acceptable and unmatched pair $\{u, w\}$ with $u \in U$ and $w \in W$ such that $\tau(M(u), w, \succeq_u^{P_0}) + \tau(M(w), u, \succeq_w^{P_0}) \leq d$. Let $P_1$ be the preference profile derived from $P_0$ by swapping $w$ forward in the preference list $\succeq_u^{P_0}$ of $u$ until she is right in front of $M(w)$ and by

swapping $u$ forward in the preference list $\succeq_w^{P_0}$ of $w$ until she is right in front of $M(u)$ (see Definition 3.11 for an analogous definition for the case without ties). By definition, $\{u, w\}$ is blocking $M$ in $P_1$ and moreover $\tau(P_0, P_1) = \tau(M(u), w, \succeq_u^{P_0}) + \tau(M(w), u, \succeq_w^{P_0}) \leq d$, a contradiction to $M$ being $d$-robust.

For the "if" direction, assume that for each acceptable and unmatched pair $\{u, w\}$ with $u \in U$ and $w \in W$ it holds that $\tau(M(u), w, \succeq_u^{P_0}) + \tau(M(w), u, \succeq_w^{P_0}) > d$. Suppose, for the sake of contradiction, that $M$ is not $d$-robust. Let $P_1$ denote a preference profile with $P_1 = ((\succ_u^{P_1})_{u \in U}, (\succ_w^{P_1})_{w \in W})$ and $\tau(P_0, P_1) \leq d$ and let $\{u, w\}$ be an acceptable pair that is blocking $M$ in $P_1$. This implies that

$$w \succ_u^{P_1} M(u) \text{ and } u \succ_w^{P_1} M(w).$$

Then, by Lemma 4.3(3), we have that $\tau(\succeq_u^{P_0}, \succeq_u^{P_1}) \geq \tau(M(u), w, \succeq_u^{P_0})$ and $\tau(\succeq_w^{P_0}, \succeq_w^{P_1}) \geq \tau(M(w), u, \succeq_w^{P_0})$. Combining this with the definition of $\tau(P_0, P_1)$, we have

$$d < \tau\left(M(u), w, \succeq_u^{P_0}\right) + \tau\left(M(w), u, \succeq_w^{P_0}\right) \leq \tau\left(\succeq_u^{P_0}, \succeq_u^{P_1}\right) + \tau\left(\succeq_w^{P_0}, \succeq_w^{P_1}\right) \leq \tau(P_0, P_1) \leq d,$$

a contradiction. □

Finding a stable matching can be done in $O(n^2)$ time even when ties are present [28]. In contrast, the presence of ties makes Robust Matching NP-complete.

Theorem 4.5. Robust Matching *with ties is NP-complete, and remains NP-hard even when the number $d$ of swaps allowed is one.*

Proof. The NP-containment follows from Proposition 4.4. To show NP-hardness, we reduce from the NP-hard Independent Set problem [23]:

Independent Set
***Input:*** An undirected graph $G$ with vertex set $V(G)$ and edge set $E(G)$, and a positive integer $k \in \mathbb{N}$.
***Question:*** Does $G$ contain an *independent set* of size $k$, i.e., a $k$-vertex subset of $V' \subseteq V(G)$ of pairwise non-adjacent vertices?

Let $I = (G, k)$ be an instance of Independent Set. Further, let $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$ denote the set of vertices and the set of edges in $G$, respectively. Without loss of generality, we assume that the number $n$ of vertices in $V$ is at least three, the solution size $k$ is at least two, and $n - k \geq 2$. We construct an instance of Robust Matching with ties, with two sets of agents, $U$ and $W$, and with the number of allowed swaps equal to $d = 1$.

**Agent set $U$.** This set consists of the following $n + 2m + 2$ agents:

$$\begin{aligned} U &= V \cup E \cup F \cup A, \text{ where} \\ V &= \{v_1, \ldots, v_n\}, \\ E &= \{e_1, \ldots, e_m\}, \\ F &= \{f_1, \ldots, f_m\}, \\ A &= \{a_0, a_1\}. \end{aligned}$$

**Agent set $W$.** This set consists of the following $n + 2m + 2$ agents:

$$\begin{aligned} W &= T \cup S \cup E_V \cup B, \text{ where} \\ T &= \{t_1, \dots, t_{n-k}\}, \\ S &= \{s_1, \dots, s_k\}, \\ E_V &= \{e_\ell^{v_i}, e_\ell^{v_j} \mid e_\ell \in E(G) \text{ with } e_\ell = \{v_i, v_j\}\}, \\ B &= \{b_0, b_1\}. \end{aligned}$$

We call the agents from $V$ *vertex agents* while the agents from $S$ *selector* agents. Note that we use $v_i$ (respectively, $e_\ell$) for both a vertex and its corresponding vertex agent (respectively, an edge and its corresponding edge agent). It will, however, be clear from the context what we are referring to.

The preference lists of the agents are defined as follows, where $[\star]$ means that the elements in $\star$ are ranked in an arbitrary but fixed order, while $(\star)$ means that the elements in $\star$ are tied. The symbol "$\cdots$" at the end of each preference list denotes an arbitrary but fixed order of the remaining not mentioned agents.

**Preference lists of the agents from $U$.**

$$\begin{aligned} \forall i \in [n], \quad & v_i \colon (T) \succ b_0 \succ \left[\left\{e_\ell^{v_i} \mid e_\ell \in E(G) \text{ s. t. } v_i \in e_\ell\right\}\right] \succ (S) \succ \cdots, \\ \forall \ell \in [m] \text{ with } e_\ell = \{v_i, v_j\}, \quad & e_\ell \colon \left(\left\{e_\ell^{v_i}, e_\ell^{v_j}, b_0\right\}\right) \succ \cdots, \\ \forall \ell \in [m] \text{ with } e_\ell = \{v_i, v_j\} \text{ and } i < j, \quad & f_\ell \colon e_\ell^{v_j} \succ e_\ell^{v_i} \succ b_0 \succ \cdots, \\ & a_0 \colon b_0 \succ b_1 \succ \cdots, \\ & a_1 \colon b_1 \succ b_0 \succ \cdots. \end{aligned}$$

**Preference lists of the agents in $W$.**

$$\begin{aligned} \forall i \in [n-k] \quad & t_i \colon (V) \succ a_0 \succ a_1 \succ \cdots, \\ \forall i \in [k] \quad & s_i \colon (V) \succ a_0 \succ a_1 \succ \cdots, \\ \forall \ell \in [m] \text{ with } e_\ell = \{v_i, v_j\} \quad & e_\ell^{v_i} \colon e_\ell \succ a_0 \succ f_\ell \succ v_i \succ \cdots, \\ & e_\ell^{v_j} \colon e_\ell \succ a_0 \succ f_\ell \succ v_j \succ \cdots, \\ & b_0 \colon a_0 \succ a_1 \succ \cdots, \\ & b_1 \colon a_1 \succ a_0 \succ \cdots. \end{aligned}$$

We use $P$ to denote the above preference profile. Before we prove the correctness of our construction, we observe some properties that every stable matching must satisfy.

Claim 1. *Every matching $M$ for $U$ and $W$ that is stable in $P$ must satisfy the following:*
*(1) Each agent $t_i \in T$ must be matched with an agent from $V$, i.e., $M(t_i) \in V$.*
*(2) Each agent $a_j \in A$ must be matched with $b_j$, i.e., $M(a_j) = b_j$.*
*(3) For each edge $e_\ell \in E(G)$ with $e_\ell = \{v_i, v_j\}$, the agents $e_\ell$ and $f_\ell$ must have $\{M(e_\ell), M(f_\ell)\} = \{e_\ell^{v_i}, e_\ell^{v_j}\}$.*
*(4) Each agent $s_i \in S$ must be matched with an agent from $V$.*

Proof. The first statement is straightforward, because every vertex agent $v_i \in V$ and every selector agent $t_j \in T$ ranks each other at the first position.

Analogously, we obtain the second statement.

Now, consider an arbitrary edge $e_\ell \in E(G)$ and let $v_i$ and $v_j$ denote the endpoints of edge $e_\ell$. Since agent $a_0$ is already matched with $b_0$, we can neglect them from the preference lists of

$e_\ell, f_\ell, e_\ell^{v_i}$, and $e_\ell^{v_j}$. Consequently, one can verify that the partners of $e_\ell$ and $f_\ell$ must be from $\{e_\ell^{v_i}, e_\ell^{v_j}\}$.

By the first three statements, there are $k$ agents left from $V$ who each must be matched with some agent from $S$, because every agent from $S$ ranks every agent from $V$ at the first position. (of Claim 1)$\diamond$

Now, we show that $G$ admits an independent set of size $k$ if and only if profile $P$ has a stable matching $M$ that remains stable in each profile $P'$ that differs from $P$ by at most one swap.

For the "only if" direction, let $V' = \{v_{i_1}, \ldots, v_{i_k}\}$ be an independent set of $k$ vertices with $i_1 < i_2 < \cdots < i_k$. For the sake of convenience, let $V \setminus V' = \{v_{j_1}, \ldots, v_{j_{n-k}}\}$ with $j_1 < j_2 < \cdots < j_{n-k}$.

We claim that the following perfect matching $M$ is stable in every profile that differs from the original one by at most one swap:

$$\begin{aligned} M = {} & \{\{a_0, b_0\}, \{a_1, b_1\}\} \cup \{\{\{s_r, v_{i_r}\} \mid v_{i_r} \in V'\}\} \cup \\ & \{\{t_r, v_{j_r}\} \mid v_{j_r} \in V \setminus V'\} \cup \\ & \left\{\left\{e_\ell, e_\ell^{v_i}\right\}, \left\{f_\ell, e_\ell^{v_j}\right\} \mid e_\ell = \{v_i, v_j\} \text{ s. t. } e_\ell \in E(G) \text{ and } v_i \in V'\right\} \cup \\ & \left\{\left\{e_\ell, e_\ell^{v_i}\right\}, \left\{f_\ell, e_\ell^{v_j}\right\} \mid e_\ell = \{v_i, v_j\} \text{ s. t. } e_\ell \in E(G) \text{ and } \{v_i, v_j\} \cap V' = \emptyset \text{ and } i < j\right\}. \end{aligned}$$

Note that the partners of $S$ and $T$ can be of arbitrary order, and that the partners of the agents $e_\ell$ and $f_\ell$ for which none of the endpoints of the corresponding edge $e_\ell$ are in the independent set $S$ can also be swapped. We fix this order for the sake of the simplicity of the reasoning.

Let us check that $M$ is stable in the original profile $P$. It suffices to verify that no agent from $W$ is involved in a blocking pair. Observe that each agent from $T \cup S \cup B$ already receives her most preferred agent. It remains to consider the agents from $E_V$. Consider an arbitrary agent $e_\ell^{v_i} \in E_V$. Since the partner of agent $e_\ell^{v_i}$ is either $e_\ell$ or $f_\ell$, by the preferences of $e_\ell^{v_i}$, agent $a_0$ is the only agent with whom $e_\ell^{v_i}$ may form a blocking pair. However, $a_0$ already receives her most preferred agent. Hence, neither is any agent from $E_V$ involved in a blocking pair. Summarizing, no agent from $W$ is involved in a blocking pair, and hence $M$ is stable in $P$.

To see why $M$ remains stable in every profile, denoted as $P'$, which differs from the original one by at most one swap, we first observe the following:

(1) No agent $a_j$ from $A$ is blocking $M$ in $P'$, since for each agent $y \in W \setminus \{b_j\}$, other than $a_j$'s partner $b_j$, it holds that if $a_j$ shall prefer $y$ to $b_j$ in $P'$, then this agent $y$ must be $b_{1-j}$, because $P$ and $P'$ differ only by one swap. However, agent $b_{1-j}$ will still prefer her partner $a_{1-j}$ to $a_j$, because $P$ and $P'$ differ by only one swap.

(2) Analogously, no agent $b_j$ from $B$ is involved in a blocking pair.

(3) No agent $z$ from $T \cup S$ is involved in a blocking pair due to the following: First, $z$ already receives one of her most preferred agents (under $P$) and there are at least two more agents (from $V$) that are tied with $M(z)$ in $P$; recall that $|V| \geq 3$. This means that if $z$ would form with some other agent, say $y$, a blocking pair in $P'$, then she must prefer $y$ to $M(z)$ in $P'$. However, the swap distance between her original preference list and the new preference list would be at least two, i.e., $\tau(M(z), y, \succeq_z^P) \geq 2$. By Lemma 4.3(3), we obtain the following: $\tau(P, P') \geq \tau(\succeq_z^P, \succeq_z^{P'}) \geq \tau(M(z), y, \succeq_z^P) \geq 2$, a contradiction.

(4) Analogously, no agent $e_\ell$ from $E$ is involved in a blocking pair due to the following: By the construction of $M$, we have that $M(e_\ell) \in \{e_\ell^{v_i}, e_\ell^{v_j}\}$ with $e_\ell = \{v_i, v_j\}$. This means that $e_\ell$ already receives her most preferred agent under $P$. Since there are at least two other agents that are tied with $M(e_\ell)$, no agent is preferred to $M(e_\ell)$ by $e_\ell$ in $P'$, since $P$ and $P'$ differ only by one swap.

(5) No agent $f_\ell$ from $F$ is involved in a blocking pair due to the following: The partner of $f_\ell$ is an agent from $\{e_\ell^{v_i}, e_\ell^{v_j}\}$ with $e = \{v_i, v_j\}$. For each agent $y \in W \setminus \{M(f_\ell)\}$, other than $f_\ell$'s partner $M(f_\ell)$, if agent $f_\ell$ would form with $y$ a blocking pair of $M$ in $P'$, then $y \neq b_0$ (recall that $b_0$ is not involved in any blocking pair) and $f_\ell$ prefers $y$ to $M(f_\ell)$ in $P'$. Since $P$ and $P'$ differ by only one swap, by the preference list of $f_\ell$ and by the definition of $M$, agent $y$ must be an agent $e_\ell^v$ with $v \in e_\ell$ and $M(e_\ell^v) = e_\ell$. However, agent $e_\ell^v$ will still prefer her partner $e_\ell$ to $f_\ell$ as the swap distance from $f_\ell$ to $e_\ell$ in the initial preference list of $e_\ell^v$ is two, i.e., $\tau(e_\ell, f_\ell, \succeq^P_{e_\ell^v}) = 2$. This is not possible by a reasoning analogous to the one given in Point (3).

(6) No agent $v$ from $V$ with $M(v) \in T$ is involved in a blocking pair of $M$ in $P'$, because for each agent $y \in W \setminus \{M(v)\}$, other than her partner $M(v)$, the swap distance from $y$ to $M(v)$ in the initial preference list of $v$ is at least $|T| - 1$, which is larger than one, i.e., $\tau(M(v), y, \succeq^P_v) > 1$. This is not possible by a reasoning analogous to the one given in Point (3).

Thus, to show that $M$ is also stable in $P'$ it suffices to show that no agent $v$ from $V$ with $M(v) \notin T$ is involved in a blocking pair of $M$ in $P'$. Now, consider an arbitrary agent $v_i \in V$ with $M(v_i) \notin T$. By the construction of $M$, it follows that $M(v_i) \in S$. Suppose, for the sake of contradiction, that $v_i$ is forming with an agent $z \in W \setminus \{M(v_i)\}$ a blocking pair of $M$ in $P'$. By the above reasoning, it follows that $z \in W \setminus (T \cup B \cup S) = E_V$. Observe that, for each agent $y$ that is less preferred than $M(v_i)$ (by $v_i$), the swap distance from $y$ to $M(v_i)$ in the preference list of $v_i$ is at least $|S|$, which is more than one. That is, if $v_i$ prefers $M(v_i)$ to $y$ in $P$, then $\tau(M(v_i), y, \succeq^P_{v_i}) > 1$ and hence, $v_i$ and $y$ are not blocking $M$ in $P'$. By the preferences of $v_i$ in $P$, we further infer that $z \in \{e_\ell^{v_i} \mid e_\ell \in E(G) \text{ s.t. } v_i \in e_\ell\}$. Assume that $z = e_\ell^{v_i}$ for some edge $e_\ell \in E(G)$ with $v_i \in e_\ell$. Since $v_i \in V'$, by the definition of $M$, we have that $M(z) = M(e_\ell^{v_i}) = e_\ell$. However, since the swap distance from $v_i$ to $e_\ell$ in the original preference list of $e_\ell^{v_i}$ is two (i.e., $\tau(e_\ell^{v_i}, v_i, \succeq^P_{e_\ell^{v_i}}) = 2$) it follows that $e_\ell^{v_i}$ and $v_i$ are not blocking $M$ in $P'$, a contradiction.

For the "if" direction, let $M$ be a stable matching that is stable in every profile that differs from the original one-by-one swap. We claim that $V' = \{v_i \mid M(v_i) \in S\}$ is an independent set of size $k$. By Claim 1(1) and (4), every agent from $V$ is matched to an agent that is either from $T$ or from $S$. Since $|S| = k$, it follows that $V'$ has $k$ vertices.

To show that $V'$ is an independent set, suppose for the sake of contradiction that $V'$ contains two adjacent vertices $v_i$ and $v_j$ and let $e_\ell = \{v_i, v_j\}$ be the incident edge. This means that $M(v_i), M(v_j) \in S$. Then, consider the profile $P'$ that differs from $P$ by one swap in the preference list of $e_\ell^{v_i}$, depicted as follows:

$$e_\ell^{v_i} : e_\ell \succ a_0 \succ v_i \succ f_\ell.$$

Since $v_i$ prefers agent $e_\ell^{v_i}$ to its partner that is from $S$, the stability of $M$ implies that $M(e_\ell^{v_i}) = e_\ell$; recall that by Claim 1(1), $M(e_\ell^{v_i})$ cannot be matched to $a_0$. Consequently, $M(e_\ell^{v_j}) = f_\ell$. However, consider the profile $P''$ that differs from $P$ by one swap in the preference list of $e_\ell^{v_j}$, depicted as follows:

$$e_\ell^{v_j} : e_\ell \succ a_0 \succ v_j \succ f_\ell.$$

Since $v_i$ prefers agent $e_\ell^{v_j}$ to its partner that is from $S$, it follows that $M$ is not stable in $P''$, as we have just reasoned that $M(e_\ell^{v_j}) = f_\ell$ but $v_j$ prefers $v_j$ to $f_\ell$ in $P''$—a contradiction. □

## 5 NEARLY STABLE MATCHINGS

In the previous two sections, we focused on the robustness of matchings, which is a stronger requirement than the stability. In this section, we move to investigating the other end of the spectrum

of different strengths of stability, namely, we will look at near stability—a particular relaxation of the classical concept of stability. Since a stable matching always exists, looking for nearly stable matchings is only justified when one is also interested in other properties of the sought matching apart from stability. Here, we present our results on the complexity of finding nearly stable matchings that are perfect or within a given egalitarian cost bound.

We start in Section 5.1 by observing that all four problems variants of near stability are NP-hard. Indeed, we provide an even stronger result, which says that under the standard complexity-theoretic assumption P ≠ NP, the minimization variants of all considered problems do not admit a polynomial-time polynomial-factor approximation algorithm. In Section 5.2, we study the influence of the number of allowed swaps on the complexity of the problem variants.

### 5.1 Classical and Approximation Hardness

To show the hardness results, we will focus on the so-called gap variants of our problems and prove that these gap variants are NP-hard. These gap problems can be solved by suitable approximation algorithms so that their NP-hardness will rule out polynomial-time approximation algorithms for our problem. Loosely speaking, for a constant $\alpha > 1$, the $\alpha$-gap variant of some minimization problem $Q$ has as input a specific instance $I$ of $Q$ and a cost upper bound $q \in \mathbb{N}$ and asks to distinguish whether (1) $I$ admits a solution of cost at most $q$, or (2) each solution for $I$ has cost strictly greater than $\alpha \cdot q$. We put no requirement on the answer given to $I$ when the optimum solution is in the "gap" interval $(q, \alpha \cdot q)$. Note that, to decide between these two options, we can use a factor-$\alpha$ approximation algorithm (if it exists), an algorithm that is guaranteed to find a solution of cost at most $\alpha \cdot \mathrm{opt}$ where opt is the minimum cost. Hence, if the $\alpha$-gap problem is NP-hard, a polynomial-time factor-$\alpha$ approximation algorithm implies P = NP.

To make the presentation easier, we use the following decision-focused definition of approximation algorithms, which only refers to the gap variant of an optimization problem. By the reasoning above, ruling out the existence of polynomial-time algorithms to decide such problems rules out the existence of standard form approximation algorithms that produce approximate solutions.

*Definition 5.1.* Let $\mathrm{poly}_1 \colon \mathbb{N} \to \mathbb{N}$ be a polynomial whose domains and co-domains are subsets of the positive integers. An algorithm $\mathcal{A}$ is a *polynomial-time and* $\mathrm{poly}_1$*-approximation algorithm for* Globally Nearly Stable Perfect Matching *(*Global-Near+Perf*)* if for each preference profile $P$ and each positive integer $\mathrm{d_G} \in \mathbb{N}$, the algorithm $\mathcal{A}$ runs in time $|P|^{O(1)}$ and satisfies the following: (1) if $P$ admits a globally $\mathrm{d_G}$-nearly stable and perfect matching, then $\mathcal{A}$ returns "yes," and (2) if $P$ admits no globally $\mathrm{poly}_1(\mathrm{d_G}) \cdot \mathrm{d_G}$-nearly stable and perfect matching, then $\mathcal{A}$ returns "no."

If such an algorithm exists, then we also say that Global-Near+Perf admits a *polynomial-time and polynomial-factor approximation algorithm*.

An approximation algorithm for Locally Nearly Stable Perfect Matching (Local-Near+Perf) is defined analogously (replace all occurrences of global and $\mathrm{d_G}$ by local and $\mathrm{d_L}$, respectively). For the variant where an additional objective is to achieve a given egalitarian cost, we use an even weaker notion of approximation algorithm that allows for bi-criteria approximation.

*Definition 5.2.* Let $\mathrm{poly}_1, \mathrm{poly}_2 \colon \mathbb{N} \to \mathbb{N}$ be two polynomials whose domains and co-domains are on the positive integers. An algorithm $\mathcal{A}$ is a *polynomial-time and* $(\mathrm{poly}_1, \mathrm{poly}_2)$*-approximation algorithm for* Globally Nearly Stable Egalitarian Matching *(*Global-Near+Egal*)* if for each preference profile $P$ and each pair of positive integers $\mathrm{d_G}, \eta \in \mathbb{N}$, the algorithm $\mathcal{A}$ runs in time $|P|^{O(1)}$ and satisfies the following: (1) if $P$ admits a globally $\mathrm{d_G}$-nearly stable matching with egalitarian cost at most $\eta$, then $\mathcal{A}$ returns "yes," and (2) if $P$ admits no globally $\mathrm{poly}_1(\mathrm{d_G}) \cdot \mathrm{d_G}$-nearly stable matching with egalitarian cost at most $\mathrm{poly}_2(\eta) \cdot \eta$, then $\mathcal{A}$ returns "no."

If such an algorithm exists, then we also say that Global-Near+Egal admits a *polynomial-time and polynomial-factor approximation algorithm*.

Here again, an approximation algorithm for Locally Nearly Stable Egalitarian Matching (Local-Near+Egal) is defined analogously (replace all occurrences of global and $d_G$ by local and $d_L$, respectively).

Theorem 5.3. *For each* $\Pi \in$ {Global-Near+Egal, Global-Near+Perf, Local-Near+Perf, Local-Near+Egal}, $\Pi$ *is NP-hard and does not admit a polynomial-time polynomial-factor approximation algorithms, unless* $P = NP$. *For* Local-Near+Perf *and* Local-Near+Egal, *the statement holds even if* $d_L = 1$.

Proof. The NP-hardness will follow from the inapproximability results by setting the corresponding approximation factors to 1. Thus, we only need to show the inapproximability results, which are based on the same basic construction. We first give the details of the construction. Then, we prove that, on the instances resulting from the construction, approximability of Global-Near+Perf, Global-Near+Egal, Local-Near+Perf, or Local-Near+Egal implies polynomial-time solvability of all problems in NP.

Let $\text{poly}_1, \text{poly}_2 \colon \mathbb{N} \to \mathbb{N}$ be two arbitrary polynomials. We will show non-existence of any polynomial-time and $\text{poly}_1$-approximation algorithm (or $(\text{poly}_1, \text{poly}_2)$-approximation algorithm), using a reduction that introduces a gap in the near stability between an optimally nearly stable solution and any other nearly stable solution.

We reduce from the NP-complete Independent Set problem (see page 37 for the definition). Let $I = (G, k)$ be an instance of Independent Set. Let $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$ denote the set of vertices and the set of edges in $G$, respectively. We interpret the edges as two-element subsets of $V(G)$. For each vertex $v_i \in V$, by $E(v_i)$, we denote the set of edges incident with vertex $v_i$ in $G$.

From $G$ and $k$, we will construct a preference profile $P$. The lower thresholds for the gap problems that we construct are as follows: We define threshold for the number of swaps for a globally nearly stable matching as $d_G := m+n$, the threshold for the number of swaps per agent of a locally nearly stable matching as $d_L := 1$. The threshold for the egalitarian cost that we aim for is the same for globally $d_G$-nearly stable and locally $d_L$-nearly stable matchings, namely,

$$\eta := k + (\text{poly}_1(d_G)\cdot d_G + \text{poly}_1(d_L)\cdot d_L + 2) \cdot (3m + (2n + k) \cdot k + (2n - k) \cdot (n - k)).$$

For ease of notation, let $d_G^* := \text{poly}_1(d_G)\cdot d_G + \text{poly}_1(d_L)\cdot d_L + 1$, and $\eta^* := \text{poly}_2(\eta) \cdot \eta + 2$. (Note that we compute all these parameters regardless of the concrete problem that we are reducing to.)

*Construction.* We construct a profile $P$ as follows: We introduce the following pairwise disjoint sets of agents: $V, T, E, F$ (men); $W, S, R, E_V$ (women). Additionally, we introduce the following pairwise disjoint sets of *auxiliary agents*: $A, C$ (men) and $B, D$ (women). Sets $V$ and $W$ will represent the vertices of $G$, sets $R$, $S$, and $T$ will force a selection of $k$ vertices of $G$, and sets $E$, $E_V$, and $F$ will ensure that the selected vertices are pairwise nonadjacent. The auxiliary agents from $A \cup B$ enforce that only swaps of some specific agents are relevant while the auxiliary agents from $C \cup D$ enforce that each matching within some appropriate egalitarian cost must be perfect.

*The non-auxiliary agents.* Specifically, the non-auxiliary sets contain the following agents:

$$V \coloneqq \{v_i \mid v_i \in V(G)\}, \quad W \coloneqq \{w_i \mid v_i \in V(G)\},$$
$$T \coloneqq \{t_i \mid v_i \in V(G)\}, \quad S \coloneqq \{s_i \mid i \in [k]\},$$
$$R \coloneqq \{r_i \mid i \in [n-k]\},$$
$$E \coloneqq \{e_\ell \mid e_\ell \in E(G)\}, \quad E_V \coloneqq \left\{e_\ell^{v_i}, e_\ell^{v_j} \mid e_\ell = \{v_i, v_j\} \text{ for some edge } e_\ell \in E(G)\right\}, \text{ and}$$
$$F \coloneqq \{f_\ell \mid e_\ell \in E(G)\}.$$

Note that we use $v_i$ (respectively, $e_\ell$) for both a vertex and its corresponding *vertex agent* (respectively, an edge and its corresponding *edge agent*). It will, however, be clear from the context what we are referring to. The preference lists of the above agents are defined as follows (men are placed on the left and women on the right). For the sake of readability the non-auxiliary agents are omitted in each list, and we will describe them in detail later on.

$$\forall v_i \in V(G)\colon \quad v_i\colon w_i \succ \left[\left\{e_\ell^{v_i} \mid e_\ell \in E(v_i)\right\}\right] \succ s_1 \succ \cdots \succ s_k, \quad w_i\colon v_i \succ t_i,$$
$$t_i\colon w_i \succ r_1 \succ \cdots \succ r_{n-k},$$
$$\forall j \in [n-k]\colon \quad r_j\colon t_1 \succ \cdots \succ t_n,$$
$$\forall j \in [k]\colon \quad s_j\colon v_1 \succ \cdots \succ v_n,$$
$$\forall e_\ell = \{v_i, v_j\} \in E(G) \text{ with } i < j\colon$$
$$e_\ell\colon e_\ell^{v_i} \succ e_\ell^{v_j}, \quad e_\ell^{v_i}\colon e_\ell \succ v_i \succ f_\ell,$$
$$f_\ell\colon e_\ell^{v_i} \succ e_\ell^{v_j}, \quad e_\ell^{v_j}\colon e_\ell \succ v_j \succ f_\ell.$$

Herein, as before, for some set of agents $X$, by $[X]$, we denote an arbitrary order of them.

*The type-one auxiliary agents $A \cup B$.* These agents ensure that only the swaps in preference lists (as above) of agents in $W \cup E$ are possible; other swaps in the preference lists above will be made impossible by padding enough of the auxiliary agents. We say that the auxiliary agents in $A \cup B$ are of *type one*. For each agent $x$ from $V \cup T \cup F \cup S \cup R \cup E_V$ and for each two consecutive agents $y_1$ and $y_2$ in $x$'s preference list (as described above), we introduce $\mathrm{d}_{\mathrm{G}}^*$ auxiliary agents $a_x^1(y_1, y_2), \ldots, a_x^{\mathrm{d}_{\mathrm{G}}^*}(y_1, y_2)$ to $A$ and $\mathrm{d}_{\mathrm{G}}^*$ auxiliary agents $b_x^1(y_1, y_2), \ldots, b_x^{\mathrm{d}_{\mathrm{G}}^*}(y_1, y_2)$ to $B$ with the following preference lists:

(i) If $x \in V \cup T \cup F$, then for all $i \in [\mathrm{d}_{\mathrm{G}}^*]$ let the preference lists of $a_x^i(y_1, y_2)$ and $b_x^i(y_1, y_2)$ be $b_x^i(y_1, y_2)$ (that is, a singleton list) and $a_x^i(y_1, y_2) \succ x$, respectively, and add all $\mathrm{d}_{\mathrm{G}}^*$ auxiliary agents $b_x^i(y_1, y_2)$ to the preference list of $x$ between agents $y_1$ and $y_2$.

(ii) Otherwise, if $x \in R \cup S \cup E_V$, then for all $i \in [\mathrm{d}_{\mathrm{G}}^*]$ let the preference lists of $a_x^i(y_1, y_2)$ and $b_x^i(y_1, y_2)$ be $b_x^i(y_1, y_2) \succ x$ and $a_x^i(y_1, y_2)$, respectively, and add all $\mathrm{d}_{\mathrm{G}}^*$ auxiliary agents $a_x^i(y_1, y_2)$ to the preference list of $x$ between agents $y_1$ and $y_2$.

In total, we have

$$\begin{aligned}
|A| &= |B| \\
&= \mathrm{d}_{\mathrm{G}}^* \cdot \left( \sum_{i=1}^{n} (|E(v_i)| + k) + |T| \cdot (n-k) + |F| + |R| \cdot (n-1) + |S| \cdot (n-1) + |E_V| \cdot 2 \right) \\
&= \mathrm{d}_{\mathrm{G}}^* \cdot \left( \sum_{i=1}^{n} (|E(v_i)| + k) + n \cdot (n-k) + m + (n-k) \cdot (n-1) + k \cdot (n-1) + 2 \cdot 2m \right) \\
&= \mathrm{d}_{\mathrm{G}}^* \cdot (2n^2 - n + 7m).
\end{aligned}$$

*Type-two auxiliary agents $C \cup D$.* To enforce that every matching with egalitarian cost at most $\mathrm{poly}_2(\eta) \cdot \eta$ must be perfect, we introduce auxiliary agents and append them to the preference list of each non-auxiliary agent and each type-one auxiliary agent. We say that the auxiliary

agents in $C \cup D$ are of *type two*. Formally, for each agent $x \in V \cup T \cup E \cup F \cup A \cup W \cup S \cup R \cup E_V \cup B$, we introduce $\eta^*$ auxiliary agents $C_x := \{c_x^1, \ldots, c_x^{\eta^*}\}$ and add them to $C$, and $\eta^*$ auxiliary agents $D_x := \{d_x^1, \ldots, d_x^{\eta^*}\}$ and add them to $D$; recall that $\eta^* = \mathrm{poly}_2(\eta) \cdot \eta + 2$. The preference lists of these agents are as follows:

(i) If $x \in V \cup T \cup E \cup F \cup A$, then for all $i \in [\eta^*]$ let the preference lists of $c_x^i$ and $d_x^i$ be $d_x^i \succ [D_x \setminus \{d_x^i\}]$ and $c_x^i \succ [C_x \setminus \{c_x^i\}] \succ x$, respectively, and append all $\eta^*$ auxiliary agents $d_x^i$ to the end of the preference list of $x$.

(ii) Otherwise, that is, $x \in W \cup S \cup R \cup E_V \cup B$, then for all $i \in [\eta^*]$ let the preference lists of $c_x^i$ and $d_x^i$ be $d_x^i \succ [D_x \setminus \{d_x^i\}] \succ x$ and $c_x^i \succ [C_x \setminus \{c_x^i\}]$, respectively, and append all $\eta^*$ auxiliary agents $c_x^i$ to the end of the preference list of $x$.

In total, we have $|C| = |D| = \eta^* \cdot (|V| + |T| + |E| + |F| + |A| + |W| + |R| + |S| + |E_V| + |B|)$. Observe that every matching with egalitarian cost at most $\mathrm{poly}_2(\eta) \cdot \eta$ must assign to every type-two auxiliary agent a partner that is also of type two, as otherwise the egalitarian cost induced by such two agents would be at least $\eta^* - 1 = \mathrm{poly}_2(\eta) \cdot \eta + 1 > \mathrm{poly}_2(\eta) \cdot \eta$.

This completes the construction of the profile $P$. Clearly, it can be constructed in polynomial time.

*Correctness of the construction.* In the following, we show that the existence of any polynomial-time $\mathrm{poly}_1$-factor approximation algorithm for Global-Near+Perf or Local-Near+Perf or any polynomial-time $(\mathrm{poly}_1, \mathrm{poly}_2)$-factor approximation algorithm for Global-Near+Egal or Local-Near+Egal implies P = NP. We first prove the following claim:

Claim 2. *(1) If $G$ admits a $k$-vertex independent set, then $P$ admits a globally $\mathsf{d}_G$-nearly stable and perfect matching, which is also locally $\mathsf{d}_L$-nearly stable and has egalitarian cost at most $\eta$.*

*(2) If $P$ admits a globally $\mathrm{poly}_1(\mathsf{d}_G) \cdot \mathsf{d}_G$-nearly stable and perfect matching, then $G$ admits a $k$-vertex independent set.*

*(3) If $P$ admits a locally $\mathrm{poly}_1(\mathsf{d}_L) \cdot \mathsf{d}_L$-nearly stable and perfect matching, then $G$ admits a $k$-vertex independent set.*

Proof. To show the first statement, assume that $V^\star \subseteq V$ is a $k$-vertex independent set of $G$. Construct a perfect matching $M$ for $P$ as follows: Let $i_1 < i_2 < \cdots < i_k$ be the indices of the vertices in $V^\star$, that is, for each $z \in [k]$, we have $v_{i_z} \in V^\star$. Similarly, let $j_1 < j_2 < \cdots < j_{n-k}$ be the indices of the vertices in $V \setminus V^\star$. Matching $M$ contains the following pairs:

(i) For each $z \in [k]$, match $\{v_{i_z}, s_z\} \in M$ and $\{t_{i_z}, w_{i_z}\} \in M$.
By the construction of the preference lists, the egalitarian cost of the pair $\{v_{i_z}, s_z\}$ is at most $(\mathsf{d}_G^* + 1) \cdot (n + k + n)$; recall that, for each agent $x$ who is not an auxiliary agent and who is not in $W \cup E$, we have placed exactly $\mathsf{d}_G^*$ type-one auxiliary agents between each pair of non-auxiliary agents in $x$'s preference list. The egalitarian cost of the pair $\{t_{i_z}, w_{i_z}\}$ is one. In total, these $2k$ pairs contribute at most $k + (\mathsf{d}_G^* + 1) \cdot (2n + k) \cdot k$ units to the egalitarian cost.

(ii) For each $z \in [n - k]$, match $\{v_{j_z}, w_{j_z}\} \in M$ and $\{t_{j_z}, r_z\} \in M$.
The egalitarian cost of the pair $\{v_{j_z}, w_{j_z}\}$ is zero, while the egalitarian cost of the pair $\{t_{j_z}, r_z\}$ is at most $(\mathsf{d}_G^* + 1) \cdot (n - k + n)$. In total, these $2 \cdot (n - k)$ pairs contribute at most $(\mathsf{d}_G^* + 1) \cdot (n - k + n) \cdot (n - k)$ units to the egalitarian cost.

(iii) Further, for each edge $e_\ell \in E(G)$, choose an endpoint from $e_\ell \cap V^\star$ or an arbitrary endpoint of $e_\ell$ if $e_\ell \cap V^\star = \emptyset$. Say that we have picked an endpoint with index $i$. Then, match $\{e_\ell, e_\ell^{v_i}\}, \{f_\ell, e_\ell^{v_j}\} \in M$, where $j$ is the index of the other endpoint of $e_\ell$, different from $i$.
The egalitarian cost of these two pairs is at most $(\mathsf{d}_G^* + 1) \cdot 3$.

(iv) Finally, for each type-one auxiliary agent $a_x^z(y_1, y_2) \in A$, match him with his counter-part from $B$, that is, match $\{a_x^z(y_1, y_2), b_x^z(y_1, y_2)\} \in M$. For each type-two auxiliary agent $c_x^z \in C$, match him with his counter-part from $D$, that is, match $\{c_x^z, d_x^z\} \in M$.
There is no egalitarian cost for these pairs.

This concludes the definition of $M$, which is clearly a perfect matching. One can verify that the egalitarian cost of $M$ is at most

$$k + (\mathrm{d}_{\mathrm{G}}^* + 1) \cdot ((2n + k) \cdot k + (2n - k) \cdot (n - k) + 3m) = \eta.$$

It remains to show that $M$ is globally $\mathrm{d}_{\mathrm{G}}$-nearly stable and locally $\mathrm{d}_{\mathrm{L}}$-nearly stable. We claim that after performing the following swaps $M$ is indeed stable: For each edge agent $e_\ell \in E$ let $e_\ell^{v_i}$ and $e_\ell^{v_j}$ be the two vertex agents in its preference lists with $i < j$. If $M(e_\ell) = e_\ell^{v_j}$, then swap the order of these two agents $e_\ell^{v_i}$ and $e_\ell^{v_j}$ in $e_\ell$'s preference list; recall that there are no other agents between these two agents. For each agent $w_i$ that is not matched to $v_i$, swap $v_i$ with $t_i$ in $w_i$'s preference list; note that there are exactly $k$ such agents. Clearly, for each agent, we have performed at most $1 = \mathrm{d}_{\mathrm{L}}$ swaps and the total number of performed swaps is at most $m + k \leq \mathrm{d}_{\mathrm{G}}$. Denote by $P'$ the profile that results from these swaps.

Next, we show that $M$ is stable for $P'$. Clearly, each auxiliary agent receives its most preferred agent, and, hence, no auxiliary agent can be involved in a blocking pair. Similarly, each agent $w_i \in W$ receives its most preferred agent (after the $k$ swaps performed on agents in $W$), and, hence, no agent from $W$ can be involved in a blocking pair. Since each edge agent $e_\ell$ is matched to its most preferred agent, no blocking pair can involve any edge agent $e_\ell$. Since the partner $M(e_\ell)$ of each edge agent $e_\ell$ already obtains its most preferred agent, no blocking pair can involve any agent $M(e_\ell)$. Further, since these agents, $M(e_\ell)$, are the only agents that may be preferred by any agent $f_\ell$ to $M(f_\ell)$, no blocking pair can involve any $f_\ell$; recall that we have just reasoned that no auxiliary agent is involved in a blocking pair. Furthermore, each vertex agent $v_i$ whose corresponding vertex does not belong to the independent set, i.e., $v_i \in V \setminus V^\star$ is matched to its most preferred agent. Similarly, each agent $t_z$ with $M(t_z) = w_z$ cannot be involved in a blocking pair, because it already obtains its most preferred agent.

A potential blocking pair must hence involve an agent from $\{t_{j_z} \mid v_{j_z} \in V \setminus V^\star\} \cup V^\star$.

Consider an agent $t_{j_z}$ with $v_{j_z} \in V \setminus V^\star$. By the construction of matching $M$, it follows that $M(t_{j_z}) = r_z$. Since neither $w_{j_z}$ nor any auxiliary agent can be involved in a blocking pair, by the preference list of $t_{j_z}$ it follows that $t_{j_z}$ could only form a blocking pair with an agent $r_{z'}$ such that $z' < z$. However, this agent $r_{z'}$ prefers its partner $M(r_{z'}) = t_{j_{z'}}$ to $t_{j_z}$. Hence, no agent $t_{j_z}$ can be involved in a blocking pair.

Consider an agent $v_{i_z}$ that corresponds to a vertex from the independent set $V^\star$. By the construction of matching $M$, it follows that $M(v_{i_z}) = s_z$. By our reasoning above, $w_{i_z}$ will not form a blocking pair with $v_{i_z}$, as it already obtains its most preferred partner. Agent $v_{i_z}$ prefers agent $s_{z'}$ to its partner $s_z$ only if $z' < z$. However, for each $z' < z$, agent $s_{z'}$ prefers its partner $M(s_{z'}) = v_{i_{z'}}$ to agent $v_{i_z}$ (recall that $i_{z'} < i_z$). Thus, no agent from $S$ will form with $v_{i_z}$ a blocking pair. Any blocking pair must thus be of the form $\{v_i, e_\ell^{v_i}\}$ where $v_i \in V^\star$ and $e_\ell^{v_i} \in E(v_i)$. However, since $v_i \in V^\star$, for each of its incident edges, say $e_\ell$, we have matched $e_\ell^{v_i}$ to its most preferred agent $e_\ell$. Thus, indeed, there is no blocking pair, showing that $M$ is stable in $P'$ and hence globally $\mathrm{d}_{\mathrm{G}}$-nearly stable and locally $\mathrm{d}_{\mathrm{L}}$-nearly stable in $P$.

For the second statement of Claim 2, assume that $M$ is a globally $\mathrm{poly}_1(\mathrm{d}_{\mathrm{G}}) \cdot \mathrm{d}_{\mathrm{G}}$-nearly stable and perfect matching for $P$ and let $P'$ be a profile that results from $P$ by making at most $\mathrm{poly}_1(\mathrm{d}_{\mathrm{G}}) \cdot \mathrm{d}_{\mathrm{G}}$ swaps such that $M$ is stable in $P'$.

Recall that for each agent $y$ that is either non-auxiliary or an auxiliary agent of type one, we have introduced $2 \cdot \eta^*$ type-two auxiliary agents, contained in $C_y$ and in $D_y$. Observe that either $C_y$ or $D_y$ finds only the agents from the other set acceptable. Hence, by the perfectness of $M$, the partners of all agents from $C_y$ are exactly the agents from $D_y$. Consequently, we can ignore all type-two auxiliary agents in the preference lists of the remaining agents when discussing the partners of type-one auxiliary agents and non-auxiliary agents. In particular, for each pair of type-one auxiliary agents $a_x^z(y_1, y_2)$ and $b_x^z(y_1, y_2)$, one of them finds only the other agent acceptable (ignoring the type-two auxiliary agents). Again, by the perfectness of $M$, we infer that each $a_x^z(y_1, y_2)$ is matched to its counter-part $b_x^z(y_1, y_2)$. Hence, from now on, when discussing the partners of a non-auxiliary agent, we only need to consider the non-auxiliary agents in its preference list.

Recall that there are at least $\mathsf{d}_\mathsf{G}^*$ type-one auxiliary agents between each pair of non-auxiliary agents in the preference list of each agent from $V \cup R \cup S \cup T \cup E_v \cup F$. By definition $\mathsf{d}_\mathsf{G}^* = \mathsf{poly}_1(\mathsf{d}_\mathsf{G}) \cdot \mathsf{d}_\mathsf{G} + \mathsf{poly}_1(\mathsf{d}_\mathsf{L}) \cdot \mathsf{d}_\mathsf{L} + 1$ (recall that we compute $\mathsf{poly}_1(\mathsf{d}_\mathsf{G}) \cdot \mathsf{d}_\mathsf{G}$ and $\mathsf{poly}_1(\mathsf{d}_\mathsf{L}) \cdot \mathsf{d}_\mathsf{L}$ regardless of the variant that we reduce to). It is impossible to perform $\mathsf{poly}_1(\mathsf{d}_\mathsf{G}) \cdot \mathsf{d}_\mathsf{G}$ swaps to switch the positions of two non-auxiliary agents in the preference list of an agent from $V \cup R \cup S \cup T \cup E_V \cup F$. Thus, the only swaps performed to obtain $P'$ are without loss of generality in the preference lists of agents in $E \cup W$ and are only within the non-auxiliary agents.

Let $V' = \{v_i \in V(G) \mid M(v_i) \in S\}$. We claim that $V'$ is a $k$-vertex independent set in $G$. First, $V'$ has cardinality $k$, because $M$ is perfect and the only remaining acceptable partners to every agent in $S$ are those in $V$. Suppose, for the sake of contradiction, that $V'$ contains two adjacent vertices $v_i$ and $v_j$ and let $e_\ell = \{v_i, v_j\}$ be their incident edge. By the definition of $V'$, it follows that both $v_i$ and $v_j$ are assigned partners from $S$. Since $v_i$ prefers $e_\ell^{v_i}$ to every agent from $S$ and since $e_\ell^{v_i}$ prefers only $e_\ell$ to $v_i$, by the stability of $M$ in $P'$, it follows that $M(e_\ell^{v_i}) = e_\ell$; recall that no swaps are performed in between any two non-auxiliary agents in the preference lists of the agents from $V \cup E_V$. Analogously, it must hold that $M(e_\ell^{v_j}) = e_\ell$—a contradiction to $M$ being a matching.

The reasoning for the third statement is analogous to what we have done for the second statement. Instead of arguing about the total number of swaps, we only need to argue that the number of swaps changed per agent is $\mathsf{poly}_1(\mathsf{d}_\mathsf{L}) \cdot \mathsf{d}_\mathsf{L}$, which is strictly smaller than $\mathsf{d}_\mathsf{G}^*$. Thus, it is still impossible to change the positions of two non-auxiliary agents in the preference list of any non-auxiliary agent from $V \cup R \cup S \cup T \cup E_v \cup F$. (of Claim 2) ⋄

The next claim establishes a close connection between the egalitarian cost and the perfectness of a matching.

Claim 3. *If $M$ is a matching for $P$ with egalitarian cost of at most $\mathsf{poly}_2(\eta) \cdot \eta$, then this matching must be perfect.*

Proof. Assume that $M$ has egalitarian cost at most $\mathsf{poly}_2(\eta) \cdot \eta$. If $M$ is not perfect, then thet egalitarian cost contributed by an unmatched agent is equal to the length of this agent's preference list. This exceeds the budget $\mathsf{poly}_2(\eta) \cdot \eta$, because the length of each agent's preference list is at least $\eta^* > \mathsf{poly}_2(\eta) \cdot \eta + 1$. (of Claim 3)⋄

Now, we continue with our correctness proof.

*Inapproximability of* Global-Near+Perf. Suppose, for the sake of contradiction, that there exists a $\mathsf{poly}_1$-approximation algorithm $\mathcal{A}$ for Global-Near+Perf, running in polynomial-time. Then, we can use algorithm $\mathcal{A}$ to decide Independent Set in polynomial time, showing P = NP. Given an arbitrary instance $I = (G, k)$ of Independent Set, we construct profile $P$ and define $\mathsf{d}_\mathsf{G}$ as described above and let $\mathcal{A}$ run on $(P, \mathsf{d}_\mathsf{G})$. If $I$ is a yes-instance, then by the first implication of Claim 2, it follows that $P$ admits a globally $\mathsf{d}_\mathsf{G}$-nearly stable and perfect matching, and

by Definition 5.1, $\mathcal{A}$ returns "yes." If $I$ is a no-instance, then by the contra-positive of the second implication of Claim 2, it follows that $P$ does not admit a globally $\mathsf{poly}_1(\mathsf{d_G})\cdot\mathsf{d_G}$-nearly stable and perfect matching. By Definition 5.1, $\mathcal{A}$ returns "no."

*Inapproximability of* Global-Near+Egal. Again, suppose, for the sake of contradiction, that there exists a $(\mathsf{poly}_1, \mathsf{poly}_2)$-approximation algorithm $\mathcal{A}$ for Global-Near+Egal, running in polynomial-time. Then, we can use algorithm $\mathcal{A}$ to decide Independent Set in polynomial time, showing P = NP, as follows: Given an arbitrary instance $I = (G, k)$ of Independent Set, we construct profile $P$ and define $\mathsf{d_G}$, $\mathsf{d_L}$, and $\eta$ as described above and let $\mathcal{A}$ run on $(P, \mathsf{d_G}, \eta)$. If $I$ is a yes-instance, then by the first implication of Claim 2 it follows that $P$ admits a globally $\mathsf{d_G}$-nearly stable matching with egalitarian cost at most $\eta$, and by Definition 5.2, $\mathcal{A}$ returns "yes." If $I$ is a no-instance, then by the contra-positive of the second implication of Claim 2 it follows that $P$ does not admit a globally $\mathsf{poly}_1(\mathsf{d_G})\cdot\mathsf{d_G}$-nearly stable perfect matching. This implies that $P$ does not admit a globally $\mathsf{poly}_1(\mathsf{d_G})\cdot\mathsf{d_G}$-nearly stable matching with egalitarian cost at most $\mathsf{poly}_2(\eta)\cdot\eta$ as otherwise, by Claim 3, we will have a globally $\mathsf{poly}_1(\mathsf{d_G})\cdot\mathsf{d_G}$-nearly stable and perfect matching for $P$—a contradiction. By Definition 5.2, $\mathcal{A}$ returns "no."

*Inapproximability of* Local-Near+Perf. Suppose, for the sake of contradiction, that there exists a $\mathsf{poly}_1$-approximation algorithm $\mathcal{A}$ for Local-Near+Perf, running in polynomial-time. Then, we can use algorithm $\mathcal{A}$ to decide Independent Set in polynomial time, showing P = NP, as follows: Given an arbitrary instance $I = (G, k)$ of Independent Set, we construct profile $P$ and define $\mathsf{d_G}$, $\mathsf{d_L}$, and $\eta$ as described above and let $\mathcal{A}$ run on $(P, \mathsf{d_L})$. If $I$ is a yes-instance, then by the first implication of Claim 2, $P$ also admits a locally $\mathsf{d_L}$-nearly stable and perfect matching. Thus, by Definition 5.1, $\mathcal{A}$ returns "yes." If $I$ is a no-instance, then by the contra-positive of the third implication from Claim 2, $P$ does not admit a locally $\mathsf{poly}_1(\mathsf{d_L})\cdot\mathsf{d_L}$-nearly stable perfect matching. By Definition 5.1, $\mathcal{A}$ returns "no."

*Inapproximability of* Local-Near+Egal. Suppose, for the sake of contradiction, that there exists a $(\mathsf{poly}_1, \mathsf{poly}_2)$-approximation algorithm $\mathcal{A}$ for Local-Near+Egal, running in polynomial-time. Then, we can use algorithm $\mathcal{A}$ to decide Independent Set in polynomial time, showing P = NP as follows: Given an arbitrary instance $I = (G, k)$ of Independent Set, we construct profile $P$ and define $\mathsf{d_G}$, $\mathsf{d_L}$, and $\eta$ as described above and let $\mathcal{A}$ run on $(P, \mathsf{d_L}, \eta)$. If $I$ is a yes-instance, then by the first implication of Claim 2 it follows that $P$ admits a locally $\mathsf{d_L}$-nearly stable matching with egalitarian cost at most $\eta$, and by Definition 5.2, $\mathcal{A}$ returns "yes." If $I$ is a no-instance, then by the contra-positive of the third implication from Claim 2, $P$ does not admit a locally $\mathsf{poly}_1(\mathsf{d_L})\cdot\mathsf{d_L}$-nearly stable perfect matching. By the contra-positive of Claim 3, $P$ does not admit a locally $\mathsf{poly}_1(\mathsf{d_L})\cdot\mathsf{d_L}$-nearly stable matching with egalitarian cost at most $\mathsf{poly}_2(\eta)\cdot\eta$. By Definition 5.2, $\mathcal{A}$ returns "no." □

### 5.2 Parameterized Complexity

We now investigate the influence of three natural parameters on the complexity of obtaining nearly stable matchings: "total number $\mathsf{d_G}$ of swaps," "number $\mathsf{n_u}$ of initially unmatched agents," and "number $\mathsf{n_m}$ of initially matched agents"; the latter two will be defined formally below. Note that Theorem 5.3 implies that even only one swap leaves the problems pertaining to locally nearly stable matchings NP-hard. This is different from the globally nearly stable variants, for which simple polynomial-time algorithms for a constant number of swaps exist. However, we show that removing the dependence on the number of swaps in the exponent in the running time is impossible unless FPT = W[1].

Proposition 5.4. Global-Near+Perf *and* Global-Near+Egal *are solvable in* $O((2n)^{2d_G+2})$ *time and* $O((2n)^{2d_G+4})$ *time, respectively.*

Proof. The algorithm for Global-Near+Perf works as follows: Iterate over all of the at most $\binom{2n(n-1)}{d_G}$ possibilities for making $d_G$ swaps. For each of the resulting at most $(2n)^{2d_G}$ profiles $P$, compute in $O(n^2)$ time a stable matching using Gale and Shapley's algorithm (see Theorem 3.3). If one of these matchings is perfect, then accept and otherwise reject. Since all stable matchings match the same set of agents, the input instance for Global-Near+Perf is positive if and only if the algorithm accepts. Clearly, the running time is $O((2n)^{2d_G+2})$, as required.

The algorithm for Global-Near+Egal is similar: Iterate over all of the at most $\binom{2n(n-1)}{d_G}$ possibilities for making $d_G$ swaps. For each of the resulting profiles, use Irving et al.'s $O(n^4)$-time algorithm for computing a stable matching with minimum egalitarian cost. If one of these matchings satisfies the given upper bound on the egalitarian cost, then accept and otherwise reject. Clearly, this algorithm is correct and runs in $O((2n)^{2d_G+4})$ time. □

A substantial improvement on the above rather trivial $n^{O(d_G)}$-time algorithms would imply a major breakthrough, as the following theorem shows:

Theorem 5.5. Global-Near+Perf *and* Global-Near+Egal *are W[1]-hard with respect to the number* $d_G$ *of swaps. Moreover, they both cannot be decided by any algorithm that runs in* $n^{o(d_G)}$ *time unless the Exponential Time Hypothesis fails.*

Proof. We first show the results for Global-Near+Perf. Then, we show how to adapt the proof to show an analogous result for Global-Near+Egal. To show the results for Global-Near+Perf, we provide a polynomial-time reduction from the W[1]-complete Independent Set problem, parameterized by the size, $k$, of the independent set [15] such that the number of allowed swaps is $d_G = 2k$ (also see page 4 for the formal definition).

*Construction.* Let $(G, k)$ be an instance of Independent Set where we seek an independent set of size $k$ in the $n$-vertex, $m$-edge graph $G$, with $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$. We construct a preference profile $P$ with two disjoint sets of agents, $A$ and $B$, each consisting of five groups and a dummy agent: $A := T \cup V \cup W \cup E \cup E_Y \cup \{h_1\}$ and $B := S \cup X \cup Y \cup F \cup F_V \cup \{h_2\}$. Note that we use the symbols $v_i$ (respectively, $e_\ell$) for both vertices and agents (respectively, for both edges and agents). It will, however, be clear from the context what we are referring to when we use them. The two dummy agents $h_1$ and $h_2$ are used to make performing some swaps non-beneficial.

*Agent sets $T$ and $S$.* For each $z \in [k]$, introduce two *selection agents $t_z$ and $s_z$* and add them to $T$ and $S$, respectively. These agents will be unmatched in every stable matching of $P$ and matching them will force a selection of $k$ vertices from $G$ into an independent set. Their acceptable agents are a subset of the vertex agents that we introduce as follows:

*Agent sets $V$, $W$, $X$, and $Y$.* For each vertex $v_i \in V(G)$, introduce four agents $v_i$, $w_i$, $x_i$, and $y_i$, and add them to the sets $V$, $W$, $X$, and $Y$, respectively. We call them *vertex agents* below. For each $i \in [n]$, these agents will form a path $v_i, x_i, w_i, y_i$ in the acceptability graph (see Figure 6). The basic idea is that, in the initial profile, *every* stable matching must match agent $v_i$ to agent $x_i$ and agent $w_i$ to $y_i$. As we will see, such matchings are imperfect, since the agents from $S$ are unmatched in every stable matching. To obtain a perfect matching, we must match some selection agent $s_z \in S$ to some vertex agent $v_i$, selecting the corresponding vertex $v_i$ into a solution for the input graph. This will incur two swaps to make the resulting matching stable. Below, we introduce edge agents

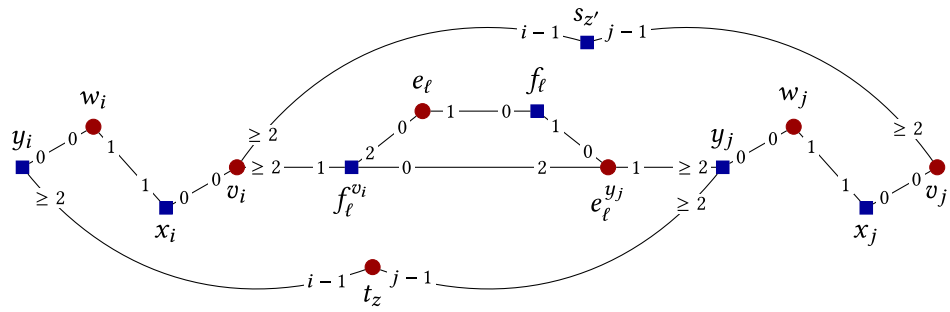



Fig. 6. Relevant part of the acceptability graph of the profile constructed in the proof of Theorem 5.5.

and add them to the preference lists of agents $v_i$ and $y_i$, to ensure that the selected vertices induce an independent set.

*Agent sets $E$, $E_Y$, $F$, and $F_V$.* For each edge $e_\ell \in E(G)$, denote the endpoints of $e_\ell$ by $v_i$ and $v_j$ such that $i < j$. Introduce four *edge agents* $e_\ell$, $e_\ell^{y_j}$, $f_\ell$, and $f_\ell^{v_i}$, and add them to $E$, $E_Y$, $F$, and $F_V$, respectively.

*Preference lists of the agents.* The preference lists of the agents are constructed as follows: Men are on the left and women on the right. Here, for each $i \in [n]$ define subsets $F^i\colon \{f_\ell^{v_i} \mid e_\ell = \{v_i, v_j\} \in E(G) \text{ with } i < j\}$ and $E^i\colon \{e_\ell^{y_i} \mid e_\ell = \{v_r, v_i\} \mid z < i\}$, and for some set $Z$, the notation $[Z]$ means an arbitrary but fixed linear order of $Z$.

$$
\begin{array}{lll}
 & h_1\colon h_2 \succ [Y], & h_2\colon h_1 \succ [V], \\
\forall z \in [k], & t_z\colon y_1 \succ \cdots \succ y_n, & s_z\colon v_1 \succ \cdots \succ v_n. \\
\forall e_\ell = \{v_i, v_j\} \in E(G) \text{ with } i < j, & & \\
 & e_\ell\colon f_\ell^{v_i} \succ f_\ell, & f_\ell\colon e_\ell \succ e_\ell^{y_j}, \\
 & e_\ell^{y_j}\colon f_\ell \succ y_j \succ f_\ell^{v_i}, & f_\ell^{v_i}\colon e_\ell^{y_j} \succ v_i \succ e_\ell. \\
\forall i \in [n], & v_i\colon x_i \succ h_2 \succ [F^i] \succ s_1 \succ \cdots \succ s_k, & y_i\colon w_i \succ h_1 \succ [E^i] \succ t_1 \succ \cdots \succ t_k, \\
 & w_i\colon y_i \succ x_i, & x_i\colon v_i \succ w_i,
\end{array}
$$

Figure 6 depicts the crucial part of the induced acceptability graph for an edge $e_\ell = \{v_i, v_j\} \in E(G)$ with $i < j$. The weights at both sides of the edges denote the ranks of the respective endpoint towards the other endpoint.

To complete the construction, define the total number of swaps as $\mathrm{d_G} = 2k$. Clearly, the construction can be done in polynomial time. As a side remark, notice that every stable matching from the constructed profile does not match exactly $2k$ agents, namely, those from $T \cup S$.

*Correctness.* To show that our construction is indeed a parameterized reduction, we need to show that $G$ admits a $k$-vertex independent set if and only if there exists a preference profile $P'$ with $\tau(P, P') \leq d = 2k$ that admits a perfect stable matching $M$.

For the "only if" part, assume that there exists a $k$-vertex independent set $V' \subseteq V$ in $G$. We define the preference profile $P'$ by performing the following $\mathrm{d_G} = 2k$ swaps that involve the agents from $W \cup X$ that correspond to the vertices from the independent set: For each $z \in [k]$, swap the two agents $y_i$ and $x_i$ in the preference list of agent $w_i$, and swap the agents $v_i$ and $w_i$ in the preference list of agent $x_i$.

Now, we construct the following perfect matching $M$ for $P'$: Let $V' = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$, where $i_1 < i_2 < \cdots < i_k$.

(1) Put $\{h_1, h_2\} \in M$.
(2) For each $z \in [k]$, put $\{v_{i_z}, s_z\}, \{w_{i_z}, x_{i_z}\}, \{t_z, y_{i_z}\} \in M$.
(3) For each $i \in [n] \setminus \{i_1, i_2, \ldots, i_k\}$, put $\{v_i, x_i\}, \{w_i, y_i\} \in M$.
(4) For each edge $e_\ell \in E(G)$, let $v_i$ and $v_j$ be the two endpoints of edge $e_\ell$ with $i < j$, and do the following: Recall that we have created two agents $f_\ell^{v_i}$ and $e_\ell^{v_j}$. If $v_i \in V'$ belongs to the independent set, implying that $v_j \in V \setminus V'$, then put $\{e_\ell, f_\ell\}, \{e_\ell^{y_i}, f_\ell^{v_j}\} \in M$. Otherwise, if $v_i \notin V'$, then put $\{e_\ell, f_\ell^{v_i}\}, \{e_\ell^{y_j}, f_\ell\} \in M$.

Clearly, $M$ is perfect. We claim that $M$ is also stable for $P'$.

Suppose, for the sake of contradiction, that $p$ is a blocking pair of $M$ for the profile $P'$. First, observe that $p$ cannot involve $h_1$, $h_2$, $w_i$, or $x_i$ for any $i \in [n]$, as these agents already obtain their most preferred agents in $P'$. For the same reason, $p$ cannot involve any agent $v_i$ for some $i \in [n] \setminus \{i_1, i_2, \ldots, i_k\}$. Further, $p$ cannot involve an agent $s_z$ or an agent $t_z$ for any $z \in [k]$ because of the following: For each agent $c$ such that agent $s_z$ prefers $c$ to $M(s_z) = v_{i_z}$, we have that $c$ equals either some agent $v_i$ with $i \in [n] \setminus \{i_1, i_2, \ldots, i_k\}$, who already obtains his most preferred partner $M(v_i) = x_i$, or $c$ equals some agent $v_{i_r}$ with $r < z$, who prefers his partner $M(v_{i_r}) = s_{i_r}$ to agent $s_{i_z}$. Using a similar reasoning, we thus obtain that $p$ can neither involve an agent $t_z$, $z \in [k]$. Moreover, $p$ cannot involve two edge agents who correspond to two different edges or one vertex agent and one edge agent such that the corresponding vertex and edge are not incident to each other. Combining all of the above observations, we infer that $p$ either (a) involves two edge agents who correspond to the same edge or (b) a vertex agent and an edge agent such that the corresponding vertex and edge are incident to each other.

For Case (a), observe that in each pair of mutually acceptable edge agents that correspond to the same edge, one receives its most-preferred partner in $M$ with respect to $P'$. Hence, Case (b) must hold. Let $e_\ell$ be the edge corresponding to the edge agent involved in $p$ and let $v_i$ and $v_j$ be the two endpoints of edge $e_\ell$ with $i < j$. Thus, $p = \{e_\ell^{y_j}, y_j\}$ or $p = \{v_i, f_\ell^{v_i}\}$.

If $p = \{e_\ell^{y_j}, y_j\}$, by the definition of blocking pairs, it follows that $M(e_\ell^{y_j}) = f_\ell^{v_i}$ and $M(y_j) = t_z$ for some $z \in [k]$. However, by the definition of $M$, from $M(y_j) = t_z$, we infer that $v_j \in V'$ and from $M(e_\ell^{y_j}) = f_\ell^{v_i}$, we infer $v_j \in V \setminus V'$, a contradiction.

Analogously, if $p = \{v_i, f_\ell^{v_i}\}$, by the definition of blocking pairs, then it follows that $M(v_i) = s_z$ for some $z \in [k]$ and that $M(f_\ell^{v_i}) = e_\ell$. However, by our definition of $M$, from $M(v_i) = s_z$, we infer $v_i \in V'$ and from $M(f_\ell^{v_i}) = e_\ell$, we infer $v_i \in V \setminus V'$, a contradiction.

Hence, indeed, $M$ is stable in $P'$.

For the "if" part, assume that there exists a perfect matching $M$ for profile $P$ and there exists a preference profile $P'$ with $\tau(P, P') \leq d = 2k$ such that $M$ is stable for $P'$. By the perfectness of $M$ there are $i_1, \ldots, i_k \in [n]$ such that for each $z \in [k]$, we have that $M(v_{i_z}) \in S$. We show that the vertex subset $V' \coloneqq \{v_{i_1}, \ldots, v_{i_k}\}$ is a $k$-vertex independent set in $G$.

First, we claim the following:

Claim 4. *For each agent $v_i \in V$ it holds that $M(v_i) \in S$ if and only if $M(y_i) \in T$. Moreover, no agent in $V \cup Y \cup E_Y \cup F_V$ changes her preference list between $P$ and $P'$.*

Proof. We define two subsets $I_1 \coloneqq \{i \in [n] \mid M(v_i) \in S\}$ and $I_2 \coloneqq \{i' \in [n] \mid M(y_{i'}) \in T\}$. To show the first statement, it suffices to show that $I_1 = I_2$. Clearly, $|I_1| = |I_2| = k$, because $|S| = |T| = k$ and $M$ is a perfect matching.

If we can show that for each $i \in I_1 \cup I_2$ there are exactly two distinct swaps in $P'$ in comparison to $P$, exactly one swap in $x_i$'s preference list and exactly one swap in $w_i$'s preference list, then by the swap budget $\mathrm{d_G} = 2k$ and by the cardinalities of $I_1$ and $I_2$, it follows that $I_1 = I_2$.

Now, consider an index $i \in I_1$; the case when $i \in I_2$ is symmetric and omitted. By our definition of $I_1$, we have that $M(v_i) \in S$. Since $M$ is perfect, it follows that $\{x_i, w_i\} \in M$. Since $x_i$ and $v_i$ are each other's most preferred agent in profile $P$, at least one swap occurs in the preference lists of $x_i$ or $v_i$ to make $M$ stable for $P'$; otherwise, they would form a blocking pair. Similarly, since $w_i$ and $y_i$ are each other's most preferred agent in $P$, at least one swap occurs in the preference lists of $w_i$ or $y_i$ to make $M$ stable for $P'$. Thus, for each index $i \in I_1$ there occurs at least two distinct swaps. Since $|I_1| = k$ and since there are overall at most $2k$ swaps, we thus infer that for each $i \in I_i$ there occurs exactly one swap in the preference lists of $x_i$ and $v_i$ and exactly one swap in the preference lists of $w_i$ and $y_i$. Furthermore, it follows that only agents in $V \cup X \cup W \cup Y$ may have a different preference list in $P'$ compared to $P$. This, in particular, implies that $\{h_1, h_2\} \in M$, as otherwise, we need at least one more swap to make $M$ stable in $P'$.

Observe that at least one agent, namely, $h_2$, is between $v_i$'s partner $M(v_i) \in S$ and $x_i$ in $v_i$'s preference list. Thus, it takes more than one swap to make $M$ stable in $P'$ if we change the preference list of $v_i$ and not the preference list of $x_i$. Analogously, at least one agent, namely, $h_1$, is between $y_i$'s partner $M(y_i)$ and $w_i$. Recall that we have just reasoned that $M(h_1) = h_2$. Thus, it also takes more than one swap to make $M$ stable in $P'$ if we change the preference list of $y_i$ and not that of $w_i$. Summarizing, to make $M$ stable for $P'$, there is exactly one swap in the preference list of $x_i$ and there is exactly one swap in the preference list of $w_i$.

The second statement of Claim 4 follows directly from our swap budget and from the above reasoning that for each $i \in I_1$, there is exactly one swap in the preference list of $x_i$ and there is exactly one swap in the preference list of $w_i$ that are performed for $P$ to obtain $P'$. (of Claim 4)$\diamond$

To show that $V'$ is indeed an independent set, suppose, towards a contradiction, that $V'$ contains two adjacent vertices $v_i$ and $v_j$ with $i < j$. Let $e_\ell = \{v_i, v_j\}$ be the incident edge. By the first statement in Claim 4, it follows that $M(y_j) \in T$. By the second statement in Claim 4, agent $v_i$ does not change her preference list in $P'$, meaning that agent $v_i$ prefers $f_\ell^{v_i}$ to her partner $M(v_i) \in S$ in $P'$. By the stability of $M$ in $P'$, it follows that $f_\ell^{v_i}$ prefers her partner $M(f_\ell^{v_i})$ to $v_i$. Again, by the second statement in Claim 4, agent $f_\ell^{v_i}$ does not change her preference list in $P'$, meaning that $f_\ell^{v_i}$ prefers only $e_\ell^{y_j}$ to $v_i$. By the stability of $M$ in $P'$, we have $M(f_\ell^{v_i}) = e_\ell^{y_j}$. This implies that $\{e_\ell^{y_j}, y_j\}$ is a blocking pair for $M$ in $P'$, because $M(y_j) \in T$ by the above and by the second statement of Claim 4. This is a contradiction. Thus, indeed $I'$ is a $k$-vertex independent set. The correctness follows.

The fact that an $n^{o(\mathrm{d_G})}$-time algorithm for Global-Near+Perf would contradict the Exponential Time Hypothesis follows from this reduction in conjunction with the fact that an $n^{o(k)}$-time algorithm for Independent Set would contradict the Exponential Time Hypothesis [15].

*The egalitarian case.* To show the desired statements for the egalitarian case, we use the same idea as we used in the proof of Theorem 5.3: We construct the preference lists exactly as in the case of perfectness. Then, we introduce a sufficiently large number, $2\Delta$, (to be specified later) of *auxiliary agents* and we append all of them to the end of each preference list (that is, they form the least-preferred agents in the new preference lists). We are then asking for a matching that has egalitarian cost of at most $\Delta - 1$. Given an independent set of size $k$ for $G$, we first proceed as in the perfectness case above, obtaining a matching $M$. We set $\Delta$ large enough so that $M$ has egalitarian cost at most $\Delta - 1$: No cost is induced in stage 1 of the construction of $M$. Stage 2 costs at most $k(2(n+k)+2k+2)$. Stage 3 does not cost anything. Stage 4 costs at most $6m$. Hence, we put $\Delta := 2nk + 4k^2 + 2k + 6m$. Then, we complete $M$ to a matching for the whole profile including the auxiliary agents: We define the preference lists of the auxiliary agents so that it is possible to match all auxiliary agents in pairs in a way that they all receive their most preferred partner (see the proof of Theorem 5.3). This way the egalitarian cost is not increased and we obtain a matching with the desired properties.

The other direction of the proof follows in an analogous way to the proofs of Claim 2(2) and Claim 3 in Theorem 5.3: The preference lists of the auxiliary agents are defined so that in each list $\Delta$ auxiliary agents come before any non-auxiliary agent. Thus, a matching $M$ for the constructed profile that has egalitarian cost at most $\Delta - 1$ does not involve any pair containing both an auxiliary agent and a non-auxiliary agent, because such a pair would already contribute cost of at least $\Delta$. From this it follows that $M$ is perfect when restricted to the non-auxiliary agents, because an unmatched non-auxiliary agent would induce cost at least $\Delta$. From there, we can continue the correctness proof as in the case of perfectness as written above. □

By Theorem 2.5, the set of unmatched agents is the same across all stable matchings of a given preference profile $P$. We call an agent *initially unmatched* if she is not contained in any stable matching of the initial profile; otherwise, she is *initially matched*. From Theorem 2.8 it follows that, to assign partners to the $\mathsf{n_u}$ initially unmatched agents, we need to allow for at least $\mathsf{d_G} \geq \mathsf{n_u}/2$ swaps in Global-Near+Perf. The number $\mathsf{n_u}$ is thus a smaller parameter than $\mathsf{d_G}$, meaning that it could be harder to obtain parameterized tractability result with respect to $\mathsf{n_u}$ than to $\mathsf{d_G}$. Indeed, we obtain intractability for $\mathsf{n_u}$.

Corollary 5.6. Global-Near+Perf *and* Global-Near+Egal *are W[1]-hard with respect to the number of initially unmatched* $\mathsf{n_u}$ *agents. This also holds for* Local-Near+Perf *and* Local-Near+Egal, *even if* $\mathsf{d_L} = 1$.

Proof. The reduction in the proof of Theorem 5.3 indeed is a parameterized reduction with respect to the number of unmatched agents that shows W[1]-hardness: First, observe that the reduction runs in polynomial time. Second, Independent Set is W[1]-hard with respect to the number $k$ of vertices in the sought independent set. Third, the number of unmatched agents in any stable matching of the initial profile $P$ constructed by the reduction is at most $2k$. To see this, observe the following: Let $M$ be a stable matching for $P$. For each vertex agent $v_i \in V$, it must hold that $\{v_i, w_i\} \in M$, since this pair would otherwise block $M$. Thus, for each edge $e_\ell = \{v_i, v_j\} \in E(G)$, the edge agent $e_\ell$ is matched by $M$ to either $e_\ell^{v_i}$ or $e_\ell^{v_j}$, and $f_\ell$ is matched to $\{e_\ell^{v_i}, e_\ell^{v_j}\} \setminus \{M(e_\ell)\}$. Furthermore, for each $z \in [n-k]$, agent $t_z \in T$ is matched to agent $r_z \in R$, saturating $R$. Hence, the only unmatched agents are the $k$ agents in $\{t_{n-k+1}, \ldots, t_n\}$ and the $k$ agents in $S$. □

In contrast, it is quite straightforward to obtain a fixed-parameter algorithm for Global-Near+Perf with respect to the number $\mathsf{n_m}$ of *initially matched agents*, that is, the number of agents that occur in every stable matching of the initial profile. (Recall that this quantity is well-defined, because every stable matching matches the same number of agents by Theorem 2.5.)

Proposition 5.7. Global-Near+Perf *can be decided in* $O(2^{2\mathsf{n_m}^2 \cdot \log(\mathsf{n_m})} \cdot n^2)$ *time and admits a problem kernel with* $2\mathsf{n_m}$ *agents that can be computed in* $O(n^2)$ *time.*

Proof. Let $P$ be a preference profile with agent sets $U$ and $W$ of size $n$ each. Further let $n_1$ (respectively, $n_2$) denote the number of agents from $U$ that are initially matched (respectively, unmatched) under any stable matching of $P$. Analogously, we define $n_3$ and $n_4$ for the set $W$, i.e., $n_3$ (respectively, $n_4$) denotes the number of agents from $W$ that are initially matched (respectively, unmatched) under any stable matching of $P$. We compute these quantities in $O(n^2)$ time using Theorem 2.5. By the definition of matching, it follows that

$$n_1 = n_3 \text{ and } n_2 = n_4.$$

Hence, together with the definition of $\mathsf{n_m}$ and $\mathsf{n_u}$, it follows that

$$n_1 + n_3 = 2n_1 = 2n_3 = \mathsf{n_m} \text{ and } n_2 + n_4 = 2n_2 = 2n_4 = \mathsf{n_u}. \tag{10}$$

Observe that no two initially unmatched agents are acceptable to each other, since, if they were, then they would form a blocking pair. Thus, in each matching that represents a solution to Global-Near+Perf, each initially unmatched agent from $U$ (respectively, $W$) is partnered with an initially matched agent from $W$ (respectively, $U$). Thus, if $n_2 > n_3$ or $n_4 > n_1$, we can immediately return no (or a trivial no-instance). By Equation (10), we obtain that

$$|U| = n_1 + n_2 \leq n_1 + n_3 = \mathsf{n_m} \text{ and } |W| = n_3 + n_4 \leq n_3 + n_1 = \mathsf{n_m}. \tag{11}$$

In other words, we obtain a problem kernel with at most $2\mathsf{n_m}$ agents. Overall, computing the kernel takes $O(n^2)$ time.

To solve Global-Near+Perf in $O(2^{2\mathsf{n_m}^2 \cdot \log(\mathsf{n_m})} \cdot n^2)$ time, observe that by (11) each agent from $U \cup W$ has at most $\mathsf{n_m}$ agents in her preference list, and there are $2\mathsf{n_m}$ agents from $U \cup W$. We hence iterate through all at most $(\mathsf{n_m}!)^{2\mathsf{n_m}}$ target profiles $P'$ that differ from $P$ by at most $\mathsf{d_G}$ swaps. For each of these preference profiles, we check in $O(n^2)$ time whether it admits a stable and perfect matching using Gale and Shapley's algorithm (see Theorem 3.3). It is clear that this is correct. The running-time bound follows, since by Stirling's approximation $\mathsf{n_m}! \leq e\mathsf{n_m}^{\mathsf{n_m}+1/2}e^{-\mathsf{n_m}}$, which is $O(2^{\mathsf{n_m}\log(\mathsf{n_m})})$. □

We conclude this section by remarking that the kernelization approach for Proposition 5.7 cannot be directly adapted to work for the egalitarian case, because not every initially unmatched agent needs to be matched in an optimal egalitarian stable matching.

## 6 CONCLUSION AND OPEN QUESTIONS

In this article, we have introduced and studied a framework describing the strength of the stability of matchings under preferences. Our framework unifies and extends some of the few approaches that already exist in the literature, such as additive $\alpha$-stability [2, 44], $r$-maximal stability [17], and robustness to the errors in inputs [35, 36]. We have demonstrated that all these approaches can be expressed by the same model, where the central idea is to investigate the preference profiles that have bounded distance to the input profile. Thus, we open up a general framework to study questions that have already received attention in the literature.

From a computational point of view, we have shown a somehow counter-intuitive relation between strength of stability and other criteria of social optimality. On the one hand, there exists a polynomial-time algorithm for finding the largest $d$ for which a $d$-robust matching exist (recall that robustness is a stronger concept than classical stability) even if we additionally aim to reach social optimality. On the other hand, if we ask about near stability instead of robustness, the problem becomes computationally hard in many aspects: It is hard to approximate and hard from the point of view of parameterized complexity. Our computational results are summarized in Table 1.

We conclude with some challenges for future research. First, our concept can be extended to other distance measures and to deal with cardinal instead of ordinal preferences.

Second, as we have shown in Section 2.2, a robust matching might not always exist. It would be interesting to characterize the preference profiles for which the robust matchings exist. We partially explore this question in Section 2.2, yet much more research in this direction is needed. When analyzing near stability, however, we focused on discussing its interplay with the egalitarian and the perfectness criteria. We chose those criteria since they feel natural and are often investigated in the literature in the context of two-sided matching [26, 27, 30, 38, 44, 45]. Investigating other criteria such as Pareto efficiency [1, 10, 12] is an important direction for future work.

Third, similarly to the local near stability, we may say that a stable matching is locally $d$-robust if it remains stable for each preference profile where each agent's preference list is no more than

$d$ swaps away from the original one. We conjecture that the idea behind our polynomial-time algorithm for Theorem 3.22 can be adapted to also work for solving local $d$-robustness.

Fourth, for the case where no $d$-robust matchings exist, we may look for a matching that admits the minimum number of blocking pairs [4, 13] in every profile that has swap distance $d$ to the input profile.

Finally, continuing our research in Section 4 where we showed that Robust Matching becomes NP-hard when ties are allowed, our near stability concept can be generalized to the case with ties. Moreover, both robustness and near stability, though introduced for the bipartite variant (Stable Marriage), can be generalized to the non-bipartite variant (Stable Roommates) [11]. We conjecture that our profile characterization framework (Section 3.2) can be adapted to determine $d$-robust matching in Stable Roommates when no ties are present.

Regarding preference restrictions [6], it would be interesting to know whether assuming a special preference structure can help in finding tractable cases for nearly stable matchings.

## ACKNOWLEDGMENTS

The authors thank the anonymous reviewers of the 2019 ACM Conference on Economics and Computation and of the journal *ACM TEAC* for detailed and helpful comments.

## REFERENCES

[1] José Alcalde and Antonio Romero-Medina. 2017. Fair student placement. *Theor. Decis.* 83, 1 (2017), 293–307.
[2] Elliot Anshelevich, Sanmay Das, and Yonatan Naamad. 2013. Anarchy, stability, and utopia: Creating better matchings. *Auton. Agents Multi-agent Syst.* 26, 1 (2013), 120–140.
[3] Haris Aziz, Péter Biró, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. 2020. Stable matching with uncertain linear preferences. *Algorithmica* 82, 5 (2020), 1410–1433.
[4] Péter Biró, David Manlove, and Eric McDermid. 2012. "Almost stable" matchings in the Roommates problem with bounded preference lists. *Theor. Comput. Sci.* 432 (2012), 10–20.
[5] James Boudreau and Vicki Knoblauch. 2013. Preferences and the price of stability in matching markets. *Theor. Decis.* 74, 4 (2013), 565–589.
[6] Robert Bredereck, Jiehua Chen, Ugo Paavo Finnendahl, and Rolf Niedermeier. 2020. Stable roommate narcissistic, single-peaked, and single-crossing preferences. *Auton. Agents Multi-agent Syst.* 34, 2 (2020), 53.
[7] Robert Bredereck, Jiehua Chen, Dušan Knop, Junjie Luo, and Rolf Niedermeier. 2020. Adapting stable matchings to evolving preferences. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*. 1830–1837.
[8] Robert Bredereck, Piotr Faliszewski, Andrzej Kaczmarczyk, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. 2021. Robustness among multiwinner voting rules. *Artif. Intell.* 290 (2021), 103403.
[9] Paul Camion. 1965. Characterization of totally unimodular matrices. *Proc. Amer. Math. Soc.* 16, 5 (1965), 1068–1073.
[10] David Cantala and Szilvia Pàpai. 2014. *Reasonably and Securely Stable Matching*. Technical Report. Mimeo Digital.
[11] Katarína Cechlárová, Ágnes Cseh, and David Manlove. 2019. Selected open problems in matching under preferences. *Bull. EATCS* 2, 128 (2019), 1–26.
[12] Yeon-Koo Che and Olivier Tercieux. 2019. Efficiency and stability in large matching markets. *J. Polit. Econ.* 127, 5 (2019).
[13] Jiehua Chen, Danny Hermelin, Manuel Sorge, and Harel Yedidsion. 2018. How hard is it to satisfy (almost) all roommates? In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP'18)*. 35:1–35:15.
[14] Jiehua Chen, Rolf Niedermeier, and Piotr Skowron. 2018. Stable marriage with multi-modal preferences. In *Proceedings of the 19th ACM Conference on Economics and Computation (ACM EC'18)*. 269–286.
[15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.
[16] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer.
[17] Joanna Drummond and Craig Boutilier. 2013. Elicitation and approximately stable matching with partial preferences. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. 97–105.
[18] Lars Ehlers and Thayler Morrill. 2020. (Il)legal assignments in school choice. *Rev. Econ. Stud.* 87, 4 (2020), 1837–1875.
[19] Haluk I. Ergin. 2002. Efficient resource allocation on the basis of priorities. *Econometrica* 70, 6 (2002), 2489–2497.

[20] Piotr Faliszewski and Jörg Rothe. 2016. Control and bribery in voting. In *Handbook of Computational Social Choice*. Cambridge University Press.
[21] Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer.
[22] David Gale and Lloyd S. Shapley. 1962. College admissions and the stability of marriage. *Amer. Math. Month.* 120, 5 (1962), 386–391.
[23] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.
[24] Begum Genc, Mohamed Siala, Gilles Simonin, and Barry O'Sullivan. 2017. Robust stable marriage. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. 4925–4926.
[25] Begum Genc, Mohamed Siala, Gilles Simonin, and Barry O'Sullivan. 2019. Complexity study for the robust stable marriage problem. *Theor. Comput. Sci.* 775 (2019), 76–92.
[26] Dan Gusfield and Robert W. Irving. 1989. *The Stable Marriage Problem–Structure and Algorithms*. The MIT Press.
[27] R. W. Irving. 2016. Optimal stable marriage. In *Encyclopedia of Algorithms*, Ming-Yang Kao (Ed.). Springer 1470–1473.
[28] Robert W. Irving. 1994. Stable marriage and indifference. *Disc. Appl. Math.* 48, 3 (1994), 261–272.
[29] Robert W. Irving and Paul Leather. 1986. The complexity of counting stable marriages. *SIAM J. Comput.* 15, 3 (1986), 655–667.
[30] Robert W. Irving, Paul Leather, and Dan Gusfield. 1987. An efficient algorithm for the "optimal" stable marriage. *J. ACM* 34, 3 (1987), 532–543.
[31] Varun Kanade, Nikos Leonardos, and Frédéric Magniez. 2016. Stable matching with evolving preferences. In *Proceedings of the Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM'16)*, Vol. 60. 36:1–36:13.
[32] Anna R. Karlin, Shayan Oveis Gharan, and Robbie Weber. 2018. A simply exponential upper bound on the maximum number of stable matchings. In *Proceedings of the 50th ACM Symposium on Theory of Computing (STOC'18)*. 920–925.
[33] Onur Kesten. 2010. School choice with consent. *Quart. J. Econ.* 125, 3 (2010), 1297–1348.
[34] Donald Knuth. 1976. *Mariages Stables*. Les Presses de L'Université de Montréal.
[35] Tung Mai and Vijay V. Vazirani. 2018. Finding stable matchings that are robust to errors in the input. In *Proceedings of the 26th European Symposium on Algorithms (ESA'18)*. 60:1–60:11.
[36] Tung Mai and Vijay V. Vazirani. 2018. *A Generalization of Birkhoff's Theorem for Distributive Lattices, with Applications to Robust Stable Matchings*. Technical Report. arXiv:1804.05537 [cs.DM]. Cornell University.
[37] David Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. 2002. Hard variants of stable marriage. *Theor. Comput. Sci.* 276, 1–2 (2002), 261–279.
[38] David F. Manlove. 2013. *Algorithmics of Matching Under Preferences*. Vol. 2. WorldScientific.
[39] Vijay Menon and Kate Larson. 2018. Robust and approximately stable marriages under partial information. In *Proceedings of the 14th Conference on Web and Internet Economics (WINE'18)*. 341–355.
[40] George A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.* 63, 2 (1956), 81–97.
[41] Shuichi Miyazaki and Kazuya Okamoto. 2019. Jointly stable matchings. *J. Combinat. Optim.* 38, 2 (2019), 646–665.
[42] Rolf Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.
[43] Christos H. Papadimitriou. 2007. The complexity of finding Nash equilibria. In *Algorithmic Game Theory*. Cambridge University Press, 29–51.
[44] Maria Silvia Pini, Francesca Rossi, K. Brent Venable, and Toby Walsh. 2013. Stability, optimality and manipulation in matching problems with weighted preferences. *Algorithms* 6, 4 (2013), 782–804.
[45] Eytan Ronn. 1990. NP-complete stable matching problems. *J. Algor.* 11, 2 (1990), 285–304.
[46] Alvin E. Roth and Marilda A. Oliveira Sotomayor. 1992. *Two-sided Matching: A Study in Game-theoretic Modeling and Analysis*. Cambridge University Press.
[47] Dmitry Shiryaev, Lan Yu, and Edith Elkind. 2013. On elections with robust winners. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*. 415–422.
[48] Peter Troyan, David Delacrétaz, and Andrew Kloosterman. 2020. Essentially stable matchings. *Games Econ. Behav.* 120 (2020), 370–390.

Received July 2019; revised March 2021; accepted April 2021