ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240899

Worst- and Average-Case Robustness of Stable Matchings: (Counting) Complexity and Experiments

Kimon Boehmer[a] **and Niclas Boehmer**[b,*]

[a]Université Paris-Saclay, France
[b]Hasso Plattner Institute, University of Potsdam, Germany

Abstract. Focusing on the bipartite STABLE MARRIAGE problem, we investigate different robustness measures related to stable matchings. We analyze the computational complexity of computing them and analyze their behavior in extensive experiments on synthetic instances. For instance, we examine whether a stable matching is guaranteed to remain stable if a given number of adversarial swaps in the agent's preferences are performed and the probability of stability when applying swaps uniformly at random. Our results reveal that stable matchings in our synthetic data are highly unrobust to adversarial swaps, whereas the average-case view presents a more nuanced and informative picture.

1 Introduction

In two-sided stable matching problems, there are two sets of agents with each agent having preferences over the agents from the other set. The goal is to find a stable matching of agents from one side to agents from the other side, i.e., a matching where no pair of agents prefer each other to their current partner. Since their introduction by Gale and Shapley [30], such problems have been studied extensively in economics and computer science (see the survey of Manlove [43]) and many real-world applications including online dating [36] and the assignment of students to schools [1] or of children to daycare places [38] have been identified. Importantly, in many applications, agents remain matched for a longer period of time, over which their preferences might change as they learn more about their current match and other options (as witnessed, e.g., by students switching colleges or roommates). This observation already motivated different lines of research, for instance, the study of *finding* robust stable matchings, that are, matchings which remain stable even if some changes are performed [21, 41, 32].

We contribute a new perspective on robustness and stable matchings. Instead of computing different types of robust stable matchings, we focus on *quantifying* the robustness of a given stable matching. In addition to making matching markets more transparent and predictable, we see multiple possible use cases for our robustness measures especially for market organizers. They can for instance be used to make an informed decision between different proposed stable matchings or more generally might serve as an additional criterion to decide between different matching algorithms. Moreover, if a matching is detected to be very non-robust, the organizers of the matching market can initiate some countermeasures to prevent the possibly high costs attached to changing a once-implemented matching to reestablish stability. For instance, they can check whether preferences were elicited correctly or ask agents to reevaluate their preferences after assisting them with further guidance or information.

A first measure of the robustness of a stable matching is via the study of destructive bribery [47]: The idea is to compute the minimum number of changes (of a certain type) that need to be applied to the instance such that the matching becomes unstable. This measure gives us a worst-case guarantee on the stability of our matching. However, this measure disregards that changes in the real world will not be adversarial. This is why we additionally study the robustness of stable matchings to random noise: The idea is to compute the probability that the given matching remains stable if we apply a given number k of changes uniformly at random. Observing how quickly this probability decreases with increasing k gives us an estimate for the average-case robustness of the matching. While these two approaches are conceptually different, they are computationally closely related: Computing the probabilities in our second measure can be done by first counting the number of instances where k changes have been applied and the given matching is stable and then dividing this number by the total number of instances at this distance. Consequently, computing our average-case measure requires solving the counting analog of the computational problem underlying our worst-case measure.

Our Contributions. We contribute a novel average-case perspective to the study of robustness and stable matchings and further explore the previously mentioned worst-case analog. In addition to studying the robustness of matchings, we also pioneer the study of the robustness of different local configurations such as whether an agent pair can be included in a stable matching or whether an agent is assigned a partner. We focus on measuring robustness against swaps in the agent's preferences or the deletion of agents.

In Section 3, we analyze the computational complexity of eight decision problems resulting from our two change types and four goals (see Table 1 for an overview). The resulting complexity picture is mixed, as the complexity decisively depends on the type of change and the assessed object. Subsequently, in Section 4, for all variants for which we did not show hardness in the decision setting, we present involved reductions showing the respective counting

* Corresponding Author. Email: niclas.boehmer@hpi.de.

	Matching		Matching (bps)		Pair		Agent	
	decision	counting	decision	counting	decision	counting	decision	counting
SWAP	P (Ob. 1)	#P-h. (Th. 6)	NP-h. (Th. 2)	–	NP-h. (Pr. 3)	–	NP-h. (Pr. 5)	–
DELETE	P (Ob. 1)	#P-h. (†)	NP-h. (Th. 2)	–	?	#P-h. (Th. 7)	P (Ob. 4)	#P-h. (Th. 7)

Table 1. Overview of our results. "Matching (bps)" stands for the problem to create a given number of blocking pairs for a given matching using a limited number of changes. The result marked with (†) follows from Proposition 6 of Boehmer et al. [13].

problem to be #P-hard.[1] The #P-hardness of the counting problems stand in contrast with simple polynomial-time algorithms for the decision variants and require intricate reductions needing new ideas (as counting problems have been rarely studied in the matching under preferences literature). We conclude the section by presenting some approximation algorithms for the counting problems.

In Section 5, customizing the diverse synthetic dataset introduced by Boehmer et al. [17], we perform extensive experiments related to the stability of stable matchings and pairs in case swaps in the agent's preferences are performed, thereby adding a new empirical component to the almost purely theoretical literature on robustness in stable matchings. We find that in our dataset almost all stable matchings can be made unstable by performing a single swap. To measure matching's robustness to random swaps, motivated by our intractability results from Sections 3 and 4, we explore a sampling-based approach building upon the popular Mallows noise model [42]. It turns out that stable matchings generally have a remarkably low robustness to random noise, yet the degree of the non-robustness significantly varies between instances; even within a single instance, different initially stable matchings may have a noticeably different average-case robustness. For instance, matchings produced by the popular Gale-Shapley algorithm tend to be less robust than so-called summed-rank minimizing stable matchings. We also present a heuristic to measure matching's average-case robustness which builds upon the number of pairs that are close to being blocking, and demonstrate that it is of excellent quality in practice. Lastly, we observe that stable pairs are generally much more robust than stable matchings.

The (full) proofs of all statements and many experimental details are deferred to our full version [9]. The code for our experiments can be found at github.com/n-boehmer/robustness-of-matchings.

Related Work. Previous work on the robustness of stable matchings predominantly focused on computing matchings that are guaranteed to remain stable even if a given number of changes are performed [21, 41, 31, 32] or for which stability can be easily reestablished by changing only a few pairs in the matching [34, 33, 35, 29]. In contrast, our work proposes and analyzes different ways to *quantify* the robustness of a given stable matching. As a result, while our paper shares a similar motivation, it is technically quite different.

From a technical perspective, closest to our work are the papers of Aziz et al. [4, 3] and Boehmer et al. [13]. Related to the worst-case robustness of matchings, Boehmer et al. [13] initiated the study of constructive bribery problems in the stable matching literature, i.e., problems asking whether a matching can be made stable by performing a given number of changes. Eiben et al. [25] and Bérczi et al. [7] extended their studies by analyzing related problems connected to ensuring the existence of a stable matching with some desired property. Notably, all three of these papers focused purely on decision problems and did not analyze matching's robustness. Related to the average-case robustness of matchings, Aziz et al. [4, 3] analyzed different problems occurring when agent's preferences are uncertain. For instance, they consider the problem of computing the probability that a given matching is stable assuming that each agent provides a probability distribution over preference lists or the full preference profile is drawn from some probability distribution. Our #MATCHING-SWAP-ROBUSTNESS problem (see Section 2) is related to a special case of the latter problem, where we draw a preference profile uniformly at random from the set of all preference profiles at a given distance from some profile. However, a formal polynomial-time reduction cannot be established, as there are exponentially many profiles in the support of this distribution.

In addition to finding robust stable matchings, there are also multiple other lines of work motivated by the observation that agent's preferences change over time. For instance, in the study of dynamic stable matchings the goal is usually to adapt classic stability notions to dynamic settings [10, 2, 5, 22, 24, 40]. Moreover, various works have studied problems related to minimally adapting stable matchings to reestablish stability after changes occurred [29, 28, 8, 37, 18].

While the idea of using bribery for robustness is new in the context of matching under preferences, there are numerous works on bribery for quantifying robustness in the voting [27, 26, 47, 6, 11, 14], tournament [23, 19] and group identification [15] literature .

2 Preliminaries

For a given set S, let $\mathcal{L}(S)$ denote the set of all strict and complete orders over elements from S.

An instance $\mathcal{I} = (U, W, \mathcal{P})$ of STABLE MARRIAGE (SM) is defined by a set $U = \{m_1, \dots, m_n\}$ of *men* and a set $W = \{w_1, \dots, w_m\}$ of *women*. Each man $m \in U$ is associated with a preference list $\succ_u \in \mathcal{L}(W)$ over women, and each woman $w \in W$ is associated with a preference list $\succ_w \in \mathcal{L}(U)$ over men. The preferences are collected in a *preference profile* $\mathcal{P} = \{\succ_u \in \mathcal{L}(W) \mid u \in U\} \cup \{\succ_w \in \mathcal{L}(U) \mid w \in W\}$. We refer to $A = U \cup W$ as the set of *agents*. For $a \in A$, we call $\succ_a$ the preference list of a and say that a *prefers* agent x over y if $x \succ_a y$. We sometimes write $\succ_a$ as $a : a_1 \succ_a a_2 \succ_a a_3 \succ_a \dots$, where the "…" at the end mean that all other agents of opposite gender follow in some arbitrary order. We write $\mathrm{rk}_b(a)$ for the position that agent a has in the *preference list* $\succ_b$ of agent b, i.e., the number of agents b prefers to a plus one. For an agent a and two agents b and c appearing in $\succ_a$, we define the *distance* between b and c (in the preferences of a) as $|\mathrm{rk}_a(b) - \mathrm{rk}_a(c)|$.

A *matching* M is a set of pairs $\{u, w\}$ with $u \in U$ and $w \in W$ such that each agent appears in at most one pair. If an agent is contained in a pair in M, they are *assigned*; otherwise, they are *unassigned*. For a matching M and an agent a, $M(a)$ is the agent a is matched to in M if a is assigned; otherwise, we set $M(a) := \perp$. A matching is *complete* if no agent is unassigned. A pair $\{u, w\}$ *blocks* a matching M if (i) u prefers w to $M(u)$ or is unassigned, and (ii) w

[1] #P is the counting analog of NP. One consequence of this is that a polynomial-time algorithm for a #P-hard problem implies a polynomial-time algorithm for all problems in NP.

prefers m to $M(w)$ or is unassigned. If a matching does not admit a blocking pair, it is *stable*. If a pair $\{u, w\} \in U \times W$ appears in some stable matching (in $\mathcal{I}$), then $\{u, w\}$ is a *stable pair* (in $\mathcal{I}$). Similarly, if an agent $a \in A$ is assigned in some stable matching (in $\mathcal{I}$), then a is a *stable agent* (in $\mathcal{I}$). Note that by the Rural Hospitals Theorem [45], the set of assigned agents is the same in all stable matchings in $\mathcal{I}$.

A *swap* operation swaps two neighboring agents in the preference list of one agent. The *swap distance* between two instances $\mathcal{I} = (U, W, \mathcal{P})$ and $\mathcal{I}' = (U, W, \mathcal{P}')$ is the minimum number of swaps that are needed to transform $\mathcal{P}$ into $\mathcal{P}'$. A *delete* operation deletes an agent from the agent set and from the preferences of all other agents. Instance $\mathcal{I} = (U, W, \mathcal{P})$ is at *deletion distance* d from $\mathcal{I}' = (U', W', \mathcal{P}')$ if $\mathcal{I}'$ can be obtained from $\mathcal{I}$ by deleting d agents.

We now define our computational problems. In the name of our problems, we first specify the object whose stability we want to measure (i.e., matchings, pairs, or agents) and then the action that we allow (i.e., swaps or deletions).

MATCHING/PAIR/AGENT-SWAP-ROBUSTNESS

Input: An SM instance $\mathcal{I} = (U, W, \mathcal{P})$, a budget $\ell \in \mathbb{N}$, and a matching M/pair $\{m, w\} \in U \times W$/agent $a \in U \cup W$.

Question: Is there an SM instance $\mathcal{I}' = (U, W, \mathcal{P}')$ at swap distance at most ℓ from $\mathcal{I}$ such that M/$\{m, w\}$/a is not stable in $\mathcal{I}'$?

MATCHING/PAIR/AGENT-DELETE-ROBUSTNESS is defined analogously by replacing swap with deletion distance. However, for the matching variant we only require that $M \cap (U' \times W')$ is not stable in the instance $\mathcal{I}' = (U', W', \mathcal{P}')$ resulting from the deletion, for the pair variant we require that neither m nor w get deleted, and for the agent variant that a is not deleted. For all defined decision problems $\mathcal{X}$, in the analogous counting problem $\#\mathcal{X}$, we ask for the number of instances at distance exactly[2] ℓ fulfilling the specified property.

These computational problems can be used to quantify the robustness of matchings, agent pairs, and whether agents are assigned. For instance, MATCHING-SWAP-ROBUSTNESS allows for computing the minimum number of swaps that are needed to make a given matching unstable (in fact, it is easy to see that any stable matching and pair can be made unstable by using at most $n - 1$ swaps [9]). Similarly, #MATCHING-SWAP-ROBUSTNESS can be used to compute the probability that a given matching is stable in case k swaps are performed uniformly at random, by taking the answer to the counting problem and dividing it by the number of instances at swap distance k. The latter quantity can be computed in polynomial time using a dynamic program [9].

Arguably, maintaing "perfect" stability even after preferences change is a quite strong requirement due to the binary nature of this criterion. This motivates us to quantify the "degree of instability" after changes are performed. We consider the number of pairs that block a matching as a measure for this. To compute this, we face the problem of calculating the (maximum) number of pairs that block a matching when a certain number of changes are performed:

BLOCKING PAIRS-SWAP [DELETE]-ROBUSTNESS

Input: An SM instance $\mathcal{I} = (U, W, \mathcal{P})$, a budget $\ell \in \mathbb{N}$, a matching M, and an integer $b \in \mathbb{N}$.

Question: Is there an SM instance $\mathcal{I}' = (U', W', \mathcal{P}')$ at swap [deletion] distance at most ℓ from $\mathcal{I}$ such that M [$M \cap (U' \times W')$] is blocked by at least b pairs in $\mathcal{I}'$?

Solving the counting version of this problem would allow us to compute the expected "degree of instability" as the expected number of pairs blocking a matching after a given number of changes are performed uniformly at random.

[2] The exact constraint here is only for presentation purposes. In fact, the exact and at most variants of the problem can be Turing reduced to each other.

3 Complexity of Decision Variants

We analyze the complexity of our decision problems starting with matchings, then pairs, and lastly agents.

Stable Matchings. Making a given matching unstable is algorithmically straightforward. For swap, it suffices to iterate over all pairs of agents $\{u, w\} \in U \times W$ that are currently not matched to each other, compute the minimum number of swaps to make this pair blocking (by swapping down $M(u)$ after w in $\succ_u$ and $M(w)$ after u in $\succ_w$), and return the minimum. For delete, we can always make a matching unstable by deleting one woman and one man, as their partners form blocking pairs with each other. Deleting one agent is sufficient in case not all agents are matched to their top choice.

Observation 1. MATCHING-SWAP/DELETE-ROBUSTNESS can be solved in $\mathcal{O}((n + m)^2)$ time.

If we want to modify the instance to create not only one but a certain number of blocking pairs, then the problem becomes NP-hard for both swap and delete. The reason for this is that there can be synergy effects when creating two blocking pairs: Making a pair blocking might become cheaper after we have already made another pair blocking by swapping down an agent's partner in their preferences. This effect allows us to devise reductions from CLIQUE and INDEPENDENT SET, respectively. In fact, the hardness both holds for the case where we ask for exactly and for at least b blocking pairs.

Theorem 2. BLOCKING PAIRS-SWAP/DELETE-ROBUSTNESS *is NP-complete.*

Stable Pairs. Turning to stable pairs, the simple algorithms for stable matchings can no longer be applied, as there is no unique straightforward way to make a pair unstable. In fact, by reducing from the NP-hard constructive problem to make a given pair stable [13], we establish the NP-hardness of PAIR-SWAP-ROBUSTNESS.

Proposition 3. PAIR-SWAP-ROBUSTNESS *is NP-complete.*

Notably, the analogous CONSTRUCTIVE-EXISTS-DELETE problem of making a given pair stable by deleting some agents was shown to be polynomial-time solvable by Boehmer et al. [13]. However, there seems to be no easy possibility to extend their algorithm to solve our PAIR-DELETE-ROBUSTNESS problem. Settling the problem's complexity remains an intriguing open question.

Stable Agents. Turning to the problem of making a given agent unstable, we observe the first difference between swap and delete. For delete, it is possible to reduce the problem to the polynomial-time solvable CONSTRUCTIVE-EXISTS-DELETE problem: Assuming that a woman w^* should be made unstable, we add a man m^* that ranks w^* first followed by all other women in an arbitrary ordering and add m^* at the end of the preferences of all women. It is easy to see that stable matchings where w^* is unmatched in the original instance correspond to matchings including $\{m^*, w^*\}$ in the constructed instance. Accordingly, it suffices to compute the number of deletions needed to make $\{w^*, m^*\}$ a stable pair:

Observation 4. AGENT-DELETE-ROBUSTNESS can be solved in $\mathcal{O}(n \cdot m)$ time.

For swap, we establish hardness by reducing from one of the NP-hard constructive bribery problems studied by Boehmer et al. [13]:

Proposition 5. AGENT-SWAP-ROBUSTNESS *is NP-complete.*

4 Complexity of Counting Variants

For all decision problems for which we have proven NP-hardness, their counting variants are naturally also computationally expensive to solve (in particular, at least as hard as the decision variants). We prove the #P-hardness of all remaining counting problems in this section, starting with the robustness of stable matchings to swaps in the preferences. Recall that we have observed in Observation 1 a very simple algorithm for the decision problem, yet the counting version turns out to be computationally intractable:

Theorem 6. #MATCHING-SWAP-ROBUSTNESS *is #P-hard.*

Proof Sketch. We reduce from the #P-hard #BIPARTITE 2-SAT WITH NO NEGATIONS problem [44], where we are given a set $V = U \cup Z$ of variables and a set $C \subseteq U \times Z$ of clauses, i.e., the formula does not contain any negative literals. The task is to count the number of truth assignments of the variables from V such that each clause contains a fulfilled literal.

In our reduction, for each positive and negative literal, we add a literal man and a literal woman that are matched to each other in the given matching. Moreover, we add a set of $|C|$ dummy men and women. The agent set is

$$U' = \{m_v, m_{\bar{v}} \mid v \in V\} \cup \{m_i^d \mid i \in [|C|]\}$$
$$W' = \{w_v, w_{\bar{v}}, \mid v \in V\} \cup \{w_i^d \mid i \in [|C|]\}$$

and the designated matching is

$$M := \{\{m_v, w_v\}, \{m_{\bar{v}}, w_{\bar{v}}\} \mid v \in V\} \cup \{\{m_i^d, w_i^d\} \mid i \in [|C|]\}.$$

We now describe the preferences of the agents, focusing on agents corresponding to variables from U first. We construct the preferences so that matching M is not stable in the initial instance: Specifically, for each variable $u \in U$, m_u and $w_{\bar{u}}$ form a blocking pair, which implies that one of the two needs to modify their preferences to resolve the pair. If we modify the preferences of m_u, then this means that we set u to true, while modifying $w_{\bar{u}}$ implies setting u to false. To ensure that the induced variable assignment fulfills all clauses, for each clause $\{p, q\} \in C$, we let $\{m_p, w_q\}$ form a blocking pair, implying that one of the involved literal agents needs to modify its preferences. Combining these ideas, the preferences are as follows. Consider a variable $u \in U$ and let $p_1, \dots, p_{c(u)} \in Z$ be all variables that appear together with u in a clause. The preferences for the corresponding agents are as follows:

$$m_u : w_{\bar{u}} \succ w_{p_1} \succ \dots \succ w_{p_{c(u)}} \succ w_1^d \succ \dots \succ w_{|C|-c(u)}^d \succ w_u \succ \dots$$
$$w_{\bar{u}} : m_u \succ m_1^d \succ \dots \succ m_{|C|}^d \succ m_{\bar{u}} \succ \dots$$
$$w_u : m_u \succ \dots \qquad m_{\bar{u}} : w_{\bar{u}} \succ \dots$$

For $z \in Z$, the preferences are constructed analogously where the roles of men and women are reversed. The dummy men have preferences $m_i^d : w_i^d \succ \dots$ and the dummy women have preferences $w_i^d : m_i^d \succ \dots$ for $i \in [|C|]$.

Now, a swap budget of $\ell := |V|(|C| + 1)$ is exactly sufficient to resolve the blocking pair for each variable by modifying the preferences of one of the two literal agents by executing $|C| + 1$ swaps to swap their partner in M in the first position. Doing so, we also need to resolve the blocking pairs induced by the clauses, which ensures that the induced variable assignment satisfies all clauses. However, this is not sufficient to establish a one-to-one correspondence between the satisfying assignments for the given 2-SAT formula and preference profiles at swap distance exactly ℓ from $\mathcal{I}'$ where M is stable. In fact, if a literal is not contained in any clause, there are two ways to modify the corresponding agent within the given budget by either swapping the designated partner in M up or the literal agent forming a blocking pair down. This results in a difficult-to-quantify blow-up in the number of solutions. To account for this, in the full construction and proof of correctness, we alter the construction by adding a second copy of the instance [9]. □

It remains to consider the problem of counting the number of agent sets whose deletion makes a given agent or pair stable. In the decision world, we proved that the former problem is polynomial-time solvable while we were unable to settle the complexity of the latter problem. In our most involved construction, we prove that both variants are #P-hard in their counting version:

Theorem 7. #PAIR/AGENT-DELETE-ROBUSTNESS *is #P-hard.*

Proof Sketch (Agent). For an instance $\mathcal{I}$, we denote by $\#\text{AD}(\mathcal{I})$ the number of solutions for this instance. To show #P-hardness for #AGENT-DELETE, we will give a Turing reduction from the #P-hard #EDGE COVER problem, where we are given a graph $G = (V, E)$ and an integer $k \in \mathbb{N}$ and want to compute the number of subsets of edges $E' \subseteq E$ with $|E'| = k$ and $\bigcup_{e \in E'} e = V$ [20]. In this proof, we let $n := |V|$, $m := |E|$, and $N := m + 1$.

We start by defining the #AGENT-DELETE instances that we will give as input to our oracle. To this end, given a graph G, let $\mathcal{I}_{i,j}^G$ be the #AGENT-DELETE instance constructed as follows. We set the deletion budget to $j + i \cdot N$. The agent set consists of *edge men* $\{m_e \mid e \in E\}$, *vertex men* $\{m_v^q \mid v \in V, q \in [N]\}$, *extra men* $\{m_p^* \mid p \in [i]\}$ and *vertex women* $\{w_v \mid v \in V\}$. The designated agent is m_i^*.

The preferences of an edge men m_e with $e = \{u, v\} \in E$ are: $m_e : w_u \succ w_v \succ \dots$. Further, for some $v \in V$, the preferences of vertex women w_v are

$$w_v : m_v^1 \succ \dots \succ m_v^N \succ m_{e_1} \succ \dots \succ m_{e_{\deg(v)}} \succ m_1^* \succ \dots \succ m_i^* \succ \dots,$$

where $e_1, \dots, e_{\deg(v)}$ are the edges incident to v. The extra men have arbitrary preferences, and the vertex men m_v^q rank the corresponding vertex woman w_v in the first position.

To give an intuition for how the solutions to this instance are connected to edge covers of G, consider the case $i = 1$ with designated agent m_1^* and budget $j + N$. Assume that we are interested in edge covers of size $m - j$. Given a size-j edge subset $E' \subseteq E$, we delete all edge men that do not correspond to an edge from E'. If E' is an edge cover, then m_1^* cannot be made stable by deleting N additional agents: For each vertex woman w_v, there is one edge man that was not deleted plus the N corresponding vertex men that w_v prefers to m_1^*. If, however, E' is not an edge cover, then there is a vertex $v \in V$ such that all edge men corresponding to incident edges are deleted. After deleting all vertex men m_v^q for $q \in [N]$, w_v can be matched to our designated agent m_1^* in a stable matching, since all vertex men and edge men that w_v prefers to m_1^* were deleted. Thus, edge sets that are not edge covers in G correspond to solutions to our constructed instance of the above-described form. However, one problem with this idea is that the correspondence is not one-to-one, as in case an edge set E' does not cover multiple vertices, we have multiple ways to ensure that m_1^* gets matched. To deal with this issue, for each non-edge cover E', we need to know how many vertices are not covered to bound the number of corresponding solutions. More formally, we can establish a connection between solutions of the constructed instance and *i-vertex-isolating sets*, that are edge sets $E' \subseteq E$ such that after deleting all edges in E' there are exactly i vertices that have no incident edges. The crucial ingredient of our Turing reduction is the following lemma whose proof appears in our full version [9]:

Lemma 8. *Let $G = (V, E)$ be a graph, $i \in [n]$, and $j \in [m]$:*

$$\#\mathrm{AD}(\mathcal{I}^G_{i,j}) = \sum_{j'=0}^{j} \sum_{i'=i}^{n} \left(\binom{(N+1)(n-i)}{j-j'} \cdot \binom{i'}{i} |\mathcal{E}^{j'}_{i'}|\right) + D_{i,j},$$

$\mathcal{E}^j_i$ is the set of all i-vertex-isolating sets of size j in G and $D_{i,j}$ is the number of solutions to $\mathcal{I}^G_{i,j}$ that delete at least one extra man.

After showing that $D_{i,j}$ can be computed via a dynamic program that queries the oracle for #AGENT-DELETE with altered versions of $\mathcal{I}^G_{i,j}$, we can use the above lemma to compute the number of i-vertex isolating sets using another dynamic program with oracle calls. From this, the number of size-k edge covers can be computed easily. □

In light of the computational hardness results from this and the previous section, we seek an algorithm to approximate the probability that a matching/pair is stable after k random changes are performed. It turns out that using Hoeffding's inequality, we can establish the effectiveness of a simple Monte-Carlo algorithm that samples profiles at distance k uniformly at random and records the fraction of these profiles in which the matching/pair is stable:

Proposition 9. *Given $\varepsilon, \delta > 0$, $\ell \in \mathbb{N}$, and SM instance $\mathcal{I}$, there is a polynomial-time algorithm that computes for a given matching M an estimate p of the probability that M is stable at profiles at swap (or deletion) distance ℓ so that $p \in [p^* - \varepsilon, p^* + \varepsilon]$ with probability $1 - \delta$, where p^* is the correct probability. The statement also applies to pairs and agents.*

Additionally, for the Matching-Swap setting, i.e. for counting the number of preference profiles at some swap distance from a given SM instance such that a given matching is unstable, we can find an $n^2 - n$-approximation by counting all profiles where a specific pair is blocking, and summing over all pairs (by that possibly overcounting the number of profiles by a factor of $n^2 - n$). We can estimate the average factor by which we overcount and obtain a fully polynomial-time randomized approximation scheme (FPRAS):

Theorem 10. *There is a FPRAS for counting the number of preference profiles at swap distance exactly ℓ from $\mathcal{I}$ where M is unstable.*

5 Experiments

In this section, we analyze the robustness of stable matchings (Section 5.1) and pairs (Section 5.2) against random or adversarial swaps in a diverse set of synthetic instances.[3] In our experiments, we sometimes measure the linear correlation between two quantities using the *Pearson Correlation Coefficient* (PCC) which is 1 in case of a perfect positive linear correlation, -1 in case of a perfect negative linear correlation and 0 if there is no linear correlation.

[3] To maintain focus, we do not consider delete operations, as we believe that swaps are the more common changes in practice. Moreover, we do not consider stable agents, as this would require the generation of instances where the two sides have a different size, which would make the setup more complicated.

Computing Stability Probabilities. The sampling algorithm described at the end of Section 4 for approximating the probability that a given matching or pair remains stable after k changes are performed requires sampling profiles at some given swap distance; a time-consuming process already for 30 agents [11]. This is why, following the works of Baumeister and Hogrebe [6] and Boehmer et al. [14] in the context of elections, we make use of the popular Mallows model [42] for adding noise to preference lists. The Mallows model is parameterized by a central preference list $\succ \in \mathcal{L}(A)$ and a dispersion parameter $\phi \in [0, 1]$, and samples a preference list $\succ' \in \mathcal{L}(A)$ with probability proportional to $\phi^{\kappa(\succ,\succ')}$, where $\kappa(\succ, \succ')$ is the swap distance between $\succ$ and $\succ'$. Note that the dispersion parameter controls the level of noise added to the central preference list: For $\phi = 0$ only the central preference list is sampled, whereas for $\phi = 1$ all lists are sampled with the same probability. However, as argued by Boehmer et al. [12] and Boehmer et al. [16] the connection between the value of ϕ and the expected number of swaps applied to the preference list is non-linear. Thus, to make our results easier to interpret, we use the normalized Mallows model introduced by Boehmer et al. [12]: Here, we specify a normalized dispersion parameter norm-$\phi \in [0, 1]$, which is internally converted to a value of the dispersion parameter ϕ such that the expected swap distance between a sampled preference list and the central one in the resulting Mallows model is a norm-ϕ fraction of the maximum possible one.

To measure the robustness of an SM instance $\mathcal{I} = (U, W, \mathcal{P})$ in our experiments, we fix a value of the norm-ϕ parameter, and for each $a \in A$, we draw a preference list $\succ'_a$ from the Mallows model with this norm-ϕ value and $\succ_a$ as the central preference list. We refer to the *stability probability* of a matching or pair at norm-ϕ as the probability of the matching or pair being stable when executing this procedure.[4] To approximate the stability probabilities in the resulting instance $\mathcal{I}' := (U, W, \mathcal{P}' := \{\succ'_a \mid a \in A\})$, we record whether a certain matching (or pair) is stable in $\mathcal{I}'$ and repeat this process 1000 times to get a Monte-Carlo-style approximation. Similar in spirit to the works of Boehmer et al. [11, 14], we sometimes assess the robustness of a matching or pair by the 50%-(stability)-threshold, which is the smallest examined value of norm-ϕ for which the estimated probability of being stable drops below 50%.

Dataset. Due to the lack of publicly available real-world data, we use the large diverse synthetic dataset created by Boehmer et al. [17] focusing on instances with 50 men and women. To generate the data, they used 10 different synthetic models (see our full version [9] for descriptions) including the Impartial Culture (IC) model, where each agent samples its preference list uniformly at random, and Euclidean models, where agents are uniformly at random sampled points in the Euclidean space and rank each other depending on their distance. Their dataset also contains three "extreme" instances: (i) The *Identity* instance, where all agents have the same preferences, (ii) the *Mutual Agreement* instance, where u ranks w in position i if w ranks u in position i, and (iii) the *Mutual Disagreement* instance, where u ranks w in position i if w ranks u in position $50 - i$.

As in instances from the dataset of Boehmer et al. [17] the worst- and average-case robustness of stable matchings is quite low, we add a *Robust* extreme instance and instances sampled form a new *Mallows-Robust* model. In the *Robust* instance, for $i \in [50]$, the preference list of man u_i is $u_i : w_i \succ w_{i+1} \succ \cdots \succ w_n \succ w_1 \succ \cdots \succ w_{i-1}$ and the preference list of woman w_i is $w_i : u_i \succ u_{i+1} \succ$

[4] In our full version [9], we prove that our counting problems can be reduced to computing the stability probability under the described model, implying that the latter task is also computationally intractable.

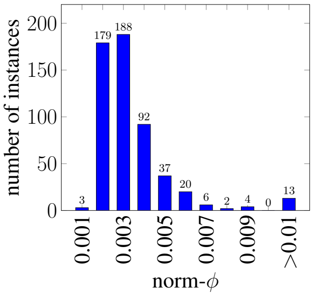

Figure 1. Distribution of the 50%-threshold of men-optimal matching.

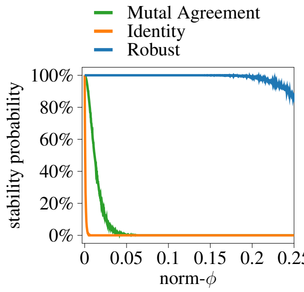

Figure 2. Stability probability of men-optimal matchings in three different instances for varying levels of noise.

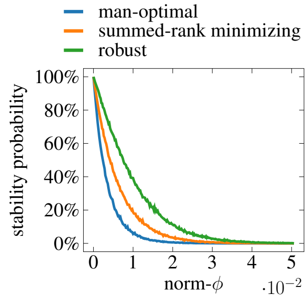

Figure 3. Stability probability of three different matchings in instance sampled from IC model.

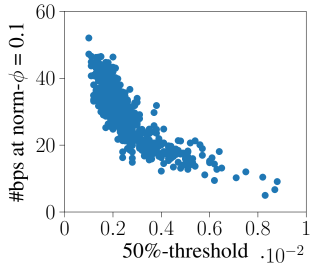

Figure 4. Correlation between 50%-threshold and average number of blocking pairs of men-optimal matching at norm-$\phi = 0.1$.

$\cdots \succ u_n \succ u_1 \succ \cdots \succ u_{i-1}$. In this instance, 50 swaps are needed to make the matching $\{\{u_i, w_i\} \mid i \in [n]\}$ unstable. The Mallows-Robust model is parameterized by the normalized dispersion parameter norm-$\phi \in [0, 1]$. Each agent generates their preferences by making a sample from the Mallows model with their preference list from the Robust instance as the central preference list and parameter norm-ϕ. We sample 20 instances for each norm-$\phi \in \{0.2, 0.4, 0.6, 0.8\}$. In total, our dataset contains 544 instances.

5.1 *Robustness of Stable Matchings*

In this section, we analyze the robustness of men-optimal and other types of stable matchings as well as a simple heuristic for matching's average-case robustness.

Men-Optimal Matching. We start by considering the robustness of men-optimal matchings computed by the popular Gale-Shapley algorithm, which turn out to be very non-robust to adversarial swaps: In almost all instances from our dataset, the men-optimal matching can be made unstable by a single swap, implying that the worst-case robustness is seemingly not capable to meaningfully distinguish the robustness of matchings in practice.

Turning to the robustness against random noise, Figure 1 depicts the distribution of the 50%-thresholds. In most instances, the 50%-threshold of the men-optimal matching is between 0.002 and 0.003. This corresponds to performing, on average, between 1.225 and 1.8375 swaps in each preference list. Considering that most swaps do not involve an agent's partner and thus do not influence the stability of the matching, this number is remarkably low. The explanation for this behavior is that in our instances there are in fact many agent pairs that only need one swap to become blocking. In our full version [9], we also examine how the 50%-threshold depends on the model from which the instance was sampled. Among others, we find that in instances sampled from the IC and Mallows-Robust model the 50%-threshold tends to be larger, whereas for instances sampled from Euclidean models it is lower. These findings highlight that while matchings are also in general quite non-robust against random noise, the picture is more nuanced than for adversarial noise.

Taking a closer look, in Figure 2 we depict how the stability probability behaves when we increase the added noise (aka. the chosen norm-ϕ parameter) for men-optimal matchings in three of the extreme instances, which indeed behave extreme here: The Identity instance is the least robust instance in our dataset. Its stability probability drops down very quickly and is already below 50% at norm-$\phi = 0.0006$ (which corresponds to making an expected number of 36 swaps in the full instance) and below 10% at norm-$\phi = 0.002$. This can be easily explained by the fact that in this instance as soon as an agent swaps down its current partner in its preferences a blocking pair is formed. In contrast, the Mutual Agreement instance is the most robust instance that is not sampled from the Mallows-Robust model. Its stability probability drops down slower, as in this matching all agents are matched to their most preferred agent. Lastly, the Robust instance has the highest stability probability in all our instances and we see that only at norm-$\phi = 0.2$ does the matching's probability of being stable drop below 99%. Moreover, we observe that for all instances (as for the three depicted ones) the stability probability of the men-optimal matching decreases monotonically (up to sampling errors) with increasing norm-ϕ.

Robustness Heuristic. Next, we examine a heuristic to estimate the 50%-threshold of a matching, which can be used as a fast approximation and sheds further light on what makes a matching robust to random noise. For this, for a matching M and a man-woman pair $\{m, w\}$, we let $\beta(m, w)$ be the minimum number of swaps needed to make $\{m, w\}$ blocking and $\#\beta(M, k)$ the number of man-woman pairs $\{m, w\} \in U \times W$ with $\beta(m, w) = k$. Intuitively, if $\#\beta(M, 1)$ is large, then the probability that we create a blocking pair after making a few random swaps is high, as there are many swaps that when executed immediately create a blocking pair. Looking at the number $\#\beta(M, 2)$ of pairs for which two swaps are needed, those are slightly less likely to become blocking (as both necessary swaps would need to be performed). If we increase the k further, the likelihood of these pairs becoming blocking decreases exponentially. Accordingly, we derive the blocking pair proximity, which is small if matchings are robust:

Definition 11. *For some* $d \in [n]$*, we define the* blocking pair proximity *of a matching* M *as* $\pi(M) := \log_n \sum_{k=1}^{d} n^{d-k} \cdot \#\beta(M, k)$.

To avoid large numbers, in our experiments, we set $d = 5$, implying that we only examine pairs that need at most 5 swaps to become blocking (those pairs have the strongest effect on the blocking pair proximity values). We find that the blocking pair proximity is indeed a very good indicator for the 50%-threshold of a matching: For the men-optimal matchings from our dataset the correlation between the two measures is with -0.965 very strong [46]. Consequently, in practice, the average-case robustness of a stable matching seems to boil down to how far agent pairs are away from being blocking and can be well approximated using the much faster-to-compute blocking pair proximity measure.

Other Stable Matchings. Stable matchings are often not unique, which motivates the question of whether different initially stable matchings have a different robustness. To this end, we consider two alternative types of stable matchings. First, the *summed-rank minimizing stable matching* [39], which minimizes the average rank of an agent's partner in their preference list and can be computed in polynomial time [48]. Second, the so-called *robust stable matching*, which is a stable matching minimizing the expression from Theorem 11 (see our full version [9] for details on how to compute this matching). In 176 of our instances the three matchings are identical.

In terms of their worst-case robustness, the three types of stable matching behave almost identically in all instances. In contrast, for the average-case robustness, there is a larger difference: In case the three matchings differ from each other, oftentimes the robust and summed-rank minimizing matching have a clearly higher 50%-threshold than the men-optimal matching. In particular, for 49 instances, the 50%-threshold of the robust and summed-rank minimizing stable matching is more than twice as high as for the men-optimal one; however, there are also few instances where the men-optimal stable matching slightly outperforms the summed-rank minimizing one. The average difference between the 50%-threshold of the robust and summed-rank minimizing matching is with 0.0004 less pronounced. Nevertheless, within one instance the stability probability of all three matchings typically behaves quite similarly when adding noise (see Figure 3 for an example). All in all, we can conclude that from a robustness perspective it is recommendable to use the summed-rank minimizing stable matching instead of the men-optimal one (the small gain from using the robust stable matching instead arguably does not justify the increase in computation time).

Average Number of Blocking Pairs. As discussed in Section 2, enforcing that an initially stable matching remains stable after changes have been performed might be viewed as a quite strict requirement. This motivated the study of the BLOCKING PAIRS-ROBUSTNESS problems and motivates us in this section to analyze the expected number of pairs by which a given matching is blocked when changes are performed. For this, we again make use of the Mallows model and compute the average number of pairs blocking a given matching when applying the Mallows model with a given norm-ϕ parameter to all preference lists. We find that the average number of pairs blocking a matching is highly correlated with the matching's 50%-threshold: In Figure 4, each point corresponds to the men-optimal matching in one of our instances with the x-axis showing the matching's 50%-threshold and the y-axis showing the average number of pairs blocking the matching at norm-$\phi = 0.1$ (plots for other values show similar trends). We see a clear correlation between the two measures, but also clear differences on the instance-level, e.g., a matching with 50%-threshold 0.02 might be blocked by between 23 and 44 pairs on average. The general connection between the measures indicates that the 50%-threshold remains informative beyond the focus on binary stability, yet the average number of blocking pairs allows for a more nuanced picture. We refer to our full version [9] for further details, e.g., we find that within one instance, the average number of blocking pairs typically grows linearly with increasing value of norm-ϕ (a clearly different behavior compared to the stability probability of matchings).

Further Experiments. In our full version [9], we analyze the dependence of our results on the number of agents, observing that the 50%-threshold decreases with an increasing number of agents. We also check whether matchings that are not initially stable can have a non-negligible stability probability in case some noise is applied. In most instances, we were unable to find such matchings.

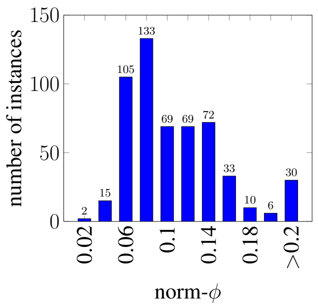

Figure 5. Distribution of the average 50%-threshold of stable pairs in each instance (rounded up to a multiple of 0.02).

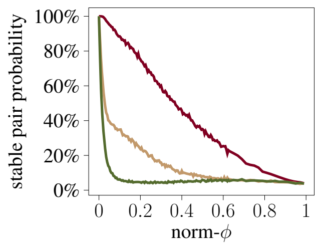

Figure 6. Stability probability of three pairs in instance sampled from a Euclidean model.

5.2 Stable Pairs

We also analyze the robustness of stable pairs in our instances. Analgous as for matchings, we define the 50%-threshold of a pair as the smallest value of norm-ϕ so that the probability of the pair being stable is below 50%. Comparing the 50%-thresholds of initially stable matchings to the 50%-thresholds of initially stable pairs, it turns out that pairs are in general much more stable than matchings. In particular, pairs oftentimes have a 50%-threshold above norm-$\phi = 0.08$: In Figure 5, we show the distribution of instances' average 50%-thresholds computed by taking the average 50%-threshold of pairs initially stable in the instance. We observe that for 422 out of our 544 instances the average 50%-threshold is above norm-$\phi = 0.08$. Note that this observation is quite intuitive, as creating a single blocking pair is sufficient to make a matching unstable. In contrast, a certain pair in this matching can still continue to be stable in other stable matchings in the instance, as oftentimes not the full matching needs to be replaced to reestablish stability.

However, there are additional differences between pairs and matchings. Recall that we have argued above that different stable matchings in one instances (cf. Figure 3) are often similarly robust to changes. In contrast, the difference between pairs can be more pronounced: As an example, in Figure 6 we see how the stability probability of three pairs in an instance sampled from a Euclidean model develops when adding more and more noise to the preference list. It is easy to think of extremely stable pairs, for instance, pairs that rank each other on the first place and are ranked last by every other agent. Our experiments indicate that such drastic examples do appear in our dataset (see [9] for details).

6 Conclusion

We have conducted an algorithmic and experimental study of the average-case and worst-case robustness of stable matchings, pairs, and agents. For future work, regarding the complexity part, settling the complexity of PAIR-DELETE-ROBUSTNESS is the most pressing open question. Moreover, it would also be interesting to examine the parameterized complexity of our problems. For instance, it is open whether our counting problems are #W[1]-hard when parameterized by the examined distance. For future experimental work, it would be interesting to run the experiments on data from other sources to further confirm our findings of the non-robustness of stable matchings in practice.

References

[1] A. Abdulkadiroğlu, P. A. Pathak, A. E. Roth, and T. Sönmez. The boston public school match. *Am. Econ. Rev.*, 95(2):368–371, 2005.

[2] M. Akbarpour, S. Li, and S. O. Gharan. Thickness and information in dynamic matching markets. *J. Political Econ.*, 128(3):783–815, 2020.

[3] H. Aziz, P. Biró, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable matching with uncertain linear preferences. *Algorithmica*, 82(5): 1410–1433, 2020.

[4] H. Aziz, P. Biró, T. Fleiner, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable matching with uncertain pairwise preferences. *Theor. Comput. Sci.*, 909:1–11, 2022.

[5] M. Baccara, S. Lee, and L. Yariv. Optimal dynamic matching. *Theor. Econ.*, 15(3):1221–1278, 2020.

[6] D. Baumeister and T. Hogrebe. On the complexity of predicting election outcomes and estimating their robustness. *SN Comput. Sci.*, 4(4):362, 2023.

[7] K. Bérczi, G. Csáji, and T. Király. Manipulating the outcome of stable matching and roommates problems. *CoRR*, abs/2204.13485, 2022.

[8] S. Bhattacharya, M. Hoefer, C. Huang, T. Kavitha, and L. Wagner. Maintaining near-popular matchings. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP '15)*, pages 504–515. Springer, 2015.

[9] K. Boehmer and N. Boehmer. Worst- and average-case robustness of stable matchings: (counting) complexity and experiments. *CoRR*, abs/2408.09160, 2024.

[10] N. Boehmer and R. Niedermeier. Broadening the research agenda for computational social choice: Multiple preference profiles and multiple solutions. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '21)*, pages 1–5. ACM, 2021.

[11] N. Boehmer, R. Bredereck, P. Faliszewski, and R. Niedermeier. Winner robustness via swap- and shift-bribery: Parameterized counting complexity and experiments. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI '21)*, pages 52–58. ijcai.org, 2021.

[12] N. Boehmer, R. Bredereck, P. Faliszewski, R. Niedermeier, and S. Szufa. Putting a compass on the map of elections. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI '21)*, pages 59–65. ijcai.org, 2021.

[13] N. Boehmer, R. Bredereck, K. Heeger, and R. Niedermeier. Bribery and control in stable marriage. *J. Artif. Intell. Res.*, 71:993–1048, 2021.

[14] N. Boehmer, R. Bredereck, P. Faliszewski, and R. Niedermeier. A quantitative and qualitative analysis of the robustness of (real-world) election winners. In *Proceedings of the Second ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO '22)*, pages 7:1–7:10. ACM, 2022.

[15] N. Boehmer, R. Bredereck, D. Knop, and J. Luo. Fine-grained view on bribery for group identification. *Auton. Agents Multi Agent Syst.*, 37(1): 21, 2023.

[16] N. Boehmer, P. Faliszewski, and S. Kraiczy. Properties of the Mallows model depending on the number of alternatives: A warning for an experimentalist. In *Proceedings of the Fortieth International Conference on Machine Learning (ICML '23)*, pages 2689–2711. PMLR, 2023.

[17] N. Boehmer, K. Heeger, and S. Szufa. A map of diverse synthetic stable roommates instances. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*, pages 1003–1011. ACM, 2023.

[18] R. Bredereck, J. Chen, D. Knop, J. Luo, and R. Niedermeier. Adapting stable matchings to evolving preferences. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI '20)*, pages 1830–1837. AAAI Press, 2020.

[19] M. Brill, U. Schmidt-Kraepelin, and W. Suksompong. Margin of victory for tournament solutions. *Artif. Intell.*, 302:103600, 2022.

[20] R. Bubley and M. E. Dyer. Graph orientations with no sink and an approximation for a hard case of #sat. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '97)*, pages 248–257. ACM/SIAM, 1997.

[21] J. Chen, P. Skowron, and M. Sorge. Matchings under preferences: Strength of stability and tradeoffs. *ACM Trans. Economics and Comput.*, 9(4):20:1–20:55, 2021.

[22] E. Damiano and R. Lam. Stability in dynamic matching markets. *Games Econ. Behav.*, 52(1):34–53, 2005.

[23] M. Döring and J. Peters. Margin of victory for weighted tournament solutions. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*, pages 1716–1724. ACM, 2023.

[24] L. Doval. Dynamically stable matching. *Theor. Econ.*, 17(2):687–724, 2022.

[25] E. Eiben, G. Z. Gutin, P. R. Neary, C. Rambaud, M. Wahlström, and A. Yeo. Preference swaps for the stable matching problem. *Theor. Comput. Sci.*, 940:222–230, 2023.

[26] P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 146–168. Cambridge University Press, 2016.

[27] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. How hard is bribery in elections? *J. Artif. Intell. Res.*, 35:485–532, 2009.

[28] I. Feigenbaum, Y. Kanoria, I. Lo, and J. Sethuraman. Dynamic matching in school choice: Efficient seat reallocation after late cancellations. *Manag. Sci.*, 66(11):5341–5361, 2020.

[29] K. Gajulapalli, J. A. Liu, T. Mai, and V. V. Vazirani. Stability-preserving, time-efficient mechanisms for school choice in two rounds. In *Proceedings of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '20)*, pages 21:1–21:15. Schloss Dagstuhl, 2020.

[30] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[31] R. R. Gangam, T. Mai, N. Raju, and V. V. Vazirani. A structural and algorithmic study of stable matching lattices of "nearby" instances, with applications. In *Proceedings of the 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '22)*, pages 19:1–19:20. Schloss Dagstuhl, 2022.

[32] R. R. Gangam, T. Mai, N. Raju, and V. V. Vazirani. The stable matching lattice under changed preferences, and associated algorithms. *CoRR*, abs/2304.02590, 2023.

[33] B. Genc, M. Siala, B. O'Sullivan, and G. Simonin. Robust stable marriage. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI '17)*, pages 4925–4926. AAAI Press, 2017.

[34] B. Genc, M. Siala, B. O'Sullivan, and G. Simonin. Finding robust solutions to stable marriage. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI '17)*, pages 631–637. ijcai.org, 2017.

[35] B. Genc, M. Siala, G. Simonin, and B. O'Sullivan. Complexity study for the robust stable marriage problem. *Theor. Comput. Sci.*, 775:76–92, 2019.

[36] G. J. Hitsch, A. Hortaçsu, and D. Ariely. Matching and sorting in online dating. *Am. Econ. Rev.*, 100(1):130–163, 2010.

[37] V. Kanade, N. Leonardos, and F. Magniez. Stable matching with evolving preferences. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX/RANDOM '16)*, pages 36:1–36:13. Schloss Dagstuhl, 2016.

[38] J. Kennes, D. Monte, N. Tumennasan, et al. *The daycare assignment problem*. Department of Economics and Business Economics, Aarhus BSS, 2011.

[39] D. E. Knuth. Marriages stables. *Technical report*, 1976.

[40] C. Liu. Stability in repeated matching markets. *CoRR*, abs/2007.03794v2, 2021. URL https://arxiv.org/abs/2007.03794.

[41] T. Mai and V. V. Vazirani. Finding stable matchings that are robust to errors in the input. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA '18)*, pages 60:1–60:11. Schloss Dagstuhl, 2018.

[42] C. Mallows. Non-null ranking models. *Biometrica*, 44:114–130, 1957.

[43] D. F. Manlove. *Algorithmics of Matching Under Preferences*, volume 2 of *Series on Theoretical Computer Science*. WorldScientific, 2013.

[44] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983.

[45] A. E. Roth. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica*, 54(2): 425–427, 1986.

[46] P. Schober, C. Boer, and L. A. Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesth. Analg.*, 126(5):1763–1768, 2018.

[47] D. Shiryaev, L. Yu, and E. Elkind. On elections with robust winners. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '13)*, pages 415–422. IFAAMAS, 2013.

[48] J. H. Vande Vate. Linear programming brings marital bliss. *Oper. Res. Lett.*, 8(3):147–153, 1989.