Journal of Artificial Intelligence Research 71 (2021) 993-1048 Submitted 02/2021; published 08/2021

Bribery and Control in Stable Marriage

Niclas Boehmer NICLAS.BOEHMER@TU-BERLIN.DE
Technische Universität Berlin,
Algorithmics and Computational Complexity,
Berlin, Germany

Robert Bredereck ROBERT.BREDERECK@HU-BERLIN.DE
Technische Universität Berlin,
Algorithmics and Computational Complexity,
Berlin, Germany, and
Humboldt-Universität zu Berlin, Institut für Informatik,
Berlin, Germany

Klaus Heeger HEEGER@TU-BERLIN.DE
Rolf Niedermeier ROLF.NIEDERMEIER@TU-BERLIN.DE
Technische Universität Berlin,
Algorithmics and Computational Complexity,
Berlin, Germany

Abstract

We initiate the study of external manipulations in Stable Marriage by considering several manipulative actions as well as several manipulation goals. For instance, one goal is to make sure that a given pair of agents is matched in a stable solution, and this may be achieved by the manipulative action of reordering some agents' preference lists. We present a comprehensive study of the computational complexity of all problems arising in this way. We find several polynomial-time solvable cases as well as NP-hard ones. For the NP-hard cases, focusing on the natural parameter "budget" (that is, the number of manipulative actions one is allowed to perform), we also conduct a parameterized complexity analysis and encounter mostly parameterized hardness results.

1. Introduction

In the Stable Marriage problem, we have two sets of agents, each agent has preferences over all agents from the other set, and the goal is to find a matching between agents of the one set and agents of the other set such that no two agents prefer each other to their assigned partners. In this paper, we study the manipulation of a Stable Marriage instance by an external agent that is able to change the set of agents or (parts of) their preferences. To motivate our studies, consider the following example.

Due to the high demand from students, Professor X decides to implement a central matching scheme using Stable Marriage to assign interested students to final-year projects offered by X's group: Every term X asks each of her group members to propose a project. Subsequently, projects are put online and interested students are asked to submit their preferences over projects, while group members are asked to submit their preferences over the students. Afterwards, a stable matching of group members/projects and students is computed and implemented.[1] As the submitted preferences are visible to all group members, two days before the deadline, one group member realizes that in the currently only stable matching he is not matched to the student with whom he already started thinking about his proposal. That is why, in order to be matched to his preferred student in at least some stable matching, he motivates two so far non-participating students that were unsure whether to do their final project this year or next year to register already this year. After this change, another group member notices that she is now matched to her least preferred choice in some of the matchings that are currently stable but she already has an idea how to change this: She quickly visits two of her fellow group members and tells a story about how great two of the registered students performed in her least year's tutorial. In this way, she convinces them to rank these two students higher in their preferences. On the last day before the deadline, Professor X realizes that there currently exist multiple stable matchings. As X believes that this may cause unnecessary discussions on which stable matching to choose, X tells the two newest group members who anyway know only few of the students how they should change their preferences.

Another application of the non-bipartite version of Stable Marriage (called Stable Roommates), which might be easily susceptible to external manipulation arises in the context of P2P networks. This affects, in particular, the BitTorrent protocol for content distribution (Lebedev, Mathieu, Viennot, Gai, Reynier, & de Montgolfier, 2007; Gai, Mathieu, de Montgolfier, & Reynier, 2007). Here, users rank each other based on some technical data, e.g., their download and upload bandwidth and based on the similarity of their interests. However, a user may easily add new users by simply entering the network multiple times with different accounts, thereby influencing the computed matching.

Looking at further applications of Stable Marriage and corresponding generalizations in the context of matching markets, there is clear evidence of external manipulations in modern applications. For instance, surveys reported that in college admission systems in China, Bulgaria, Moldova, and Serbia, bribes have been performed in order to gain desirable admissions (Heyneman, Anderson, & Nuraliyeva, 2008; Liu & Peng, 2015). Focusing on the most basic scenario Stable Marriage, we initiate a thorough study of manipulative actions (bribery and control) from a computational complexity perspective. Notably, bribery scenarios have also been used as a motivation in other papers around Stable Marriage, e.g., when finding robust stable matchings (Chen, Skowron, & Sorge, 2019) or when studying strongly stable matchings in the Hospitals/Residents problem with ties (Irving, Manlove, & Scott, 2003).

External manipulation may have many faces such as deleting agents, adding agents, or changing agents' preference lists. We consider three different manipulation goals and five different manipulative actions.

We introduce the manipulation goals *Constructive-Exists*, *Exact-Exists*, and *Exact-Unique*, where *Constructive-Exists* is the least restrictive goal and asks for modifications such that a desired agent pair is contained in some stable matching. More restrictively, *Ex-*

1. Notably, there is a wide spectrum of literature concerned with finding a (stable) matching of students to final-year projects/supervisors (see Manlove, 2013, Chapter 5.5 for a survey). A central mechanism for finding a (stable) matching of students to final-year projects has been implemented at the University of Glasgow (Manlove, 2013), the University of York (Kazakov, 2001), and several other universities (Hussain, Gamage, Sagor, Tariq, Ma, & Imran, 2019; Anwar & Bahaj, 2003).

act-Exists asks for modifications such that a desired matching is stable. Most restrictively, *Exact-Unique* requires that a desired matching becomes the only stable matching.

As manipulative actions, we investigate *Swap*, *Reorder*, *DeleteAcceptability*, *Delete*, and *Add*. The actions *Swap* and *Reorder* model bribery through an external agent. While a single *Reorder* action allows to completely change the preferences of an agent (modeling a briber who can "buy an agent"), a *Swap* action is more fine-granular and only allows to swap two neighboring agents in some agent's preference list (modeling a briber who has to slightly convince agents where the costs/effort to convince an agent of some preferences is larger if the preferences are further away from the agent's true preferences). For both actions, the external agent might actually change the true preferences of the influenced agent, for example, by advertising some possible partner. However, in settings where the agents' preferences serve as an input for a centralized mechanism computing a stable matching which is subsequently implemented and cannot be changed, it is enough to bribe the agents to cast untruthful preferences. *Delete* and *Add* model control of the instance. They are useful to model an external agent, i.e., the organizer of some matching system, with the power to accept or reject agents to participate or to change the participation rules. While a *Delete* (resp. *Add*) action allows to delete (resp. add) an agent to the instance, a *DeleteAcceptability* action forbids for a specific pair of agents the possibility to be matched to each other and to be blocking. The latter can be seen as a hybrid between bribery and control because it can model an external agent that changes acceptability rules or it can model a briber who convinces an agent that another agent is unacceptable at some cost.

We conduct a complete analysis of all fifteen combinations of our five manipulative actions and three manipulation goals. Note, however, that for the two actions *Delete* and *Add* the definition of *Exact-Exists* and *Exact-Unique* introduced above cannot be directly applied, which is why we propose an adapted definition in Section 2.4.

Related Work. Since its introduction (Gale & Shapley, 1962), Stable Marriage has been intensely studied by researchers from different disciplines and in many contexts (see, e.g., the surveys of Gusfield & Irving, 1989; Knuth, 1976; Manlove, 2013).

A topic related to manipulation in stable matchings is the study of *strategic behavior*, which focuses on the question whether agents can misreport their preferences to fool a given matching algorithm to match them to a better partner. Numerous papers have addressed the question of strategic behavior for different variants of computing stable matchings, matching algorithms, types of agents' preferences and restrictions on the agents that are allowed to misreport their preferences (e.g., Aziz, Seedig, & von Wedel, 2015; Hosseini, Umar, & Vaish, 2021; Pini, Rossi, Venable, & Walsh, 2011; Roth, 1982; Teo, Sethuraman, & Tan, 2001; Shen, Tang, & Deng, 2018; see Manlove, 2013, Chapter 2.9 for a survey). This setting is related to ours in the sense that the preferences of agents are modified to achieve a desired outcome, while it is fundamentally different with respect to the allowed modifications and their goal: In the context of strategic behavior an agent is only willing to change *its* preferences if the agent directly benefits from it. Notably, in the context of strategic behavior, Gonczarowski (2014) investigated how all agents from one side together can alter their preferences (including to declare some agents unacceptable) in order to achieve that a given matching is the unique stable matching. This goal corresponds to our *Exact-Unique* setting. However, while Gonczarowski (2014) allows for arbitrary changes in the preferences of all agents from one side including the deletion of the acceptability of agent pairs, in the problems considered in this paper we always allow the briber to influence all agents and only either allow for reordering the preferences arbitrarily (*Reorder*) or deleting the acceptability of agent pairs (*DeleteAcceptability*) (and aim to minimize the number of such manipulations).

Generally speaking, using our manipulative action *Swap* combined with an appropriate manipulation goal, it is possible to model most computational problems where a single agent or a group of agents wants to fool an algorithm by misreporting their preferences: In order to do so, we set the budget such that the preferences of the agents from the cheating group can be modified arbitrarily. Further, we introduce dummy agents that are always matched among themselves in any stable matching (see Lemma 7). We add these dummy agents between each pair of consecutive agents in the preferences of all non-manipulating agents such that the budget is not sufficient to swap two non-dummy agents in the preferences of these agents. While this reduction shows how *Swap* can model computational problems related to strategic behavior of a group of agents, as already mentioned, the goals typically considered in the study of strategic behavior differ significantly from the ones studied in our paper. In the context of strategic behavior, the focus of a cheating group usually lies on misreporting their preferences such that all of them are matched to a better partner in a matching returned by a specific matching algorithm (as done, e.g., by Aziz et al., 2015; Pini et al., 2011; Roth, 1982; Teo et al., 2001), while our focus lies on making a specific pair or matching stable.

While we are interested in finding ways to influence a profile to change the set of stable matchings, finding *robust* stable matchings (Chen et al., 2019; Mai & Vazirani, 2018a, 2018b) corresponds to finding stable matchings such that a briber cannot easily make the matching unstable. For instance, Chen et al. (2019) introduced the concept of *d-robustness*: A matching is d-robust if it is stable in the given instance and remains stable even if d arbitrary swaps in preference lists are performed. One motivation for their study of d-robust stable matchings is that d-robust stable matchings withstand bribers which may perform up to d swaps.

Conceptually, our work is closely related to the study of bribery and control in elections (see Faliszewski & Rothe, 2016 for a survey). In election control problems (Bartholdi III, Tovey, & Trick, 1992), the goal is to change the structure of a given election, e.g., by modifying the candidate or voter set, such that a designated candidate becomes the winner/looser of the resulting election. In bribery problems (Faliszewski, Hemaspaandra, & Hemaspaandra, 2009), the briber is allowed to modify the votes in the election to achieve the goal. Most of the manipulative actions we consider are inspired by either some control operation or bribery action already studied in the context of voting.

Our manipulation goals are also related to problems previously studied in the stable matching literature: For example, the *Constructive-Exists* problem with given budget zero reduces to the Stable Pair problem, which aims at deciding whether a given agent pair is contained in at least one stable matching. While the problem is polynomial-time solvable for Stable Marriage instances without ties (Gusfield, 1987), deciding whether an agent pair is part of a "weakly stable" matching is NP-hard if ties are allowed (Manlove, Irving, Iwama, Miyazaki, & Morita, 2002). This directly implies hardness of the *Constructive-Exists* problem if ties are allowed even when the budget is zero. Similarly, deciding

Action/Goal	*Constructive-Exists*	*Exact-Exists*	*Exact-Unique*
Swap	$\forall\epsilon > 0$: W[1]-hard wrt. ℓ to approx. within a factor of $\mathcal{O}(n^{1-\epsilon})$ (Th. 3)	P (Th. 5)	NP-c. (Pr. 9)
Reorder	W[1]-h. wrt. ℓ (Th. 2) 2-approx in P (Pr. 2)	P (Pr. 3)	W[2]-h. wrt. ℓ (Th. 6)
Delete Accept.	W[1]-h. wrt. ℓ (Th. 2)	P (Ob. 1)	P (Th. 7)
Delete	P (Th. 4)	NP-c. (Pr. 6) FPT wrt. ℓ (Pr. 7)	NP-c. (Pr. 6)
Add	W[1]-h. wrt. ℓ (Th. 1) NP-c. even if $\ell = \infty$ (Th. 1)	P (Pr. 5)	W[2]-h. wrt. ℓ (Pr. 8)

Table 1: Overview of our results, where ℓ denotes the given budget. All stated W[1]- and W[2]-hardness results also imply NP-hardness. See Section 2 for definitions of parameterized complexity classes and formal problem definitions.

whether there exists a weakly stable matching *not* containing a given agent pair is also NP-hard in the presence of ties (Cseh & Heeger, 2020). Moreover, several authors studied sufficient or necessary conditions for a STABLE MARRIAGE instance to admit a unique stable matching (Clark, 2006; Consuegra, Kumar, & Narasimhan, 2013; Drgas-Burchardt, 2013; Eeckhout, 2000; Gelain, Pini, Rossi, Venable, & Walsh, 2011; Reny, 2021).

Our Contributions. Providing a complete polynomial-time solvability vs. NP-hardness dichotomy, we settle the computational complexity of all problems emanating from our manipulation scenarios. We also conduct a parameterized complexity analysis of these problems based on the budget parameter ℓ, that is, the number of elementary manipulative actions that we are allowed to perform. At some places, we also consider the approximability of our problems in polynomial time or FPT time. For instance, we prove that CONSTRUCTIVE-EXISTS-SWAP does not admit an $\mathcal{O}(n^{1-\epsilon})$-approximation in $f(\ell)n^{\mathcal{O}(1)}$ time for any $\epsilon > 0$ and any computable function f unless FPT=W[1]. Table 1 gives an overview of our results. Furthermore, for all problems we observe XP-algorithms[2] with respect to the parameter ℓ. The CONSTRUCTIVE-EXISTS-REORDER and EXACT-UNIQUE-REORDER problem require non-trivial algorithms to show this.

We highlight the following five results and techniques.

- We develop a quite general framework for constructing parameterized reductions from the W[1]-hard graph problem CLIQUE to the *Constructive-Exists* problem and design the required gadgets for *Add*, *DeleteAcceptability*, and *Reorder* (Theorems 1 and 2).
- We design a simple and efficient algorithm for CONSTRUCTIVE-EXISTS-DELETE (Theorem 4), based on a non-trivial analysis.
- We design a concise parameterized reduction from the W[2]-hard problem HITTING SET to EXACT-UNIQUE-REORDER. Surprisingly, in the constructed instance, to make

2. That is, polynomial-time algorithms if ℓ is constant.

the target matching the unique stable matching, the preferences of some agents need to be reordered by swapping *down* their (desired) partner in the given matching (Theorem 6).

- We analyze how the manipulative actions *DeleteAcceptability* and *Reorder* can be used to modify the so-called rotation poset to make a given matching the unique stable matching (Theorems 7 and 8).
- Our polynomial-time algorithms exhibit surprising connections between manipulations in Stable Marriage and the classical graph problems Bipartite Vertex Cover (Proposition 3), Minimum Cut (Theorem 5), and Weighted Minimum Spanning Arborescence (Theorem 7).

Comparing the results for the different combinations of manipulation goals and manipulative actions, we observe a quite diverse complexity landscape: While for all other manipulative actions the corresponding problems are computationally hard, Constructive-Exists-Delete and Exact-Unique-DeleteAcceptability are polynomial-time solvable. Relating the different manipulation goals to each other, we show that specifying a complete matching that should be made stable instead of just one agent pair that should be part of some stable matching makes the problem of finding a successful manipulation significantly easier. In contrast to this, providing even more information about the resulting instance by requiring that the given matching is the unique stable matching instead of just one of the stable matchings makes the problem of finding a successful manipulation again harder.

From a high-level perspective, our computational hardness results can be seen as a shield against manipulative attacks. Of course, these shields are not unbreakable, as they only offer a worst-case protection against computing an attack of minimum cost. However, we slightly strengthen these (worst-case) shields by also proving parameterized hardness results for the parameter budget, which might be small compared to the number of agents especially in large matching markets. In contrast to this, our polynomial-time algorithms suggest that market makers shall be extra cautious in situations where the corresponding manipulative action can be easily performed.

There also is a more positive interpretation of bribery and control (Faliszewski, Skowron, & Talmon, 2017; Boehmer, Bredereck, Knop, & Luo, 2020): The minimum cost of a successful attack for *Constructive-Exists* can be interpreted as a measure for the "distance from stability" of the corresponding pair. Similarly, the minimum cost for *Exact-Exists* can be interpreted as the "distance from stability" of the corresponding matching. Both metrics might be particularly interesting in applications where a central authority decides on a matching for which stability-related considerations are important but perfect stability is not vital. For instance, in our introductory example, Professor X might be dissatisfied with the matchings that are currently stable and therefore could use the "distance from stability" measure to decide between a few different matchings she deems acceptable. As *Swap* can be understood as the most fine-grained of our considered manipulative actions, in such situations it might be particularly appealing to use our polynomial-time algorithm for Exact-Exists-Swap to compute the swap distance from stability of a matching. Lastly, *Exact-Unique* offers a measure for the "distance from unique stability" of a matching. In practice, this distance could, for instance, serve as a tie-breaker between different stable matchings. However, there also exists a destructive view on bribery and control problems, where the goal is to prevent a given pair/matching from being stable. Destructive bribery can thus be interpreted as a distance measure from being unstable and can be used to quantify the robustness of the stability of a pair or a matching (Faliszewski et al., 2017; Boehmer et al., 2020; Boehmer, Bredereck, Faliszewski, & Niedermeier, 2021). While we focus on a constructive view, some of our results such as the polynomial-time algorithm for Constructive-Exists-Delete carry over to the destructive variant (see Section 3.3.2).

Organization of the Paper. Section 2 delivers background on parameterized complexity analysis and the Stable Marriage problem, and it formally defines the different considered manipulative actions and manipulation goals. Afterwards, we devote one section to each of the manipulation goals we analyze. In Section 3, we consider the *Constructive-Exists* setting for all manipulative actions. We split this section into two parts; in the first part, we prove several W[1]-hardness results, and in the second part, we present a polynomial-time algorithm for the *Delete* action and a polynomial-time factor-2-approximation algorithm for the *Reorder* action. In Section 4, we present our results for *Exact-Exists*. After considering the manipulative actions *Swap*, *Reorder*, and *DeleteAcceptability* in the first part of this section, we analyze the actions *Add* and *Delete* in the second part. In Section 5, we start by presenting hardness results for the *Exact-Unique* setting and then derive a polynomial-time for *DeleteAcceptability* as well as one XP algorithm for *Reorder*. We conclude in Section 6, indicating directions for future research and presenting few very preliminary insights from experimental work with our algorithms.

2. Preliminaries and First Observations

In this section, we start by recapping some fundamentals of parameterized complexity theory (Section 2.1) and defining the Stable Marriage problem and related concepts (Section 2.2). Subsequently, we introduce and formally define the five manipulative actions (Section 2.3) and three manipulation goals (Section 2.4) we study. Lastly, in Section 2.5, we make some first observations about the relationship of the different manipulative actions on a rather intuitive level.

2.1 Parameterized Complexity

A *parameterized problem* consists of a problem instance $\mathcal{I}$ and a (typically integer) parameter value k (in our case the budget ℓ).[3] It is called *fixed-parameter tractable* with respect to k if it can be solved by an *FPT-algorithm*, i.e., an algorithm running in $f(k)|\mathcal{I}|^{O(1)}$ time for a computable function f. Moreover, it lies in XP with respect to k if it can be solved in $|\mathcal{I}|^{f(k)}$ time for some computable function f. There is also a theory of hardness of parameterized problems that includes the notion of W[t]-hardness with W[t] $\subseteq$ W[t'] for $t \leq t'$. If a problem is W[t]-hard for a given parameter for any $t \geq 1$, then it is widely believed not to be fixed-parameter tractable for this parameter. The usual approach to prove that a given

3. To simplify complexity-theoretic matters, by default parameterized problems are framed as decision problems. However, our positive algorithmic results easily extend to the corresponding optimization and search problems.

parameterized problem is W[t]-hard is to describe a parameterized reduction from a known W[t]-hard problem to it. In our case, we only use the following special case of parameterized reductions: Standard many-one reductions that run in polynomial time and ensure that the parameter of the output instance is upper-bounded by a function of the parameter of the input instance.

2.2 Stable Marriage

An instance $\mathcal{I}$ of the Stable Marriage (SM) problem consists of a set $U = \{m_1, \dots m_n\}$ of men and a set $W = \{w_1, \dots, w_n\}$ of women, together with a strict *preference list* $\mathcal{P}_a$ for each $a \in U \cup W$.[4] Note that following conventions from the literature and as this simplifies discussions in some places, we assume that in all considered SM instances $\mathcal{I} = (U, W, \mathcal{P})$, it holds that $|U| = |W| = n$ (for Add, this means that after adding all agents to the instance the number of women and men is the same).[5] Moreover, we call the elements of $U \cup W$ *agents* and $A = U \cup W$ denotes the set of agents. The preference list $\mathcal{P}_a$ of an agent a is a strict order over the agents of the opposite gender. We denote the preference list of an agent $a \in A$ by $a : a_1 \succ a_2 \succ a_3 \succ \dots$, where a_1 is a's most preferred agent, a_2 is a's second most preferred agent, and so on. For the sake of readability, we sometimes only specify parts of the agents' preference relation and end the preferences with "$\succ \overset{\text{(rest)}}{\dots}$". In this case, it is possible to complete the given profile by adding the remaining agents in an arbitrary order. We say that *a prefers* a' to a'' if a ranks a' above a'' in its preference list, i.e., $a' \succ_a a''$. For two agents $a, a' \in A$ of opposite gender, let $\operatorname{rank}(a, a')$ denote the rank of a' in the preference relation of a, i.e., one plus the number of agents which a prefers to a'.

A *matching* M is a set of pairs $\{m, w\}$ with $m \in U$ and $w \in W$ such that each agent is contained in at most one pair. An agent is *assigned* in a matching M if some pair of M contains this agent, and *unassigned* otherwise. For a matching M and an assigned agent $a \in A$, we denote by $M(a)$ the agent a is matched to in M, i.e., $M(a) = a'$ if $\{a, a'\} \in M$. We slightly abuse notation and write $a \in M$ for an agent a if there exists some agent a' such that $\{a, a'\} \in M$. A matching is called *complete* if no agent is unassigned. For a matching M, a pair $\{m, w\}$ with $m \in U$ and $w \in W$ is *blocking* if both m is unassigned or prefers w to $M(m)$ and w is unassigned or prefers m to $M(w)$. A matching is *stable* if it does not admit a blocking pair. We denote the set of stable matchings in an SM instance $\mathcal{I}$ by $\mathcal{M}_{\mathcal{I}}$. Given a matching M and some subset $A' \subseteq A$ of agents, we denote by $M|_{A'}$ the restriction of M to A', i.e., $M|_{A'} = \{\{u, w\} \in M : u, w \in A'\}$. A stable matching M is called *man-optimal* if for every man $m \in U$ and every stable matching M' it holds that m does not prefer $M'(m)$ to $M(m)$. Symmetrically, a stable matching M is called *woman-optimal* if for every woman $w \in W$ and every stable matching M' it holds that w does not

4. We are well aware of the fact that Stable Marriage can be criticized for advocating and transporting outdated role models or conservative marriage concepts. First, we emphasize that we use the old concepts (men matching with women) for notational convenience and for being in accordance with the very rich, also recent literature. Second, we remark that in real-world applications Stable Marriage models general two-sided matching markets, which may appear in different scenarios such as matching students with supervisors or matching mines with deposits.
5. Notably, this assumption is crucial for our polynomial-time 2-approximation algorithm for Constructive-Exists-Reorder in Section 3.3.2.

prefer $M'(w)$ to $M(w)$. Gale and Shapley (1962) showed that a man-optimal as well as a woman-optimal matching always exist.

The Stable Marriage with Incomplete Lists (SMI) problem is a generalization of the Stable Marriage problem where each agent a is allowed to specify incomplete preferences of agents of the opposite gender. Then, a pair of agents $\{m, w\}$ with $m \in U$ and $w \in W$ can only be part of a stable matching M if they both appear in each others' preference list. A pair $\{m, w\}$ with $m \in U$ and $w \in W$ is blocking if m and w appear on each other's preferences and both m is unassigned or prefers w to $M(m)$, and w is unassigned or prefers m to $M(w)$. Let $\mathrm{ma}(M)$ denote the set of agents matched in a stable matching M. Note that by the Rural Hospitals Theorem (Roth, 1986) it holds for all stable matchings $M, M' \in \mathcal{M}_{\mathcal{I}}$ that $\mathrm{ma}(M) = \mathrm{ma}(M')$. Moreover, for an SMI instance $\mathcal{I}$, let $\mathrm{ma}(\mathcal{I})$ denote the set of agents that are matched in a stable matching in $\mathcal{I}$.

2.3 Manipulative Actions

We introduce five different manipulative actions and necessary notation in this subsection. We denote by $\mathcal{X} \in \{\mathit{Swap}, \mathit{Reorder}, \mathit{DeleteAcceptability}, \mathit{Delete}, \mathit{Add}\}$ the type of a manipulative action.

Swap. A *Swap* operation changes the order of two neighboring agents in the preference list of an agent.

Example 1. *Let a be an agent, and let its preference list be $a : a_1 \succ a_2 \succ a_3$. There are two possible (single) swaps: Swapping a_1 and a_2, resulting in $a : a_2 \succ a_1 \succ a_3$, and swapping a_2 and a_3, resulting in $a : a_1 \succ a_3 \succ a_2$.*

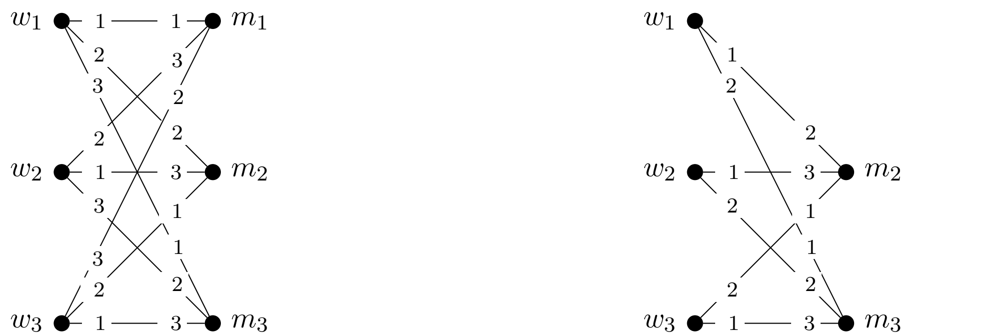
Reorder. A *Reorder* operation of an agent's preference list reorders its preferences arbitrarily, i.e., one performs an arbitrary permutation.

Example 2. *For an agent a with preference list $a : a_1 \succ a_2 \succ a_3$, there are six possible reorderings, resulting in one of the six possible strict total orders over $\{a_1, a_2, a_3\}$.*

Delete Acceptability. A *DeleteAcceptability* operation is understood as deleting the mutual acceptability of a man and a woman. This enforces that such a deleted pair cannot be part of any stable matching and cannot be a blocking pair for any stable matching. Thus, after applying a *DeleteAcceptability* action, the given SM instance is transformed into an SMI instance. For two agents $a, a' \in A$, we sometimes also say that we delete the pair or edge $\{a, a'\}$ if we delete the mutual acceptability of the two agents a and a'.

Example 3. *Let m be a man with preferences $m : w_1 \succ w_2 \succ w \succ w_3$, and w be a woman with preferences $w : m \succ m_1 \succ m_2 \succ m_3$. Deleting the pair $\{m, w\}$ results in the following preferences:*
$m : w_1 \succ w_2 \succ w_3$ *and* $w : m_1 \succ m_2 \succ m_3$.

Delete. A *Delete* operation deletes an agent from the instance. Note that we allow for deleting an unequal number of men and women.



(a) Original instance

(b) Modified instance after deleting m_1

Figure 1: Visualization of Example 4 for a *Delete* operation. The numbers on the edges encode the preferences of the agents: The number i of an edge closer to an agent a means that a considers the other endpoint of this edge as its i-th best partner.

Example 4. *Let $\mathcal{I}$ be an SM instance with agents $U = \{m_1, m_2, m_3\}$ and $W = \{w_1, w_2, w_3\}$, and the following preferences:*

$$m_1 : w_1 \succ w_3 \succ w_2, \qquad m_2 : w_3 \succ w_1 \succ w_2, \qquad m_3 : w_1 \succ w_2 \succ w_3,$$
$$w_1 : m_1 \succ m_2 \succ m_3, \qquad w_2 : m_2 \succ m_1 \succ m_3, \qquad w_3 : m_3 \succ m_2 \succ m_1.$$

The instance $\mathcal{I}$ is visualized in Figure 1a. Deleting agent m_1 results in the instance $\mathcal{I}' = (U', W', \mathcal{P}')$ with $U' = \{m_2, m_3\}$, $W' = \{w_1, w_2, w_3\}$, and preferences:

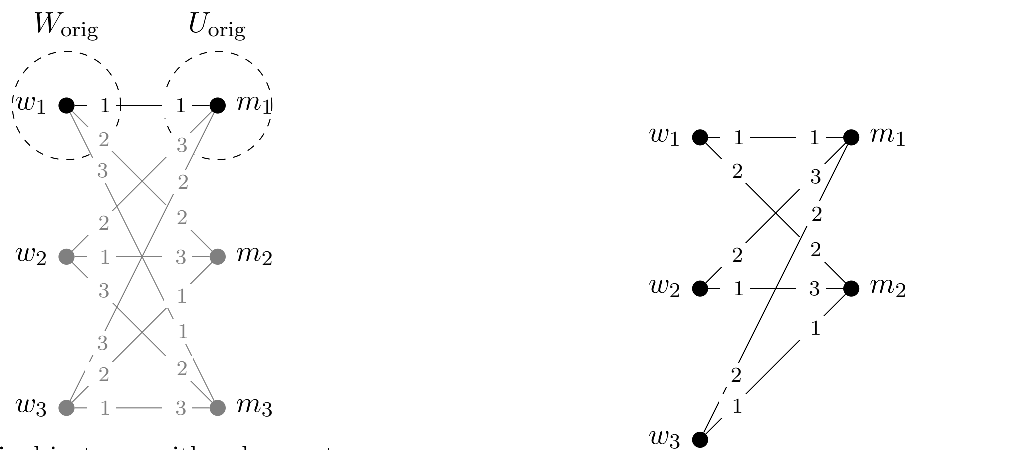
$$m_2 : w_3 \succ w_1 \succ w_2, \qquad m_3 : w_1 \succ w_2 \succ w_3,$$
$$w_1 : m_2 \succ m_3, \qquad w_2 : m_2 \succ m_3, \qquad w_3 : m_3 \succ m_2.$$

The modified instance is visualized in Figure 1b.

Given an SM instance $\mathcal{I} = (U, W, \mathcal{P})$ and a subset of agents $A' \subseteq A$, we define $\mathcal{I} \setminus A'$ to be the instance that results from deleting the agents A' from $\mathcal{I}$.

Add. An *Add* operation adds an agent from a predefined set of agents to the instance. Formally, the input for a computational problem considering the manipulative action *Add* consists of an SM instance $(U, W, \mathcal{P})$ together with two subsets $U_{\text{add}} \subseteq U$ and $W_{\text{add}} \subseteq W$. The sets U_{add} and W_{add} contain agents that are not initially present and can be added to the original instance. All other men $U_{\text{orig}} := U \setminus U_{\text{add}}$ and women $W_{\text{orig}} := W \setminus W_{\text{add}}$ are already initially present and part of the original instance. Adding a set of agents $X_A = X_U \cup X_W$ with $X_U \subseteq U_{\text{add}}$ and $X_W \subseteq W_{\text{add}}$ results in the instance $(U_{\text{orig}} \cup X_U, W_{\text{orig}} \cup X_W, \mathcal{P}')$, where $\mathcal{P}'$ is the restriction of $\mathcal{P}$ to agents from $U_{\text{orig}} \cup X_U \cup W_{\text{orig}} \cup X_W$. Note that as mentioned in Section 2.2, we require that $|U| = |W|$ but do not impose constraints on $|U_{\text{add}}|$ and $|W_{\text{add}}|$ or whether the same number of men and women is added to the instance by the manipulation.

Example 5. *An example for the application of an Add operation is depicted in Figure 2. The instance consists of three men $U = \{m_1, m_2, m_3\}$ and three women $W = \{w_1, w_2, w_3\}$*



(a) Original instance with only agents w_1 and m_1 present (to which agents w_2, m_2, w_3, and m_3 can be added).

(b) Modified instance after adding m_2, w_2, and w_3.

Figure 2: Example for the application of manipulative action *Add*.

with $U_{add} = \{m_2, m_3\}$ *and* $W_{add} = \{w_2, w_3\}$. *That is, only the agents* m_1 *and* w_1 *are initially present in the instance and all other agents can be added by a manipulative action. After adding* $X_U = \{m_2\}$ *and* $X_W = \{w_2, w_3\}$, *the agent sets change to* $U^* = \{m_1, m_2\}$ *and* $W^* = \{w_1, w_2, w_3\}$, *resulting in the preference profile shown in Figure 2b.*

2.4 Manipulation Goals

In the *Constructive-Exists* setting, the goal is to modify a given SM instance using manipulative actions of some given type such that a designated man-woman pair is part of some stable matching. For $\mathcal{X} \in \{\mathit{Swap}, \mathit{DeleteAcceptability}, \mathit{Delete}, \mathit{Add}\}$, the formal definition of the problem is presented below.

Constructive-Exists-$\mathcal{X}$

Input:	Given an SM instance $\mathcal{I} = (U, W, \mathcal{P})$, a man-woman pair $\{m^*, w^*\}$, and a budget $\ell \in \mathbb{N}$.
Question:	Is it possible to perform at most ℓ manipulative actions of type $\mathcal{X}$ such that $\{m^*, w^*\}$ is part of at least one matching that is stable in the altered instance?

If one applies the above problem formulation to *Reorder*, then the resulting problem always allows a trivial solution of size two by reordering the preferences of m^* and w^* such that they are each other's top-choice. Hence, for Constructive-Exists-Reorder, we forbid the reordering of the preferences of m^* and w^*, resulting in the following problem formulation.

Constructive-Exists-Reorder

Input: Given an SM instance $\mathcal{I} = (U, W, \mathcal{P})$, a man-woman pair $\{m^*, w^*\}$, and a budget $\ell \in \mathbb{N}$.

Question: Is it possible to perform at most ℓ reorderings of the preferences of agents other than m^* and w^* such that $\{m^*, w^*\}$ is part of at least one stable matching in the altered instance?

In the *Exact* setting, in contrast to the *Constructive* setting, we are given a complete matching. Within this setting, we consider two different computational problems. First, we consider the Exact-Exists problem where the goal is to modify a given SM instance such that the given matching is stable in the instance. Second, we consider the Exact-Unique problem where the goal is to modify a given SM instance such that the given matching is the *unique* stable matching.

Exact-Exists (Unique)-$\mathcal{X}$

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

Question: Is it possible to perform at most ℓ manipulative actions of type $\mathcal{X}$ such that M^* is a (the unique) stable matching in the altered instance?

For manipulative actions *Delete* and *Add*, the definitions of Exact-Exists-$\mathcal{X}$ and Exact-Unique-$\mathcal{X}$ are not directly applicable, as the set of agents changes by applying *Delete* or *Add* operations. That is why, for these actions, we need to slightly adapt the definitions from above. The general idea behind the proposed adaption is that we specify a complete matching on all agents (including those from $U_{\text{add}} \cup W_{\text{add}}$ for *Add*), and require that the restriction of the specified matching M^* to the agents contained in the manipulated instance should be the (unique) stable matching in the manipulated instance. The reasoning behind this is that we only want to allow pairs that we approve to be part of a (the) stable matching. This results in the following definition for Exact-Exists (Unique)-Delete:

Exact-Exists (Unique)-Delete

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

Question: Is it possible to delete at most ℓ agents from $U \cup W$ such that there exists some $M' \subseteq M^*$ which is a (the unique) stable matching in the altered instance?

Similarly, we get the following definition for Exact-Exists (Unique)-Add:

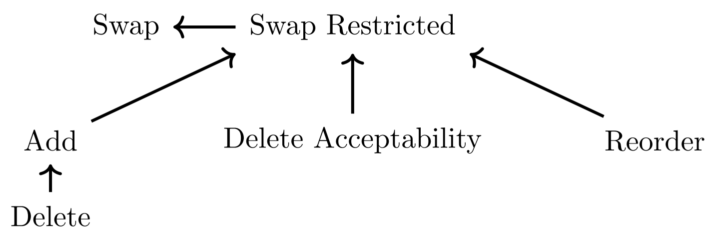


Figure 3: Relationship between manipulative actions as described in Section 2.5. An arc from an action $\mathcal{X}$ to an action $\mathcal{Y}$ indicates that we present a “simple” description how to model action $\mathcal{X}$ by action $\mathcal{Y}$. Note that an arc does not imply that a computational problem for $\mathcal{X}$ is always reducible to the same problem for $\mathcal{Y}$.

Exact-Exists (Unique)-Add

Input:	Given an SM instance $\mathcal{I} = (U, W, \mathcal{P})$ together with subsets $U_{\text{add}} \subseteq U$ and $W_{\text{add}} \subseteq W$, a complete matching M^*, and budget $\ell \in \mathbb{N}$.
Question:	Is it possible to add at most ℓ agents $X_A \subseteq U_{\text{add}} \cup W_{\text{add}}$ such that there exists some $M' \subseteq M^*$ which is a (the unique) stable matching in the altered instance?

There also exist natural optimization variants of all considered decision problems which ask for the minimum number of manipulative actions that are necessary to alter a given SM instance to achieve the specified goal. The (in-)approximability results in Theorem 3 and Proposition 2 refer to the optimization variants of these problems.

2.5 Relationship Between Different Manipulative Actions

In this section, to get an overview of the different manipulative actions, we analyze how they relate to each other. This section, however, does not aim at introducing formal relationships in the sense of a general notion of reducibility of two manipulative actions $\mathcal{X}$ and $\mathcal{Y}$, as different manipulation goals require different properties of a reduction. Instead, we present high-level ideas how it is possible to simulate one action with another action. We mainly aim at giving an intuition for the relationships between the actions that helps to understand, relate, and classify the results we present in the paper. Note that most of the sketched relationships are applicable when relating computational problems around the *Constructive-Exists* goal for the different manipulative actions to each other (and not so much for *Exact-Exists* and *Exact-Unique*). For an overview of the relations see Figure 3.

***Delete* via *Add*.** It is possible to model *Delete* actions by *Add* actions by adapting the considered SM instance as follows. We keep all agents $a \in A$ from the original instance and introduce for each of them a new designated *binding agent* a' of opposite gender that ranks a first and all other agents in an arbitrary ordering afterwards. Moreover, we add a' at the first position of the preferences of a. The set of agents that can be added to the instance are the binding agents.
Then, adding the binding agent a' in the modified instance corresponds to deleting the

corresponding non-binding agent a in the original instance, as in this case the agent and its binding agent are their mutual top-choices and thereby always matched in a stable matching.

Restricted *Swap* via *Swap*. Before we describe how different manipulative actions can be modeled by *Swap* actions, we first sketch how it is possible to model a variant of *Swap* where for a given set of pairs consisting of adjacent positions swapping agents on each such pair of neighboring positions is forbidden and swapping the first and the second element in the preference relation of an agent may have some specified non-unit cost. To model this variant, we introduce $n(\ell+1)$ dummy men and $n(\ell+1)$ dummy women each ranking all dummy agents of the opposite gender before the other agents. Due to the preferences of the dummy agents, regardless of which ℓ swaps are performed in the preferences of dummy agents, in all stable matchings all dummy agents are matched to dummy agents and a dummy agent is never part of a blocking pair together with a non-dummy agent (see Lemma 7). Now, for each preference list of a non-dummy agent a, if we want to restrict that agent a' at rank i cannot be swapped with agent a'' ranked directly behind a' in the preferences of a, we place the $\ell+1$ dummy agents with indices $(i-1)(\ell+1)+1$ to $i(\ell+1)$ of opposite gender between a' and a'' in the preference list of a. Thus, the given budget never suffices to swap agents a' and a'' in the preference list of a. Moreover, if we want to introduce a non-unit cost $2 \leq c \leq \ell$ of swapping the first and the second agent in some agent's preference list, then we put the dummy agents 1 to $c-1$ of opposite gender between its most preferred and second-most preferred agent in its preference list. We use this variant of *Swap* (and a more restrictive version of it) in the proofs of Theorem 3 and Proposition 9.

***Add* via Restricted *Swap*.** It is possible to model *Add* actions by (restricted) *Swap* actions by modifying a given SM instance $(A = U \cup W, \mathcal{P})$ with sets $U_{\text{add}} \subseteq U$ and $W_{\text{add}} \subseteq W$ as follows. We keep all agents $a \in A$ from the original instance and introduce for each agent $a \in U_{\text{add}} \cup W_{\text{add}}$ one agent a' of the opposite gender and one agent a'' of the same gender with preferences $a' : a \succ a'' \succ \overset{\text{(rest)}}{\dots}$ and $a'' : a' \succ \overset{\text{(rest)}}{\dots}$. Moreover, we put for each $a \in U_{\text{add}} \cup W_{\text{add}}$ agent a' as the top-choice of a. All preference lists are completed by appending the remaining agents in an arbitrary order at the end. Now, we introduce dummy agents such that the only allowed swaps are swapping a with a'' in a''s preference list for some $a \in U_{\text{add}} \cup W_{\text{add}}$. Not adding an agent $a \in U_{\text{add}} \cup W_{\text{add}}$ corresponds to leaving the preferences of a' unchanged, which results in $\{a', a\}$ being part of every stable matching. Adding an agent $a \in U_{\text{add}} \cup W_{\text{add}}$ corresponds to modifying the preferences of a' by swapping a and a'', which results in $\{a', a''\}$ being part of every stable matching. In this case, a is now able to pair up with other agents from the original instance. We prove the correctness of this transformation in Lemma 6 and use it in the proof of Theorem 3.

***Reorder* via Restricted *Swap*.** It is possible to model *Reorder* actions by (restricted) *Swap* actions. To do so, we construct a new instance from a given SM instance with agent set A and a given budget ℓ as follows. First of all, we keep all agents $a \in A$. Moreover, for each agent $a \in A$, we add a copy a' with the same preferences as well as a *binding agent* $\widetilde{a}$ of opposite gender with preferences $\widetilde{a} : a' \succ a \succ \overset{\text{(rest)}}{\dots}$. We adjust the budget to $\ell' = \ell \cdot (4n^2 + 2n)$. For each $a \in A$, we modify the preferences of agents a and a' by inserting $\widetilde{a}$ as their top-choice. Moreover, for each agent b of the opposite gender of a in

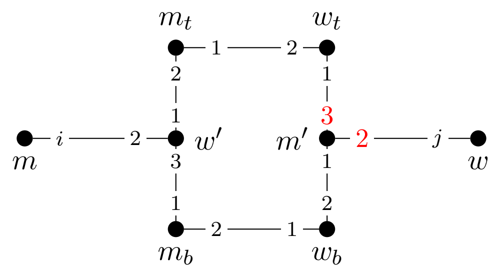


Figure 4: Gadget to model *DeleteAcceptability* by *Swap*. Swapping w and w^t in the preferences of m' (marked in red) corresponds to deleting edge $\{m, w\}$.

the constructed instance, we insert the copy a' directly after the agent a in the preferences of b. All preference lists are completed arbitrarily. We now only allow to swap the first two agents in the preferences of $\widetilde{a}$ at cost $4n^2$ and to swap all agents except $\widetilde{a}$ in the preferences of a' at unit cost. The general idea of the construction is that only one of a and a' can be free to pair up with a non-binding agent, as the other is matched to the corresponding binding agent in all stable matchings. We cannot change the preferences of a at all, while we can change the preferences of a' at cost $2n$ arbitrarily (except its top-choice). Initially, a' is always matched to the binding agent but can be "freed" by modifying the preferences of $\widetilde{a}$ at cost $4n^2$. Reordering the preferences of an agent a in the original instance then corresponds to freeing a' and reordering the preferences of a' arbitrarily (except its top-choice which is irrelevant in this case). Overall, we can free at most ℓ agents a', while for each of them we can fully reorder the relevant part of their preferences. We use a similar construction in the proof of Proposition 9.

***DeleteAcceptability* via Restricted *Swap*.** It is possible to model deleting the acceptability of two agents by performing (restricted) swaps. To do so, we modify the given SM instance by introducing for each man-woman pair $\{m, w\}$ with $m \in U$ and $w \in W$ where m ranks w at position i and w ranks m at position j the gadget depicted in Figure 4 (note that this gadget was introduced by Cechlárová and Fleiner (2005) to model parallel edges in a Stable Marriage instance). Moreover, we only allow swapping w and w_t in the preferences of some man m' (as indicated in Figure 4). Matching m to w in the original instance corresponds to matching m to w' and m' to w in the modified instance. Note that it is never possible that only one of $\{m, w'\}$ and $\{m', w\}$ is part of a stable matching. Deleting the acceptability of a pair $\{m, w\}$ now corresponds to swapping w and w_t in m''s preference relation, as in this case neither $\{m, w'\}$ nor $\{m', w\}$ can be part of any stable matching.

Summarizing, we conclude that performing swaps is, in some sense, the most powerful manipulative action considered, as all other actions can be modeled using it. However, this does not imply that if one of our computational problems is computationally hard for some manipulative action, then it is also hard for *Swap* since, for example, for the *Exact-Exists* setting a modified problem definition is used for the manipulative actions *Add* and *Delete*.

3. *Constructive-Exists*

In this section, we analyze the computational complexity of CONSTRUCTIVE-EXISTS-$\mathcal{X}$. In Section 3.1, we start with showing intractability for $\mathcal{X} \in \{$*Add*, *Swap*, *DeleteAcceptability*, *Reorder*$\}$. We complement these intractability results with an XP-algorithm for CONSTRUCTIVE-EXISTS-REORDER (for the other manipulative actions, an XP-algorithm is trivial) in Section 3.2. Subsequently, in Section 3.3, we show that CONSTRUCTIVE-EXISTS-DELETE is solvable in $\mathcal{O}(n^2)$ time, and CONSTRUCTIVE-EXISTS-REORDER admits a 2-approximation with the same running time.

3.1 A Framework for Computational Hardness

All our W[1]-hardness results for *Constructive-Exists* essentially follow from the same basic idea for a parameterized reduction. We now explain the general framework of the reduction, using the manipulative action *Add* as an example. The modifications needed to transfer the approach to the manipulative actions *Swap*, *DeleteAcceptability*, and *Reorder* are described afterwards.

We construct a parameterized reduction from CLIQUE, where given an undirected graph $G = (V, E)$ and an integer k, the question is whether G admits a size-k clique, i.e., a set of k vertices that are pairwise adjacent. Parameterized by k, CLIQUE is W[1]-hard (Downey & Fellows, 2013). Fix an instance $(G = (V, E), k)$ of CLIQUE and denote the set of vertices by $V = \{v_1, \ldots, v_{|V|}\}$ and the set of edges by $E = \{e_1, \ldots, e_{|E|}\}$. Let d_v denote the degree of vertex v. Moreover, let $e_1^v, \ldots, e_{d_v}^v$ be a list of all edges incident to v.

The high-level idea is as follows. We start by introducing two agents m^* and w^*, and the edge $\{m^*, w^*\}$ is the edge which shall be contained in a stable matching. Furthermore, we add $q := \binom{k}{2}$ women $w_1^\dagger, \ldots, w_q^\dagger$, which we call *penalizing women*. The idea is that m^* prefers every penalizing woman to w^*, and thereby, a stable matching containing the edge $\{m^*, w^*\}$ can only exist if every penalizing woman $w_j^\dagger$ is matched to a man she prefers to m^*, as otherwise $\{m^*, w_j^\dagger\}$ would be a blocking pair for any matching containing $\{m^*, w^*\}$. In addition, we introduce one vertex gadget for every vertex and one edge gadget for every edge; these differ for the different manipulative actions. Each vertex gadget includes a *vertex woman* and each edge gadget an *edge man*: A penalizing woman can only be matched to an edge man in a stable matching containing $\{m^*, w^*\}$. However, an edge man can only be matched to a penalizing woman if the gadgets corresponding to the endpoints of the edge and the gadget corresponding to the edge itself are manipulated. Thus, one has to perform manipulations in at least $\binom{k}{2}$ edge gadgets and in all vertex gadgets corresponding to the endpoints of these edges. In this way, a budget of $\ell = k + \binom{k}{2}$ suffices if and only if G contains a clique of size k.

3.1.1 *Add*

We now give the details of the parameterized reduction (following the general approach sketched before) for the manipulative action *Add*. For each vertex $v \in V$, we introduce a vertex gadget consisting of one vertex woman w_v and two men m'_v and m_v. For each edge $e \in E$, we introduce an edge gadget consisting of an edge man m_e, one man m'_e, and one woman w_e. Additionally, we introduce a set of k women $\widetilde{w}_1, \ldots, \widetilde{w}_k$. The agents that can be added are $U_{\text{add}} := \{m'_v : v \in V\} \cup \{m'_e : e \in E\}$ and $W_{\text{add}} := \emptyset$, while all other agents are part of the original instance. We set the budget $\ell := k + \binom{k}{2}$. (Note that we will show that the reduction, in fact, works even if $\ell = \infty$.)

In this reduction, adding the man m'_v for some $v \in V$ corresponds to manipulating the corresponding vertex gadget, whereas adding m'_e for some $e \in E$ corresponds to manipulating the corresponding edge gadget. We call the constructed Constructive-Exists-Add instance $\mathcal{I}_{\text{add}}$.

For each vertex $v \in V$ that is incident to edges $e_1^v, \dots, e_{d_v}^v$, the preferences of the agents from the corresponding vertex gadget are as follows:

$$w_v : m'_v \succ m_{e_1^v} \succ \dots \succ m_{e_{d_v}^v} \succ m_v \succ \overset{\text{(rest)}}{\dots}, \qquad m'_v : w_v \succ \overset{\text{(rest)}}{\dots},$$
$$m_v : w_v \succ \widetilde{w}_1 \succ \dots \succ \widetilde{w}_k \succ w^* \succ \overset{\text{(rest)}}{\dots}.$$

For each edge $e = \{u, v\} \in E$, the agents from the corresponding edge gadget have the following preferences:

$$w_e : m'_e \succ m_e \succ \overset{\text{(rest)}}{\dots}, \qquad m_e : w_e \succ w_u \succ w_v \succ w_1^\dagger \succ \dots \succ w_q^\dagger \succ \overset{\text{(rest)}}{\dots},$$
$$m'_e : w_e \succ \overset{\text{(rest)}}{\dots}.$$

Lastly, the agents m^* and w^* and, for $i \in [q]$ (recall that $q = \binom{k}{2}$) and $t \in [k]$, the agents $w_i^\dagger$ and $\tilde{w}_t$ have the following preferences:

$$\widetilde{w}_t : m_{v_1} \succ \dots \succ m_{v_{|V|}} \succ \overset{\text{(rest)}}{\dots}, \qquad w_i^\dagger : m_{e_1} \succ \dots \succ m_{e_{|E|}} \succ m^* \succ \overset{\text{(rest)}}{\dots},$$
$$m^* : w_1^\dagger \succ \dots \succ w_q^\dagger \succ w^* \succ \overset{\text{(rest)}}{\dots}, \qquad w^* : m_{v_1} \succ \dots \succ m_{v_{|V|}} \succ m^* \succ \overset{\text{(rest)}}{\dots}.$$

Note that in all stable matchings containing $\{m^*, w^*\}$, every penalizing woman $w_i^\dagger$ is matched to a man she prefers to m^* and every man m_v is matched to a woman which he prefers to w^*, as otherwise the matching is blocked by $\{m^*, w_i^\dagger\}$ or $\{m_v, w^*\}$. This ensures that at most k men m'_v can be added to the instance (which will be used later to show that the reduction also works if $\ell = \infty$), as there exist only k women $\widetilde{w}_i$ that can be matched to some m_v from a manipulated vertex gadget. Parts of the construction are visualized in Figure 5.

Note that in the instance as described above there are $2|V|+2|E|+1$ men and $|V|+|E|+k+q+1$ women. However, our definition of Stable Marriage requires the instance to have the same number of men and women. This can be achieved by adding $|V|+|E|-k-q$ *filling women* (to W_{orig}). These filling women have arbitrary preferences and every man prefers any non-filling woman to any filling woman. The presence or absence of filling women does not change the existence of a stable matching containing $\{m^*, w^*\}$ because every stable matching in the presence of filling women can be transformed into a stable matching in the absence of filling women by deleting all edges incident to a filling woman. Moreover, every stable matching M in the absence of filling women can be transformed to a stable matching in the presence of filling women by adding to M a stable matching between the set of men unassigned by M and filling women (note that such a stable matching has to exist since every Stable Marriage instance admits a stable matching). In order to

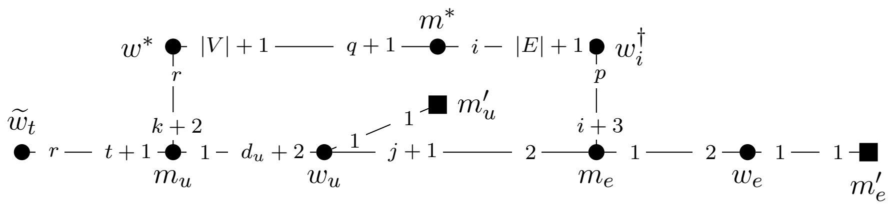


Figure 5: A vertex gadget and an edge gadget for the hardness reduction for Add, where $e = e^u_j = e_p$ and $u = v_r$. The squared vertices are the vertices from U_{add} that can be added to the instance. In the figure, we only exemplarily depict one penalizing woman $w_i^\dagger$ for some $i \in [q]$ and one woman $\widetilde{w}_t$ for some $t \in [k]$. For an edge $\{x, y\}$, the number on this edge closer to x indicates the rank of y in x's preference order.

keep the proof of correctness of the reduction simpler, we will ignore all filling women and assume they are not part of the instance.

Lemma 1. *If G contains a clique of size k, then $\mathcal{I}_{\text{add}}$ is a YES-instance.*

Proof. Let C be a clique in G. For an edge $e = \{u, v\} \in E$, we write $e \subseteq C$ to express that e lies in C, i.e., $u \in C$ and $v \in C$. Further, let $C[i]$ denote the vertex with i-th lowest index in C and $D[i]$ the edge with i-th lowest index in C. We add the ℓ agents $\{m'_v : v \in C\}$ and $\{m'_e : e \subseteq C\}$, and claim that

$$\begin{aligned} M := &\{\{m^*, w^*\}\} \cup \{\{m'_v, w_v\} : v \in C\} \cup \{\{m_{C[i]}, \widetilde{w}_i\} : i \in [k]\} \cup \\ &\{\{m_v, w_v\} : v \in V \setminus C\} \cup \{\{m_e, w_e\} : e \not\subseteq C\} \cup \\ &\{\{m'_e, w_e\} : e \subseteq C\} \cup \{\{m_{D[i]}, w_i^\dagger\} : i \in [q]\} \end{aligned}$$

is a stable matching, containing $\{m^*, w^*\}$. We now iterate over all agents present in the instance after adding $\{m'_v : v \in C\}$ and $\{m'_e : e \subseteq C\}$ and argue why they cannot be part of a blocking pair. Let $A' := U_{\text{orig}} \cup W_{\text{orig}} \cup \{m'_v : v \in C\} \cup \{m'_e : e \subseteq C\}$ be the agents contained in the instance arising through the addition of $\{m'_v : v \in C\} \cup \{m'_e : e \subseteq C\}$.

First, note that for each $e \not\subseteq C$ the agents m_e and w_e are matched to their top-choice in the instance and, therefore, cannot be part of a blocking pair.

For each vertex $v \in V \setminus C$, man m_v is matched to his first choice and thus is not part of a blocking pair. Since $v \in V \setminus C$, every edge e incident to v is not contained in the clique, and consequently, m_e is not part of a blocking pair. As all agents in A' which w_v prefers to m_v are not part of a blocking pair, also w_v is not part of a blocking pair.

For each vertex $v \in C$, the agents m'_v and w_v are matched to their top-choice and thus are not part of a blocking pair. Consider a man m_v for some $v \in C$. This man is matched to woman $\widetilde{w}_j$ for some $j \in [k]$. Edge $\{m_v, w_v\}$ is not blocking, as w_v is not part of a blocking pair. Moreover, there cannot exist a blocking pair of the form $\{m_v, \widetilde{w}_i\}$ for some $i \in [k]$, as m_v only prefers women $\widetilde{w}_i$ with $i < j$. However, all women $\widetilde{w}_i$ with $i < j$ prefer their current partner to m_v, as they are all assigned a man corresponding to a vertex with a smaller index than v.

Recall that there cannot exist a blocking pair involving an agent from $\{w_v : v \in V\}$. Thus, since all agents from $\{m_e : e \subseteq C\}$ have the same preferences over the penalizing women, and the penalizing women prefer each man from $\{m_e : e \subseteq C\}$ to m^*, there is no blocking pair involving agents from $\{m_e : e \subseteq C\} \cup \{w_i^\dagger : i \in [q]\}$.

Finally, neither m^* nor w^* are part of a blocking pair, as all agents which they prefer to each other (i.e., penalizing women $w_i^\dagger$ or men m_v) are not contained in a blocking pair. Thus, M is stable. Note that if the k vertices from C were not to form a clique, then the set $\{m_e : e \subseteq C\}$ would consist of less than $\binom{k}{2}$ men. Thus, not all penalizing women are matched to an edge man in M which implies that m^* and a penalizing woman form a blocking pair for M. □

We now proceed with the backward direction.

Lemma 2. *If there exists a set X_A of agents (of arbitrary size) such that after their addition there exists a stable matching containing $\{m^*, w^*\}$, then G contains a clique of size k.*

Proof. Let M be a stable matching containing $\{m^*, w^*\}$. Since the edges $\{m^*, w_i^\dagger\}$ are not blocking, all penalizing women are matched to an edge man m_e for some $e \in E$. This requires that $m'_e \in X_A$, as otherwise $\{m_e, w_e\}$ is a blocking pair. Moreover, for each such edge $e = \{u, v\}$, the vertex women w_u and w_v have to be either matched to other edge men or to the men m'_u or m'_v. Note that in both cases, the corresponding agents m_u and m_v are matched to one of the women $\widetilde{w}_i$, as otherwise $\{m_u, w^*\}$ or $\{m_v, w^*\}$ is a blocking pair. Thus, there exist at most k vertices $v \in V$ where w_v is matched to an edge men or to m'_v.

Since there are $\binom{k}{2}$ penalizing women, and each of them is matched to an edge man, it follows that these edge men correspond to the edges in the clique formed by the k vertices $v \in V$ where w_v is either matched to an edge men or m'_v. □

From Lemma 1 and Lemma 2, we conclude that there exists a parameterized reduction from Clique parameterized by k to Constructive-Exists-Add parameterized by ℓ, implying the following.

Theorem 1. *Parameterized by the budget ℓ, it is W[1]-hard to decide whether* Constructive-Exists-Add *has a solution with at most ℓ additions or has no solution with an arbitrary number of additions, even if we are only allowed to add agents of one gender.*

3.1.2 *DeleteAcceptability* and *Reorder*

The reduction for *Add* presented in Section 3.1.1 cannot be directly applied for *DeleteAcceptability* and *Reorder*. Instead, new vertex and edge gadgets need to be constructed. One reason for this is that for *Add* (and *Swap*), we could ensure that the penalizing women are not manipulated. However, they can be manipulated by *DeleteAcceptability* and *Reorder* operations, and, therefore, in our construction for *Add*, there exists an easy solution with q manipulations (recall that $q = \binom{k}{2}$), which just deletes the acceptabilities $\{m^*, w_i^\dagger\}$ in the case of *DeleteAcceptability* or moves m^* to the end of $w_i^\dagger$'s preferences for each $i \in [q]$ in the case of *Reorder*. To avoid such solutions, we modify the construction for *Add* as follows. We add q additional penalizing men $m_1^\dagger, \ldots, m_q^\dagger$, and one manipulation of an edge

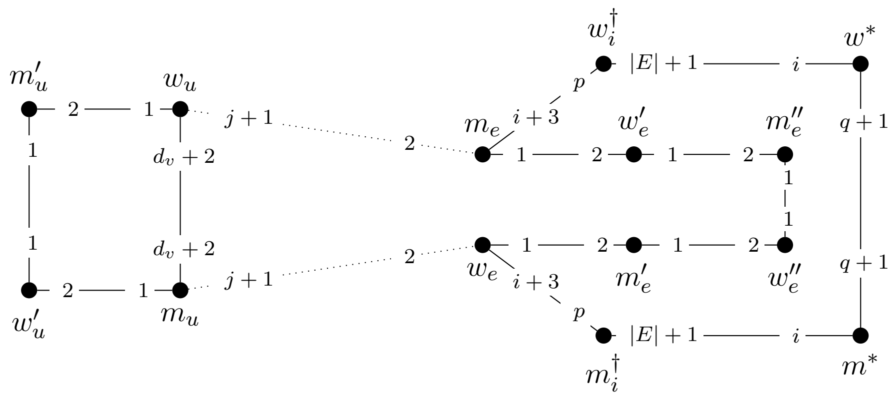


Figure 6: Visualization of the reduction for *DeleteAcceptability* and *Reorder*. For some edge $e_p = \{u, v\} \in E$, the edge gadget corresponding to e_p and the vertex gadget corresponding to u (assuming that $e_p = e^u_j$) are included as well as m^* and w^* together with penalizing agents $m_i^\dagger$ and $w_i^\dagger$ for some arbitrary $i \in [q]$. Edges between the vertex and the edge gadget are dotted.

gadget will now allow to match both a penalizing woman and a penalizing man to this edge gadget. We assume without loss of generality that $k \geq 6$, as this makes the proof of Lemmas 4 and 5 easier.

We now describe the details of the construction of the SM instance that we want to manipulate, which is the same for manipulative actions *Reorder* and *DeleteAcceptability*. Given an instance of Clique consisting of a graph $G = (V, E)$ and an integer k, for each vertex $v \in V$ we introduce a gadget consisting of one vertex woman w_v and one vertex man m_v together with one woman w'_v and one man m'_v. The preferences are as follows:

$$w_v : m'_v \succ m_{e_1^v} \succ \cdots \succ m_{e_{d_v}^v} \succ m_v \succ \overset{\text{(rest)}}{\dots}, \qquad w'_v : m'_v \succ m_v \succ \overset{\text{(rest)}}{\dots},$$

$$m_v : w'_v \succ w_{e_1^v} \succ \cdots \succ w_{e_{d_v}^v} \succ w_v \succ \overset{\text{(rest)}}{\dots}, \qquad m'_v : w'_v \succ w_v \succ \overset{\text{(rest)}}{\dots}.$$

For each edge $e = \{u, v\} \in E$, we introduce a gadget consisting of one edge man m_e, one edge woman w_e together with two men m'_e and m''_e, and two women w'_e and w''_e. The preferences are as follows:

$$m_e : w'_e \succ w_u \succ w_v \succ w_1^\dagger \succ \cdots \succ w_q^\dagger \succ \overset{\text{(rest)}}{\dots}, \qquad m'_e : w''_e \succ w_e \succ \overset{\text{(rest)}}{\dots},$$

$$w_e : m'_e \succ m_u \succ m_v \succ m_1^\dagger \succ \cdots \succ m_q^\dagger \succ \overset{\text{(rest)}}{\dots}, \qquad w'_e : m''_e \succ m_e \succ \overset{\text{(rest)}}{\dots},$$

$$m''_e : w''_e \succ w'_e \succ \overset{\text{(rest)}}{\dots}, \qquad w''_e : m''_e \succ m'_e \succ \overset{\text{(rest)}}{\dots}.$$

See Figure 6 for an example of a vertex gadget and an edge gadget. The preferences of the agents m^*, w^* and the penalizing agents are as follows:

$$w_i^\dagger : m_{e_1} \succ \cdots \succ m_{e_{|E|}} \succ m^* \succ \overset{\text{(rest)}}{\dots}, \qquad m^* : w_1^\dagger \succ \cdots \succ w_q^\dagger \succ w^* \succ \overset{\text{(rest)}}{\dots},$$

$$m_i^\dagger : w_{e_1} \succ \cdots \succ w_{e_{|E|}} \succ w^* \succ \overset{\text{(rest)}}{\dots}, \qquad w^* : m_1^\dagger \succ \cdots \succ m_q^\dagger \succ m^* \succ \overset{\text{(rest)}}{\dots}.$$

Finally, we set $\ell := \binom{k}{2} + k$. By $\mathcal{I}_{\text{del}}$ we denote the resulting instance of Constructive-Exists-DeleteAcceptability and by $\mathcal{I}_{\text{reor}}$ the resulting instance of Constructive-Exists-Reorder. In the following, in Lemma 3, we prove the forward direction of the reduction for both *Reorder* and *DeleteAcceptability*. In Lemma 4, we prove the correctness of the backward direction for *Reorder* and in Lemma 5 for *DeleteAcceptability*.

Lemma 3. *If G contains a clique of size k, then $\mathcal{I}_{\text{del}}$ and $\mathcal{I}_{\text{reor}}$ are YES-instances.*

Proof. Let $C \subseteq V$ be a clique. We denote by $D[i]$ the edge with i-th lowest index in the clique and for an edge $e = \{u, v\} \in E$, we write $e \subseteq C$ if $u \in C$ and $v \in C$. In $\mathcal{I}_{\text{del}}$, we delete the following ℓ edges. For each $v \in C$, we delete $\{m'_v, w'_v\}$, and for each edge $e \subseteq C$, we delete $\{m''_e, w''_e\}$. In $\mathcal{I}_{\text{reor}}$, we manipulate the following ℓ agents. For each $v \in C$, we change the preferences of m'_v to $m'_v : w_v \succ \overset{\text{(rest)}}{\dots}$. For each edge $e \subseteq C$, we change the preferences of m''_e to $m''_e : w'_e \succ \overset{\text{(rest)}}{\dots}$.

We claim that the matching

$$\begin{aligned} M := &\{\{m^*, w^*\}\} \cup \{\{m_v, w'_v\}, \{m'_v, w_v\} : v \in C\} \cup \{\{m'_v, w'_v\}, \{m_v, w_v\} : v \in V \setminus C\} \cup \\ &\{\{m_e, w'_e\}, \{m'_e, w_e\}, \{m''_e, w''_e\} : e \not\subseteq C\} \cup \{\{m'_e, w''_e\}, \{m''_e, w'_e\} : e \subseteq C\} \cup \\ &\{\{m_{D[i]}, w^\dagger_i\}, \{m^\dagger_i, w_{D[i]}\} : i \in [q]\}, \end{aligned}$$

which contains $\{m^*, w^*\}$, is stable in both modified instances.

To see this, first note that for each $e \not\subseteq C$, the agents m_e, m''_e, w_e, and w''_e are matched to their top-choices, and therefore are not part of a blocking pair. As a consequence, also m'_e and w'_e cannot be part of a blocking pair.

For each agent $v \in V \setminus C$, both m'_v and w'_v are matched to their top-choices and thus not part of a blocking pair. The only agents which m_v prefers to w_v are w'_v and w_e for every edge $e = \{v, u\} \in E$ incident to v; however, we showed for all these agents that they are not part of a blocking pair. Thus, m_v is not part of a blocking pair. Symmetrically, the only agents which w_v prefers to m_v are m'_v and m_e for every edge $e = \{v, u\} \in E$ incident to v, and also these agents are not contained in a blocking pair. Therefore, also w_v is not part of a blocking pair.

For each agent $v \in C$, the agents m_v, m'_v, and w_v are matched to their top-choices and thus are not part of a blocking pair (note that for both *Reorder* and *DeleteAcceptability*, we have modified the preferences of m'_v such that w_v is his top-choice). In *DeleteAcceptability*, also woman w'_v is matched to her top-choice. In *Reorder*, woman w'_v is matched to her second choice, while her top-choice m'_v is not part of a blocking pair. Thus, w'_v is also not part of a blocking pair.

Since all agents from $\{m_e : e \subseteq C\}$ have the same preferences over the penalizing women, and the penalizing women prefer each man from $\{m_e : e \subseteq C\}$ to m^*, there is no blocking pair involving only agents from $\{m_e : e \subseteq C\} \cup \{m^*\} \cup \{w^\dagger_i : i \in [q]\}$. Symmetrically, it follows that no blocking pair involves only agents from $\{w_e : e \subseteq C\} \cup \{w^*\} \cup \{m^\dagger_i : i \in [q]\}$. Thus, M is stable. Note that if the k vertices from C were not to form a clique, then the set $\{m_e : e \subseteq C\}$ would consist of less than $\binom{k}{2}$ men. Thus, not all penalizing women are matched to an edge men in M which implies that m^* and a penalizing women form a blocking pair for M. $\square$

We now prove the backward direction for *Reorder*.

Lemma 4. *If $\mathcal{I}_{\text{reor}}$ is a YES-instance, then G contains a clique of size k.*

Proof. Let X_{reor} be the set of at most ℓ agents whose preferences have been reordered, and let M be a stable matching containing $\{m^*, w^*\}$. In the following, we call a vertex agent a_v *unhappy* if it is neither contained in X_{reor} nor matched to one of its $d_v + 1$ most preferred partners, i.e., it prefers all edge agents of edges incident to v to its current partner. Note that for each edge gadget for an edge $e \in E$ such that no agent from the edge gadget is contained in X_{reor}, every stable matching contains the edges $\{m_e, w'_e\}$, $\{m'_e, w_e\}$, and $\{m''_e, w''_e\}$. If an edge agent $a_e \in \{m_e, w_e\}$ is the only agent from this edge gadget contained in X_{reor}, then matching M contains the edge $\{m_e, w'_e\}$ if $a_e = w_e$ and $\{w_e, m'_e\}$ if $a_e = m_e$. Furthermore, each penalizing agent $a_i^\dagger$ needs to be contained in X_{reor} or matched to an edge agent a_e since, otherwise, $\{m_i^\dagger, w^*\}$ if $a_i^\dagger = m_i^\dagger$ or $\{m^*, w_i^\dagger\}$ if $a_i^\dagger = w_i^\dagger$ blocks M.

Let p be the number of agents from X_{reor} which are edge agents matched to a penalizing agent or are penalizing agents. As there are $2\binom{k}{2}$ penalizing agents, at least $2\binom{k}{2} - p$ of them need to be matched to edge agents which are not in X_{reor}, as otherwise at least one penalizing agent forms a blocking pair together with m^* or w^*. Since, as argued above, matching an edge agent to a penalizing agent requires that at least one agent from the corresponding edge gadget is part of X_{reor} (and it is not sufficient to reorder the preferences of the other edge agent from this gadget), it follows that at least $\frac{2\binom{k}{2}-p}{2} + p = \binom{k}{2} + \frac{p}{2} \leq \ell$ agents from X_{reor} are agents from edge gadgets or penalizing agents. As $\ell = \binom{k}{2} + k$, it follows that $p \leq 2k$.

For every vertex agent $a_v \in \{m_v, w_v\}$ it holds that if none of the vertices from its vertex gadget is contained in X_{reor}, then agent a_v is either matched to an edge agent or unhappy. As all but at most $k - \frac{p}{2}$ agents from X_{reor} are either a penalizing agent or contained in an edge gadget, at least $2\binom{k}{2} - p$ edge agents are matched to penalizing agents. Because an edge agent may only be matched to an agent outside its edge gadget if the preferences of at least one agent from the gadget are modified, it follows that there can be at most $2k - p$ happy vertex agents. Thus, without loss of generality, there are at most $k - \frac{p}{2}$ happy vertex men (otherwise we apply the following argument for happy vertex women).

Note that at least $\binom{k}{2} - p$ penalizing men are matched to edge women which are not contained in X_{reor} in M. These $\binom{k}{2} - p$ edge women w_e for some $e \in E$ prefer the vertex men corresponding to the endpoints of e to each penalizing man. Since only the at most $k - \frac{p}{2}$ happy vertex men may not prefer to be matched to an edge women corresponding to an incident edge, it follows that the $\binom{k}{2} - p$ edges to which the $\binom{k}{2} - p$ edge women w_e correspond have at most $k - \frac{p}{2}$ endpoints. This is only possible if $\binom{k}{2} - p \leq \binom{k-\frac{p}{2}}{2}$, which is equivalent to $0 \leq p \cdot (p - 4k + 10)$, implying (as $p \geq 0$) that $p = 0$ or $p \geq 4k - 10$. However, the latter case combined with our previous observation that $p \leq 2k$ implies that $2k \geq p \geq 4k - 10$, a contradiction for $k \geq 6$. It follows that $p = 0$. This implies that indeed k vertex men are happy, no penalizing woman has been manipulated, and that in $\binom{k}{2}$ edge gadgets exactly one agent, but no edge agent has been manipulated. Thus, each penalizing woman is matched to an edge man m_e such that both endpoints of e are happy. It follows that there are $\binom{k}{2}$ edges whose endpoints are among the k vertices whose vertex man m_v is happy. These k vertices clearly form a clique. $\square$

In a similar way, we can show the analogous statement for $\mathcal{I}_{\text{del}}$.

Lemma 5. *If $\mathcal{I}_{\text{del}}$ is a YES-instance, then G contains a clique of size k.*

Proof. Let X_{del} be the set of at most ℓ pairs which have been deleted, and let M be a stable matching containing $\{m^*, w^*\}$. Note that, for each edge gadget for an edge e such that no pair from the edge gadget is contained in X_{del}, any stable matching contains the edges $\{m_e, w'_e\}$, $\{m'_e, w_e\}$, and $\{m''_e, w''_e\}$. For each penalizing agent $a_i^\dagger$, either the edge $\{a_i^\dagger, a^*\}$ where a^* is the agent from $\{m^*, w^*\}$ of opposite gender is contained in X_{del} or $a_i^\dagger$ is matched to an edge agent a_e. Let p be the number of pairs in X_{del} containing a penalizing agent and an agent from $\{m^*, w^*\}$. As there are $2\binom{k}{2}$ penalizing agents, at least $\frac{2\binom{k}{2}-p}{2} + p = \binom{k}{2} + \frac{p}{2} \leq \ell = \binom{k}{2} + k$ deletions happen where either both involved agents are from the same edge gadget or one of the involved agents is a penalizing agent and the other one is from $\{m^*, w^*\}$. From this, it follows that $p \leq 2k$.

For every vertex agent a_v it holds that if no pair in the corresponding vertex gadget is contained in X_{del}, then the agent a_v is either matched to an edge agent or unhappy (where a_v is unhappy if none of the edges incident to it is deleted and it is not matched to one of its first $d_v + 1$ most preferred partners). Since, as argued above, all but $k - \frac{p}{2}$ of the deleted pairs do not involve a vertex agent, and at least $2\binom{k}{2} - p$ edge agents are matched to penalizing agents, there can be at most $2k - p$ happy vertex agents. The rest of the argument is the same as in the proof of Lemma 4. $\square$

The following theorem follows directly from Lemmas 3 to 5.

Theorem 2. *Parameterized by the budget ℓ,* Constructive-Exists-DeleteAcceptability *is W[1]-hard. Parameterized by ℓ,* Constructive-Exists-Reorder *is W[1]-hard, and this also holds if one is only allowed to reorder the preferences of agents of one gender.*

Note that the presented construction, in contrast to the reduction for *Add* and *Swap*, does not have implications in terms of inapproximability of *Reorder* and *DeleteAcceptability*. In particular, as described in the beginning of this section, there always exists a trivial solution of cost $2q$. In fact, we show in the next section that *Reorder* admits a factor-2 approximation. Note further that the presented construction is also a valid parameterized reduction from Clique to Constructive-Exists-Swap; however, we will derive a stronger hardness result for this problem (yielding also FPT-inapproximability) in Section 3.1.3.

3.1.3 *Swap*

Theorem 1 showed that it is W[1]-hard to distinguish whether it is possible to make an edge $e^* = \{m^*, w^*\}$ part of a stable matching or no set of agents whose addition makes e^* part of a stable matching exists. We now use this W[1]-hardness to derive an FPT-inapproximability result for Constructive-Exists-Swap by a reduction from Constructive-Exists-Add. We achieve this result in two steps. First, we consider a variant of *Swap*, which we call *SwapRestricted*: Here, a subset $\hat{A}$ of agents is given, and one is only allowed to swap the first two agents in the preference lists of agents from $\hat{A}$, while the preferences of all agents $A \setminus \hat{A}$ need to remain unmodified. By reducing from Constructive-Exists-Add, we show that parameterized by the budget ℓ, it is W[1]-hard to decide whether Constructive-Exists-SwapRestricted has a solution with at most ℓ swaps or has no solution with an arbitrary number of allowed swaps. Second, we derive our FPT-inapproximability result for Constructive-Exists-Swap by reducing from Constructive-Exists-SwapRestricted.

We start by showing W[1]-hardness of Constructive-Exists-SwapRestricted.

Lemma 6. *Parameterized by the budget ℓ, it is W[1]-hard to decide whether* Constructive-Exists-SwapRestricted *has a solution with at most ℓ swaps or has no solution with an arbitrary number of allowed swaps.*

Proof. We reduce from Constructive-Exists-Add, for which it is W[1]-hard to distinguish whether there is a solution with at most ℓ additions or no solution for any number of additions (Theorem 1). Let $(\mathcal{I}, U_{\text{add}}, W_{\text{add}}, \ell)$ be an instance of Constructive-Exists-Add. We form an equivalent instance of Constructive-Exists-SwapRestricted by adding for each man $m \in U_{\text{add}}$ a woman w_m and a man m', where w_m is added to m's preferences as the top-choice. The man m' has w_m as his top-choice and the woman w_m has m as her top-choice and m' as her second most preferred man. All other agents follow in an arbitrary order in the preferences of w_m and m'. All other agents add w_m or m' at the end of their preferences. Similarly, we add for each woman $w \in W_{\text{add}}$ two agents m_w and w' whose preferences are constructed analogously. We set $\hat{A} := \{m_w : w \in W_{\text{add}}\} \cup \{w_m : m \in U_{\text{add}}\}$.

It remains to show that the instances are equivalent. From a solution to Constructive-Exists-Add, consisting of the agents $X_U \cup X_W$, one can get a solution to Constructive-Exists-SwapRestricted by swapping the two most preferred men m and m' in the preferences of w_m for each $m \in X_U$ and modifying the preferences of m_w for each $w \in X_W$ analogously. Vice versa, from a solution to Constructive-Exists-SwapRestricted modifying the preferences of a set of agents X one can get a solution to Constructive-Exists-Add by adding $w \in W_{\text{add}}$ if $m_w \in X_U$ and $m \in U_{\text{add}}$ if $w_m \in X_W$. It is now straightforward to verify the correctness. □

We continue by reducing *SwapRestricted* to *Swap*. The basic idea behind this reduction is that we can introduce a set of dummy agents and use these agents to make the swaps not allowed in the *SwapRestricted* instance too expensive. First, we show that we can add a set of $r > \ell$ men $m_1^d, \ldots, m_r^d$ and r women $w_1^d, \ldots, w_r^d$ to an SM instance such that, in any instance arising through at most ℓ swaps, any stable matching contains the edges $\{m_i^d, w_i^d\}$ for all $i \in [r]$ regardless where the newly inserted agents are placed in the preferences of the other agents. This allows us to make swapping two neighboring non-dummy agents a and a' in the preference list of some non-dummy agent b very expensive, as we can insert (some of) the newly added dummy agents in between a and a' in b's preference list.

Given an SM instance $\mathcal{I}$ and a list of swap operations S, we denote by $\mathcal{I}[S]$ the SM instance resulting from applying the swaps from S to $\mathcal{I}$.

Lemma 7. *Let $\mathcal{I}$ be a* Stable Marriage *instance containing r men $m_1^d, \ldots, m_r^d$ and r women $w_1^d, \ldots, w_r^d$ such that, for all $i \in [r]$, the preferences of w_i^d and m_i^d match* *the following pattern (where all indices are taken modulo r):*

$$m_i^d : w_i^d \succ w_{i+1}^d \succ w_{i+2}^d \succ \cdots \succ w_{r+i-1}^d \succ \overset{\text{(rest)}}{\cdots},$$
$$w_i^d : m_i^d \succ m_{i+1}^d \succ m_{i+2}^d \succ \cdots \succ m_{r+i-1}^d \succ \overset{\text{(rest)}}{\cdots}.$$

For any list S of at most $r-1$ swap operations, any stable matching in the instance $\mathcal{I}[S]$ contains the edges $\{m_i^d, w_i^d\}$ for all $i \in [r]$.

Proof. Let S be any list of at most $r-1$ swap operations. For the sake of contradiction, assume that there exists an $i_1 \in [r]$ and a stable matching $M \in \mathcal{M}_{\mathcal{I}[S]}$ such that $\{m_{i_1}^d, w_{i_1}^d\} \notin M$. This implies that there either exist s indices $i_1, \dots i_s$ such that $\{m_{i_1}^d, w_{i_s}^d\}$ and $\{m_{i_{j+1}}^d, w_{i_j}^d\} \in M$ for $j \in [s-1]$ or there exist some $w \notin \{w_1^d, \dots, w_r^d\}$ and $m \notin \{m_1^d, \dots, m_r^d\}$ together with s indices $i_1, \dots i_s$ such that $\{m_{i_1}^d, w\} \in M$ and $\{m, w_{i_s}^d\} \in M$ and $\{m_{i_{j+1}}^d, w_{i_j}^d\} \in M$ for $j \in [s-1]$.

In the first case, for every $j \in [s]$, at least one of $w_{i_j}^d$ and $m_{i_j}^d$ needs to prefer his or her partner in M to $m_{i_j}^d$ and $w_{i_j}^d$, since $\{m_{i_j}, w_{i_j}\}$ does not block M. Thus, we assume without loss of generality that $w_{i_1}^d$ prefers $m_{i_2}^d$ to $m_{i_1}^d$. Furthermore, there exists no i_j such that both $w_{i_j}^d$ and $m_{i_{j+1}}^d$ prefer their partners in M to $m_{i_j}^d$ and $w_{i_{j+1}}^d$, respectively, as this would already require r swaps. It follows that $w_{i_j}^d$ prefers $m_{i_{j+1}}^d$ to $m_{i_j}^d$ for every $j \in [s]$ (where $i_{s+1} := i_s$). Define $\text{dist}(p, p') := p' - p \bmod r$. To make $w_{i_j}^d$ prefer $m_{i_{j+1}}^d$ to $m_{i_j}^d$, one needs to perform at least $\text{dist}(i_j, i_{j+1})$ swaps in the preference list of $w_{i_j}^d$ (where $i_{s+1} = i_1$). Summing over the number of swaps for $w_{i_1}, \dots, w_{i_s}$, we get that at least r swaps have been performed, a contradiction.

In the second case, we may assume without loss of generality by the same argument as in the first case that all women $w_{i_j}^d$ prefer $m_{i_{j+1}}^d$ to $m_{i_j}^d$, but $m_{i_{j+1}}^d$ does not prefer $w_{i_j}^d$ to $w_{i_{j+1}}^d$ for $j \in [s-1]$. However, as otherwise $\{w_{i_s}^d, m_{i_s}^d\}$ forms a blocking pair, this implies that $w_{i_s}^d$ prefers m to $m_{i_s}^d$, which needs at least r swaps, a contradiction. $\square$

We now use Lemma 7 to model *SwapRestricted* by *Swap* by adding sufficiently many agents with preferences as in Lemma 7 such that any swap not allowed in the *SwapRestricted* instance drastically exceeds the budget in the *Swap* instance:

Lemma 8. *Let $(\mathcal{I} = (U, W, \mathcal{P}), \hat{A})$ be a SwapRestricted instance with n men and n women, and $4 \le c \in \mathbb{N}$. One can create a* Stable Marriage *instance $\mathcal{I}'$ by adding $2n^c(n-1)$ dummy agents $m_1^d, \dots, m_{n^c(n-1)}^d$, $w_1^d, \dots, w_{n^c(n-1)}^d$ such that*

(a) all swaps which are allowed in $\mathcal{I}$ are also possible in $\mathcal{I}'$ (i.e., for every agent a which is allowed to swap agents b and b' in $\mathcal{I}$, there is no agent between b and b' in the preferences of agent a in $\mathcal{I}'$),

(b) for any list S of allowed swap operations in $\mathcal{I}$, it holds that $\mathcal{M}_{\mathcal{I}'[S]} = \{M \cup \{\{m_i^d, w_i^d\} : i \in [n^c(n-1)]\} : M \in \mathcal{M}_{\mathcal{I}[S]}\}$, and

(c) for any list S' of at most n^c swap operations in the instance $\mathcal{I}'$, it holds that $\mathcal{M}_{\mathcal{I}[S]} = \{M|_{U \cup W} : M \in \mathcal{I}[S']\}$, where S is the sublist of S' containing the swaps in S' between

the two most preferred agents of some agent $a \in \hat{A}$ in $\mathcal{I}$. Furthermore, any $M \in \mathcal{I}[S']$ contains the edge $\{m_i^d, w_i^d\}$ for $i \in [n^c(n-1)]$.

Proof. Let $(\mathcal{I} = (U, W, \mathcal{P}), \hat{A})$ be an instance of *SwapRestricted*. We modify the given SM instance $\mathcal{I}$ to obtain a new SM instance $\mathcal{I}'$ by adding $n^c(n-1)$ additional men $m_1^d, \dots, m_{n^c(n-1)}^d$ and $n^c(n-1)$ additional women $w_1^d, \dots, w_{n^c(n-1)}^d$. For the rest of the proof, all indices are taken modulo $n^c(n-1)$. The preferences of these men and women are as follows:

$$m_i^d : w_i^d \succ w_{i+1}^d \succ w_{i+2}^d \succ \dots \succ w_{n^c(n-1)+i-1}^d \succ \overset{\text{(rest)}}{\cdots},$$
$$w_i^d : m_i^d \succ m_{i+1}^d \succ m_{i+2}^d \succ \dots \succ m_{n^c(n-1)+i-1}^d \succ \overset{\text{(rest)}}{\cdots}.$$

For a man $m \in U \setminus \hat{A}$ with $m : w_1 \succ w_2 \succ \dots \succ w_n$ and a woman $w \in W \setminus \hat{A}$ with $w : m_1 \succ m_2 \succ \dots \succ m_n$, their modified preferences look as follows:

$$m : w_1 \succ w_1^d \succ w_2^d \succ \dots \succ w_{n^c}^d \succ w_2 \succ w_{n^c+1}^d \succ w_{n^c+2}^d \succ \dots \succ w_{2n^c}^d \succ w_3 \succ \dots \succ w_n,$$

$$w : m_1 \succ m_1^d \succ m_2^d \succ \dots \succ m_{n^c}^d \succ m_2 \succ m_{n^c+1}^d \succ m_{n^c+2}^d \succ \dots \succ m_{2n^c}^d \succ m_3 \succ \dots \succ m_n.$$

For a man $m \in U \cap \hat{A}$ with $m : w_1 \succ w_2 \succ \dots \succ w_n$ (and analogously for a woman $w \in W \cap \hat{A}$), his modified preferences look as follows:

$$\begin{aligned} m : w_1 \succ w_2 \succ w_{n^c+1}^d \succ w_{n^c+2}^d \succ \dots \succ w_{2n^c}^d \succ w_3 \succ w_{2n^c+1} \succ \dots \succ w_{3n^c} \\ \succ w_4 \succ \dots \succ w_n \succ w_1^d \succ w_2^d \succ \dots \succ w_{n^c}^d. \end{aligned}$$

We now prove that the constructed SM instance $\mathcal{I}'$ indeed fulfills the three conditions from the lemma. Part (a) is obvious. Part (b) is also clearly fulfilled, as m_i^d and w_i^d are their mutual top choices in $\mathcal{I}'[S]$ for $i \in [n^c(n-1)]$ and therefore are matched together by any stable matching in $\mathcal{I}'[S]$ and can never form a blocking pair with another agent (dummy agents can never be part of a swap from S, as they are not part of the instance $\mathcal{I}$).

To prove part (c), let S' be any set of at most n^c swap operations in $\mathcal{I}'$. By Lemma 7, any stable matching in $\mathcal{I}'[S']$ contains the edges $\{m_i^d, w_i^d\}$ for $i \in [r]$. As any change which does not involve at least one agent m_i^d or w_i^d swaps the two most preferred agents of some agent $a \in \hat{A}$, the lemma follows. $\square$

Combining the hardness result from Lemma 6 and the construction from Lemma 8 modeling *SwapRestricted* by *Swap*, we now prove the following inapproximability result. Note that the inapproximability is tight in the sense that there is always a solution using at most $2(n-1)$ swaps which, given the edge $\{m^*, w^*\}$ that shall be contained in a stable matching, just swaps m^* to be the most preferred man of w^* and swaps w^* to be the most preferred woman of m^*.

Theorem 3. *Unless FPT = W[1],* CONSTRUCTIVE-EXISTS-SWAP *does not admit an $\mathcal{O}(n^{1-\epsilon})$-approximation in $f(\ell)n^{\mathcal{O}(1)}$ time for any $\epsilon > 0$ and any computable function f. This also holds if one is only allowed to swap the first two men in the preferences of women.*

Proof. Fix $\epsilon > 0$. We assume without loss of generality that $\epsilon = 4c^{-1}$ for some $c \in \mathbb{N}$ (if this is not the case, then we can choose a smaller ϵ fulfilling this condition).

We reduce from Constructive-Exists-SwapRestricted, for which it is W[1]-hard to distinguish whether there is a solution with at most ℓ manipulations or no solution for any number of allowed swaps (Lemma 6). Let $(\mathcal{I}, \hat{A}, \ell)$ be an instance of Constructive-Exists-SwapRestricted with at most n_R men and at most n_R women. By Lemma 8, we can construct an instance of Constructive-Exists-Swap which contains a solution of size at most ℓ if and only if the corresponding instance of Constructive-Exists-SwapRestricted contains a solution of size at most $\ell \leq n_R^3$, and otherwise any solution has size at least n_R^c. Let $n = n_R^c(n_R^c - 1) + n_R^c = \Theta(n_R^{c+1})$ be the number of men in the Constructive-Exists-Swap instance. Thus, given an instance of Constructive-Exists-SwapRestricted, an $\mathcal{O}(n^{1-\epsilon})$-approximation for Constructive-Exists-Swap allows to decide whether there exists a solution of size $\mathcal{O}(\ell n^{1-\epsilon}) \leq \mathcal{O}(n_R^{(c+1)(1-\epsilon)+3}) = \mathcal{O}(n_R^{(\frac{4}{\epsilon}+1)(1-\epsilon)+3}) = \mathcal{O}(n_R^{\frac{4}{\epsilon}-4+1-\epsilon+3}) = \mathcal{O}(n_R^{\frac{4}{\epsilon}-\epsilon}) = \mathcal{O}(n_R^{c-\epsilon}) < n_R^c$ in the given instance (the last inequality holds only for sufficiently large n_R). Therefore, assuming that we have an $\mathcal{O}(n^{1-\epsilon})$-approximation algorithm for Constructive-Exists-Swap running in $f(\ell)n^{\mathcal{O}(1)}$ time for some computable function f allows to decide Constructive-Exists-SwapRestricted in $f(\ell)n_R^{\mathcal{O}(1)}$ time, which implies by Lemma 6 that FPT = W[1]. □

3.2 An XP Algorithm for Constructive-Exists-Reorder

Observe that for all manipulative actions except *Reorder*, the membership of Constructive-Exists-$\mathcal{X}$ parameterized by ℓ in XP follows from a straightforward brute-force algorithm. We now show that Constructive-Exists-Reorder also lies in XP parameterized by the budget ℓ using a simple algorithm.

Proposition 1. Constructive-Exists-Reorder *can be solved in* $\mathcal{O}(2^\ell n^{2\ell+2})$ *time.*

Proof. We guess the set X of ℓ agents whose preferences are modified, and for each agent $a \in X$, we guess an agent $T(a)$ to which a is matched in a stable matching containing $\{m^*, w^*\}$. For each $a \in X$, we modify the preferences of a such that $T(a)$ is its top-choice. We check whether the resulting instance contains a stable matching containing the edge $\{m^*, w^*\}$. If any guess results in a stable matching containing $\{m^*, w^*\}$, then we return YES, and NO otherwise.

The running time of the algorithm is $\mathcal{O}(2^\ell n^{2\ell+2})$, as we guess a set of ℓ out of $2n$ agents whose preference are modified, and for each of these ℓ agents we guess to which agent it is matched to. Checking whether there is a stable matching containing $\{m^*, w^*\}$ can be done in $\mathcal{O}(n^2)$ time for each guess (Gusfield, 1987).

It remains to show the correctness. If the algorithm returns YES, then the instance is clearly a YES-instance. To prove the other direction, assume that there exists a set Y of at most ℓ agents such that in an instance $\mathcal{I}_Y$ arising through reordering the preferences of Y there exists a stable matching M containing the edge $\{m^*, w^*\}$. Then, there exists a guess (X, T) for our algorithm such that $Y \subseteq X$ and $T(a) = M(a)$ for all $a \in X$. We claim that for the instance constructed by the algorithm using this guess, the matching M is stable, and therefore the algorithm returns YES. Note that no agent from X can be contained in a blocking pair, as they are all matched to their top-choice. However, any blocking pair containing no agent from X would also be a blocking pair for M in $\mathcal{I}_Y$, and therefore, no blocking pair exists. □

3.3 Polynomial-Time Algorithms

Having encountered computational hardness for all manipulative actions but *Delete*, we now give a polynomial-time algorithm for Constructive-Exists-Delete. We then use this algorithm to derive a 2-approximation algorithm for Constructive-Exists-Reorder. The polynomial-time algorithm for *Delete* might be particularly surprising because of the strong hardness result that we derived for the related manipulative action *Add*. However, the reason for this is that, in a Constructive-Exists-$\mathcal{X}$ instance, we can compute a set of "conflicting agents" that prevent that the pair $\{m^*, w^*\}$ is part of a stable matching, and one *Delete* operation can only decrease the number of conflicting agents by at most one (and, in fact, by deleting a conflicting agent, we can also always decrease the number of conflicting agents by one). For *Add*, however, there are presumably multiple ways or no way how one could resolve a conflicting agent. Similarly, we derived an inapproximability result for *Swap*, while the seemingly similar manipulative action Reorder admits a factor-2 approximation in the *Constructive-Exists* setting. The reason for this is that the *Reorder* operation is more powerful, allowing to create in some sense trivial solutions again by identifying a set of "conflicting agents". These trivial solutions then consist of reordering the preferences of all conflicting agents and we prove that this is a 2-approximation. Thus, compared to the *Swap* setting where one can save quite some manipulative actions by choosing the swaps to perform, here it does not make a fundamental difference how the agent's preferences are reordered. We start by presenting the polynomial-time algorithm for *Delete*, and then describe the factor-2 approximation algorithm for *Reorder*.

3.3.1 *Delete*

In sharp contrast to the hardness results for all other considered manipulative actions, there is a simple algorithm solving a given instance of Constructive-Exists-Delete consisting of an SM instance $\mathcal{I} = (U, W, \mathcal{P})$ together with a man-woman pair $\{m^*, w^*\}$ and an integer ℓ in time linear in the size of the input. The algorithm is based on the following observation. Let W^* be the set of women preferred by m^* to w^*, and U^* the set of men preferred by w^* to m^*. In every stable matching M which includes $\{m^*, w^*\}$, every woman in W^* needs to be matched to a man whom she prefers to m^*, or she needs to be deleted. Analogously, every man in U^* needs to be matched to a woman which he prefers to w^*, or he needs to be deleted. Consequently, all pairs consisting of an agent $a \in U^* \cup W^*$ and an agent a' which a does not prefer to w^* or m^* cannot be part of any stable matching. After deleting these pairs and the agents m^* and w^*, consider a stable matching M in the resulting instance. If we add the pair $\{m^*, w^*\}$ to M, then in the original instance $\mathcal{I}$ each woman $w \in W^*$ not assigned by M will always create a blocking pair $\{m^*, w\}$ (independent of how and whether we extend M). Similarly, each unassigned man $m \in U^*$ will always create a blocking pair $\{m, w^*\}$. Thus, the agents from $U^* \cup W^*$ that are unassigned in M will be the set of "conflicting agents". This observation motivates the following algorithm:

1. Let U^* be the set of agents which w^* prefers to m^*.
   Let W^* be the set of agents which m^* prefers to w^*.

2. Delete all edges $\{m, w\}$ such that $m \in U^*$ and m does not prefer w to w^*.
   Delete all edges $\{m, w\}$ such that $w \in W^*$ and w does not prefer m to m^*.
   Delete the agents m^* and w^*.
   Call the resulting SMI instance $\hat{\mathcal{I}}$.

3. Compute a stable matching $\hat{M}$ in $\hat{\mathcal{I}}$.

4. Return the set A' of agents from $U^* \cup W^*$ which are unassigned in $\hat{M}$ as the agents to be deleted.

To show the correctness of this algorithm, we first investigate the influence of a *Delete* operation on the set of agents matched in a stable matching.

Lemma 9. *Let $\mathcal{I}'$ be an* SMI *instance and $a \in A$ some agent. Then, there exists at most one agent $a' \in A$ which was unassigned in $\mathcal{I}'$, i.e., $a' \notin \mathrm{ma}(\mathcal{I}')$, and is matched in $\mathcal{I}' \setminus \{a\}$, i.e., $a' \in \mathrm{ma}(\mathcal{I}' \setminus \{a\})$.*

Proof. Let M' be a stable matching in $\mathcal{I}'$ and M^a be a stable matching in $\mathcal{I}' \setminus \{a\}$. We prove that $|\,\mathrm{ma}(M^a) \setminus \mathrm{ma}(M')| \leq 1$, which clearly implies the lemma. To do so, we examine the symmetric difference S of the two matchings M' and M^a. The set S is a union of paths and even-length cycles. We do not care about cycles, as all agents in such a cycle are matched in both M' and M^a. Therefore, we turn to paths.

First, of all, we claim that all paths include a. Assume that there exists a path $P = (a_1, \ldots, a_{k+1})$ for some $k \in \mathbb{N}$ not including a, starting without loss of generality with a pair from M'. Then, as $\{a_1, a_2\}$ is not a blocking pair in M^a and a_1 is unassigned in M', it holds that $a_3 \succ_{a_2} a_1$. As $\{a_2, a_3\}$ is not a blocking pair in M', it holds that $a_4 \succ_{a_3} a_2$. Consequently, it holds that $a_{k+1} \succ_{a_k} a_{k-1}$. If k is even, then a_{k+1} is unassigned in M', and thus $\{a_k, a_{k+1}\}$ is a blocking pair in M', contradicting the stability of M'. Otherwise, a_{k+1} is unassigned in M^a, and $\{a_k, a_{k+1}\}$ is blocking in M^a.

From this it follows that there exists at most one maximal path P, which needs to involve agent a. As agent a cannot be part of M^a, it needs to be one of the endpoints of P. Consequently, the only agent that could be matched in M^a but not matched in M' is the other endpoint of P. □

Using Lemma 9, we now show the correctness of the algorithm.

Theorem 4. Constructive-Exists-Delete *is solvable in $\mathcal{O}(n^2)$ time.*

Proof. Since a stable matching in an SMI instance can be computed in $\mathcal{O}(n^2)$ time (Gale & Shapley, 1962), the set A' clearly can be computed in $\mathcal{O}(n^2)$. We claim that the given Constructive-Exists-Delete instance is a YES-instance if and only if $|A'| \leq \ell$.

First assume $|A'| \leq \ell$. Let $\hat{M}$ be the stable matching in $\hat{\mathcal{I}}$ computed by the algorithm. We define $M := \hat{M} \cup \{\{m^*, w^*\}\}$, and claim that M is a stable matching in $\mathcal{I} \setminus A'$, showing that the given Constructive-Exists-Delete instance is a YES-instance. For the sake of a contradiction, assume that there exists a blocking pair $\{m, w\}$. Note that $\{m, w\}$ contains neither m^* nor w^* since all agents which m^* and w^* prefer to each other are either deleted or matched to an agent they prefer to m^* and w^*. Since $\{m, w\}$ is not a blocking pair in $\hat{\mathcal{I}} \setminus A'$, it contains an agent a from $(U^* \cup W^*) \setminus A'$, and a prefers w^* to w if $a = m$ or a prefers m^* to m if $a = w$. We assume without loss of generality that $a = m$. As m is matched in $\hat{M}$, he prefers $\hat{M}(m) = M(m)$ to w^*. Thus, m prefers $M(m)$ to w, a contradiction to $\{m, w\}$ being blocking for M.

Now assume that $|A'| > \ell$. For the sake of contradiction, assume that there exists a set of agents $B' = \{b_1, \ldots, b_k\}$ with $k \leq \ell$ such that $\mathcal{I} \setminus B'$ admits a stable matching M containing $\{m^*, w^*\}$. For each $i \in \{0, 1, \ldots, k\}$, let $\hat{M}_i$ be a stable matching in $\hat{\mathcal{I}} \setminus \{b_1, \ldots, b_i\}$. By the definition of A', all agents from A' are unassigned in $\hat{M}_0$. Note that each agent $a \in A' \subseteq U^* \cup W^*$ is either part of B' or prefers $M(a)$ to m^* or w^* due to the stability of M; in particular, a is either contained in B' or matched in M. Since $k \leq \ell < |A'|$, there exists an i such that there exist two agents $a, a' \in A'$ which are unassigned in $\hat{M}_{i-1}$ and not contained in $\{b_1, \ldots, b_{i-1}\}$ but matched in $\hat{M}_i$ or contained in $\{b_1, \ldots b_i\}$. It follows from Lemma 9 that it is not possible that both a and a' are unassigned in $\hat{M}_{i-1}$ but matched in $\hat{M}_i$. Consequently, without loss of generality it needs to hold that $a = b_i$ with a being unassigned in $\hat{M}_{i-1}$, and $a' \in \mathrm{ma}(\hat{\mathcal{I}} \setminus \{b_1, \ldots b_i\}) \setminus \mathrm{ma}(\hat{\mathcal{I}} \setminus \{b_1, \ldots b_{i-1}\})$. However, deleting an agent that was previously unassigned does not change the set of matched agents, i.e., $\mathrm{ma}(\hat{\mathcal{I}} \setminus \{b_1, \ldots b_{i-1}\}) = \mathrm{ma}(\hat{\mathcal{I}} \setminus \{b_1, \ldots b_i\})$, since a matching that is stable in $\hat{\mathcal{I}} \setminus \{b_1, \ldots b_{i-1}\}$ is also stable in $\mathrm{ma}(\hat{\mathcal{I}} \setminus \{b_1, \ldots b_i\})$ (deleting unassigned agents cannot create new blocking pairs). This contradicts $a' \in \mathrm{ma}(\hat{\mathcal{I}} \setminus \{b_1, \ldots b_i\}) \setminus \mathrm{ma}(\hat{\mathcal{I}} \setminus \{b_1, \ldots b_{i-1}\})$. $\square$

The algorithm of Theorem 4 directly extends to the setting where an arbitrary number of edges is given and the goal is to delete agents such that there exists a stable matching which is a superset of the given set of edges.

3.3.2 Factor-2 Approximation for *Reorder*

We now follow a similar approach as for Constructive-Exists-Delete to construct a factor-2 approximation for the optimization version of Constructive-Exists-Reorder (notably, this algorithms crucially relies on our assumption that $|U| = |W|$). We construct an instance $\hat{\mathcal{I}}$ identically as in the case of Constructive-Exists-Delete: Let W^* be the set of women preferred by m^* to w^*, and U^* be the set of men preferred by w^* to m^*. In every stable matching M which contains $\{m^*, w^*\}$, every woman which m^* prefers to w^* needs to be matched to a man which she prefers to m^*, or her preference list needs to be reordered. Analogously, every man which w^* prefers to m^* needs to be matched to a woman which he prefers to w^* or his preferences need to be reordered. Consequently, all pairs consisting of an agent $a \in U^* \cup W^*$ and an agent a' which a does not prefer to w^* or m^* cannot be part of any stable matching. This observation motivates a transformation of the given SM instance $\mathcal{I}$ to a SMI instance $\hat{\mathcal{I}}$ through the deletion of all such pairs. We also delete w^* and m^* from $\hat{\mathcal{I}}$ and compute a stable matching M in the resulting instance.

We observe analogously to Lemma 9 that by reordering the preferences of an agent, at most two previously unassigned agents become matched in an SMI instance:

Lemma 10. *Let $\mathcal{I}'$ be an* SMI *instance, $a^* \in A$ some agent, and let $\mathcal{I}^{a^*}$ denote the instance arising from $\mathcal{I}'$ by reordering and extending a^*'s preferences arbitrarily. Then, there exists* *at most one man $m \in U$ and at most one woman $w \in W$ who are unassigned in $\mathcal{I}'$, i.e., $m, w \notin \mathrm{ma}(\mathcal{I}')$, and who are matched in $\mathcal{I}^{a^*}$, i.e., $m, w \in \mathrm{ma}(\mathcal{I}^{a^*})$.*

Proof. The proof proceeds analogously to the proof of Lemma 9. Let M' be some stable matching in $\mathcal{I}'$ and M^{a^*} a stable matching in $\mathcal{I}^{a^*}$. We examine again the symmetric difference of M' and M^{a^*} and conclude that only the unique maximal path including a^* can change the set of matched agents. As this path has only two endpoints, at most two agents which are not matched in M' can become matched in M^{a^*}. Assuming that the path has an even number of edges, only one of the endpoints can be matched in M^{a^*}. Assuming that the path has an odd number of edges, one of the endpoints needs to correspond to a woman and one to a man. $\square$

Now, similarly to Theorem 4, it is possible to construct a straightforward solution, which matches one agent from $U^* \cup W^*$ which is currently not matched in any stable matching in $\hat{\mathcal{I}}$ by reordering the preferences of one agent. Using Lemma 10, we now show that this approach yields a factor-2 approximation of the optimal solution:

Proposition 2. *One can compute a factor-2 approximation of the optimization version of* Constructive-Exists-Reorder *in $\mathcal{O}(n^2)$ time.*

Proof. Given an instance of Constructive-Exists-Reorder consisting of an SM instance $\mathcal{I} = (U, W, \mathcal{P})$, budget ℓ, and the pair $\{m^*, w^*\}$, we construct an SMI instance $\hat{\mathcal{I}}$ and define U^* and W^* as described above. The 2-approximation algorithm proceeds as follows (recall that $\mathrm{ma}(M)$ is the set of agents assigned by matching M):

- Compute a stable matching $\hat{M}$ in $\hat{\mathcal{I}}$.
- Let $A^* := (U^* \cup W^*) \setminus \mathrm{ma}(\hat{M})$ and $A' := A \setminus \mathrm{ma}(\hat{M})$. Reorder the preferences of each agent from A^* such that all agents from A' of opposite gender are in the beginning of the preferences.

Let N be a stable matching on the instance restricted to agents from A' (with modified preferences). Note that the number of men which $\hat{M}$ leaves unassigned equals the number of women which $\hat{M}$ leaves unassigned (here we use that $|U| = |W|$), and thus, N matches every agent from A'. The correctness of the solution returned by the algorithm and the approximation factor can be proven in a way similar to the proof of Theorem 4:

First, we show that $M' := \hat{M} \cup N \cup \{\{m^*, w^*\}\}$ is a stable matching after the described reorderings of the preferences. No agent from A^* is part of a blocking pair, as N is stable and every agent from A^* prefers every agent from A' of opposite sex to every agent from $A \setminus A'$. No agent from $A' \setminus A^*$ is part of a blocking pair, as N is stable and no agent from $A \setminus A'$ prefers an agent from A' to its partner in $\hat{M}$. By the same arguments as in Theorem 4, no agent from $A \setminus A'$ is part of a blocking pair.

It remains to show that at least $q := \frac{|A^*|}{2}$ reorderings are needed. Assume that less than q reorderings are needed, and let $B = \{b_1, \ldots, b_k\}$ with $k < q$ be the set of agents whose preferences have been reordered; we refer to the SM instance arising through these reorderings as $\mathcal{I}^*$. Let M^* be a stable matching containing $\{m^*, w^*\}$ in the instance $\mathcal{I}^*$. Let $\hat{M}_i$ be a stable matching in the instance $\hat{\mathcal{I}}_i$ arising from $\hat{\mathcal{I}}$ by replacing the preference list of $b_1, \ldots, b_i$ by their reordered, complete preferences (this includes adding for each agent $a \in A \setminus \{b_1, \ldots, b_i\}$ such that $\{a, b_j\}$ is not contained in $\hat{\mathcal{I}}$ the agent b_j in a's preferences at its position in a's preferences in $\mathcal{I}$). Note that M^* is a stable matching in $\hat{\mathcal{I}}_k$ (if M^* contained an edge $\{m, w\}$ not contained in $\hat{\mathcal{I}}_k$, then $m \in U^*$ and m would prefer w^* to $M^*(m)$, implying that $\{m, w^*\}$ is a blocking pair, or $w \in W^*$ and w would prefer m^* to $M^*(w)$, implying that $\{m^*, w\}$ is a blocking pair). Then, every agent from A is matched in $\hat{M}_k$. By the definition of A^*, all agents from A^* are unassigned in $\hat{M}_0$. As it holds that $k < \frac{|A^*|}{2}$ and all agents from A^* are unassigned in a stable matching $\hat{\mathcal{I}}_0$, there needs to exist some $j \in \{0, 1, \ldots, k-1\}$ such that three agents from A^* that are unassigned in a stable matching in $\hat{\mathcal{I}}_j$ are matched in a stable matching in $\hat{\mathcal{I}}_{j+1}$. This contradicts Lemma 10. It follows that the assumption $|B| < \frac{|A^*|}{2}$ is wrong, and thus, the algorithm computes a 2-approximation. □

Note that the above approach does not directly carry over to the case $|U| \neq |W|$. The problem is that the matching M' constructed in the proof of Proposition 2 is not necessarily stable in this case. The reason for this is that $|U \cap A^*| > |W \cap A'|$ (or $|W \cap A^*| > |U \cap A'|$) might hold. In this case, there remains at least one unassigned man from $U \cap A^*$ (or at least one unassigned woman from $W \cap A^*$) in M' which then forms a blocking pair with w^* (or m^*) for M'. In fact, an optimal solution might be much larger than $|A^*|$, showing that better lower bounds are needed to design a constant-factor approximation for the general case. For example, consider an instance of Constructive-Exists-Reorder with $|U| = |W| + 1$, where w^* prefers all but one man m' to m^*, and every other woman has m' as her top-choice. Furthermore, every man but m^* has w^* as his last choice, while m^* has w^* as his top-choice. Then $\hat{\mathcal{I}}$ arises through the deletion of m^* and w^*, and every stable matching in $\hat{\mathcal{I}}$ leaves exactly one man $m \in U \setminus \{m^*, m'\}$ unassigned. Consequently, we have $A^* = \{m\}$. However, every stable matching has to match every man from $U \setminus \{m^*, m'\}$ to a woman from $W \setminus \{w^*\}$, implying that m' will be the only man unassigned in a stable matching. Therefore, any optimal solution has to reorder the preferences of every woman from $W \setminus \{w^*\}$.

Remark (Destructive-Exists). As already briefly discussed in the introduction, instead of considering *Constructive-Exists* it is also possible to consider *Destructive-Exists* where given a SM instance and a man-woman pair $\{m^*, w^*\}$, we want to alter the SM instance such that $\{m^*, w^*\}$ is not part of at least one stable matching. Notably, polynomial-time algorithms for *Constructive-Exists* carry over to *Destructive*-Exists, as we can solve the latter problem by running the algorithm for *Constructive-Exists* for all man-woman pairs involving one of m^* and w^*. Moreover, it is also possible to adapt our hardness reduction for Constructive-Exists-Add to Destructive-Exists-Add: We introduce an additional man m^{**} which has w^* as his top-choice and change the preferences of w^* to $w^*\colon m^* \succ m^{**} \succ \ldots$ and set $\{m^{**}, w^*\}$ to the pair that we want to exclude from some stable matching. Notably, a stable matching does not include $\{m^{**}, w^*\}$ if and only if it includes $\{m^*, w^*\}$. After the described modifications, the reduction no longer has any implications concerning inapproximability, as we need to use that our budget is $\ell = k + \binom{k}{2}$ in the proof of correctness of the backward direction of the reduction (Lemma 2), which needs to be slightly adapted. Thus, we can conclude that Destructive-Exists-Add parameterized by ℓ is W[1]-hard. By modeling *Add* by *Swap* as described in Section 3.1.3, we can also conclude that Destructive-Exists-Swap parameterized by ℓ is W[1]-hard.

4. *Exact-Exists*

In this section, we aim to make a given matching in an SM instance stable by performing some manipulative actions. The difference to the *Constructive-Exists* setting considered in the previous section is that now instead of one edge that should be included in some stable matching, a complete matching is given which shall be made stable. At first sight, it is unclear whether making the goal more specific in the sense of providing the complete matching instead of just one edge in some matching makes the problem easier or harder. In this section, we prove that providing this additional information makes the problem usually easier, as for all manipulative actions $\mathcal{X}$ for which we showed hardness in the previous section, Exact-Exists-$\mathcal{X}$ becomes polynomial-time solvable.[6] The intuitive reason for this difference is that the problem of making a given matching M^* stable simplifies to "resolving" all pairs that are blocking for M^*, which turns out to be solvable in polynomial-time. In contrast to this, for Constructive-Exists-$\mathcal{X}$, we also need to decide which matching including the given pair we want to make stable, a task which turned out to be hard for most manipulative actions.

Next, we start by considering the actions *DeleteAcceptability*, *Reorder*, and *Swap* before turning to the manipulative actions *Delete* and *Add* for which we need to adapt the problem definition.

4.1 Polynomial-Time Algorithms for *DeleteAcceptability*, *Reorder*, and *Swap*

As argued above, in the *Exact-Exists* setting, we need to "resolve" all pairs that are blocking for the given matching M^*. This requires manipulating the preferences of at least one agent a in each blocking pair such that it no longer prefers the other agent in the blocking pair to $M^*(a)$. For a matching M and an SM instance $\mathcal{I}$, we denote by $\mathrm{bp}(M, \mathcal{I})$ the set of all blocking pairs of M in $\mathcal{I}$. For a blocking pair $\beta = \{m, w\} \in \mathrm{bp}(M, \mathcal{I})$ and an agent $a \in \beta$, we denote by $\beta(a)$ the other agent in the blocking pair.

The optimal solution for an instance of Exact-Exists-DeleteAcceptability is to delete the acceptability of all blocking pairs. The set of blocking pairs can be computed in $\mathcal{O}(n^2)$ time. This solution is optimal since it is always necessary to delete the acceptability of all blocking pairs and, by doing so, no new pairs will become blocking.

Observation 1. Exact-Exists-DeleteAcceptability *is solvable in $\mathcal{O}(n^2)$ time.*

We now turn to *Reorder*. Since it is possible to resolve all blocking pairs containing an agent by manipulating this agent, Exact-Exists-Reorder corresponds to finding a minimum-cardinality subset $A' \subseteq U \cup W$ such that A' covers $\mathrm{bp}(M^*, \mathcal{I})$, i.e., each pair that blocks M^* in the given SM instance $\mathcal{I}$ contains an agent from A'. From this observation, the following proposition directly follows.

Proposition 3. Exact-Exists-Reorder *reduces to finding a vertex cover in a bipartite graph and is, hence, solvable in $\mathcal{O}(n^{2.5})$ time.*

6. The only manipulative action for which *Exact-Exists* is harder than *Constructive-Exists* is *Delete*. However, recall that we came up with a modified definition of *Exact-Exists* for *Delete*. As this definition makes Exact-Exists-Delete quite different from the problem for the other actions and also from Constructive-Exists-Delete, this observation does not contradict our previous claim.

Proof. Given an instance of Exact-Exists-Reorder consisting of an SM instance $\mathcal{I} = (U, W, \mathcal{P})$ together with a matching M^* and budget ℓ, we construct a bipartite graph as follows. For each $a \in A$, we introduce a vertex v_a and we connect two vertices $v_a, v_{a'}$ if $\{a, a'\} \in \text{bp}(M^*, \mathcal{I})$. Note that the resulting graph is bipartite, as there cannot exist a blocking pair consisting of two men or two women. We compute a minimum vertex cover X in the graph, i.e., a subset of vertices such that all edges are incident to at least one vertex in X, using the Hopcroft–Karp algorithm in $\mathcal{O}(n^{2.5})$ time (Hopcroft & Karp, 1973). For each $v_a \in X$, we reorder a's preferences such that $M^*(a)$ becomes a's top-choice. After these reorderings, M^* is stable, as for each blocking pair β, for at least one involved agent $a \in \beta$, the agent $M^*(a)$ is now a's top-choice, and no new blocking pairs are created by this procedure. Moreover, the computed solution is optimal. For the sake of contradiction, let us assume that there exists a smaller solution. Then, there exists an $\{m, w\} \in \text{bp}(M^*, \mathcal{I})$ where neither m's nor w's preferences have been modified. However, this implies that $\{m, w\}$ still blocks M^*. □

Now, we turn to the manipulative action *Swap*. Here, the cost of resolving a blocking pair $\{m, w\}$ by manipulating m's preferences is the number of swaps needed to swap $M^*(m)$ with w in the preferences of m, and the cost of resolving the pair by manipulating w's preferences is the number of swaps needed to swap $M^*(m)$ with w in the preferences of w. This observation could lead to the conjecture that it is optimal to determine for each blocking pair the agent with the lower cost and then resolve the pair by performing the corresponding swaps. However, this approach is not optimal, as by resolving some blocking pair involving an agent also another blocking pair involving this agent might be resolved, as we observe it in the following example:

Example 6. *Consider an instance of* Exact-Exists-Swap *consisting of the following SM instance together with budget $\ell = 3$ and $M^* = \{\{m_1, w_3\}, \{m_2, w_2\}, \{m_3, w_1\}\}$:*

- $m_1 : w_1 \succ w_2 \succ w_3$,
- $m_2 : w_2 \succ w_3 \succ w_1$,
- $m_3 : w_2 \succ w_3 \succ w_1$,
- $w_1 : m_1 \succ m_2 \succ m_3$,
- $w_2 : m_1 \succ m_3 \succ m_2$, *and*
- $w_3 : m_1 \succ m_2 \succ m_3$.

The set of blocking pairs of M^* is: $\text{bp}(M^*, \mathcal{I}) = \{\{m_1, w_1\}, \{m_1, w_2\}, \{m_3, w_2\}\}$. *In this example, the cost to resolve the pair $\{m_1, w_1\}$ by modifying m_1's preference list is two, i.e., swap w_3 and w_2 and subsequently w_3 and w_1. However, by doing this, also the other blocking pair $\{m_1, w_2\}$ including m_1 is resolved. In fact, these swaps are part of the unique optimal solution to make M^* stable, which is to swap w_3 and w_2 and subsequently w_3 and w_1 in m_1's preference relation and to swap m_3 and m_2 in w_2's preference relation.*

In the following, we describe how on instance of Exact-Exists-Swap can be solved in time cubic in the number of agents by reducing it to an instance of Minimum Cut. In the Minimum Cut problem, we are given a directed graph $G = (V, E)$, a cost function

$c: E \to \mathbb{N} \cup \{\infty\}$, two distinguished vertices s and t and an integer k, and the question is to decide whether there exists a subset of arcs of total weight at most k such that every (s,t)-path in G includes at least one of these arcs.

Before describing the reduction, let us introduce some notation. For two agents $a, a' \in A$, let $c(a, a')$ denote the number of swaps needed such that a prefers $M^*(a)$ to a', i.e., $c(a, a') := \max\big(\operatorname{rank}(a, M^*(a)) - \operatorname{rank}(a, a'), 0\big)$, where $\operatorname{rank}(a, a')$ is one plus the number of agents which a prefers to a'. Moreover, for each $a \in A$, let q_a denote the number of blocking pairs involving a and let $\beta_1^a, \dots, \beta_{q_a}^a$ be a list of these blocking pairs ordered decreasingly by the number of swaps in a's preferences needed to resolve the blocking pair, i.e., $c(a, \beta_1^a(a)) \geq c(a, \beta_2^a(a)) \geq \cdots \geq c(a, \beta_{q_a}^a(a))$. For a blocking pair $\beta \in U \times W$ with $a \in \beta$, we denote by $\operatorname{id}(a, \beta)$ the position of blocking pair β in a's list of blocking pairs, that is, $\operatorname{id}(a, \beta) = i$ if $\beta = \beta_i^a$. Using this notation, we now prove the following:

Theorem 5. Exact-Exists-Swap *is solvable in $\mathcal{O}(n^4)$ time.*

Proof. Assume we are given an instance of Exact-Exists-Swap consisting of an SM instance $\mathcal{I} = (U, W, \mathcal{P})$, a matching M^*, and budget ℓ. Let $A = \{a_1, \dots, a_{2n}\}$. We first show that there is always an optimal solution which, for each agent $a \in A$, only swaps $M^*(a)$ (upwards) in the preference of a (i.e., $M^*(a)$ becomes more preferred by a). Let S be a list of swap operations of minimum cardinality such that after performing the swap operations S, the given matching M^* is stable. Let $a \in A$ be some agent and $i \in \mathbb{N}$ the number of swap operations from S modifying the preferences of a. Then, it is possible to replace these i swap operations by swapping $M^*(a)$ by i positions to the top in a's preference list. The resulting list of swap operations S' consists of the same number of swaps and makes M^* still stable, as there is no agent which a prefers to $M^*(a)$ after the swap operations in S' but not after the swap operations in S. As a consequence, it is enough to consider the solutions to the given Exact-Exists-Swap instance that correspond to a tuple $(d_{a_1}, \dots, d_{a_{2n}})$, where d_a encodes the number of times $M^*(a)$ is swapped with its left neighbor in a's preference relation. Note that $(d_{a_1}, \dots, d_{a_{2n}})$ is a valid solution to the problem if for each blocking pair $\{m, w\} \in \operatorname{bp}(M^*, \mathcal{I})$ it holds that $d_m \geq c(m, w)$ or that $d_w \geq c(w, m)$. Now, we are ready to reduce the given Exact-Exists-Swap instance to an instance of the Minimum Cut problem.

Reduction to Minimum Cut. We start by constructing a weighted directed graph $G = (V, E)$ as follows: For each man m, we introduce one vertex for each blocking pair m is part of: $u_1^m, \dots, u_{q_m}^m$. Similarly, for each woman $w \in W$, we introduce one vertex for each blocking pair w is part of: $u_1^w, \dots, u_{q_w}^w$. Moreover, we add a source s and a sink t.

Turning to the arc set, for each $m \in U$ that is included in at least one blocking pair, we introduce an arc from s to u_1^m of cost $c(m, \beta_1^m(m))$. Moreover, for each $i \in [q_m - 1]$, we introduce an arc from u_i^m to u_{i+1}^m of cost $c(m, \beta_{i+1}^m(m))$. For each woman $w \in W$ that is included in at least one blocking pair, we introduce an arc from u_1^w to t of cost $c(w, \beta_1^w(w))$. Moreover, for each $i \in [q_w - 1]$, we introduce an arc from u_{i+1}^w to u_i^w of cost $c(w, \beta_{i+1}^w(w))$. For each blocking pair $\beta = \{m, w\} \in \operatorname{bp}(M^*, \mathcal{I})$, we introduce an arc from $u_{\operatorname{id}(m,\beta)}^m$ to $u_{\operatorname{id}(w,\beta)}^w$ of infinite cost. Finally, we set $k := \ell$. We visualize the described reduction in Figure 7 where the graph corresponding to Example 6 is displayed.

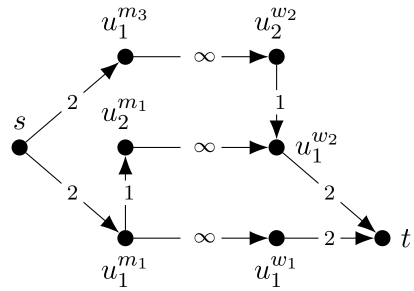


Figure 7: Min-Cut graph constructed to solve Example 6. The number on an edge denotes its weight. Note that $\beta_1^{m_1} = \{m_1, w_1\}$, $\beta_2^{m_1} = \{m_1, w_2\}$ with $c(m_1, \beta_1^{m_1}(m_1)) = 2$ and $c(m_1, \beta_2^{m_1}(m_1)) = 1$; $\beta_1^{m_3} = \{m_3, w_2\}$ with $c(m_3, \beta_1^{m_3}(m_3)) = 2$; $\beta_1^{w_1} = \{m_1, w_1\}$ with $c(w_1, \beta_1^{w_1}(w_1)) = 2$; $\beta_1^{w_2} = \{m_1, w_2\}$, $\beta_2^{m_2} = \{m_3, w_2\}$ with $c(w_2, \beta_1^{w_2}(w_2)) = 2$ and $c(w_2, \beta_2^{w_2}(w_2)) = 1$. The unique minimum (s,t)-cut is $E' := \{(s, u_1^{m_1}), (u_2^{w_2}, u_1^{w_2})\}$.

The general idea of the construction is that cutting an arc incident to some vertex u_i^a of an agent $a \in A$ of cost c is equivalent to swapping up $M^*(a)$ in a's preference list c times. Thereby, all blocking pairs with costs at most c for a are resolved (all paths visiting the corresponding vertices are cut) and we encode for each agent a the entry in the solution tuple by the arc incident to one of its vertices u_i^a contained in the cut (where no arc contained in the cut corresponds to performing no swaps). For each blocking pair the involved woman or the involved man needs to resolve the pair, as otherwise there still exists an (s,t)-path.

Formally, we compute a minimum cut $E' \subseteq E$ of the constructed graph, which can be done in $\mathcal{O}(|V| \cdot |E|) = \mathcal{O}(n^4)$ time (King, Rao, & Tarjan, 1994; Orlin, 2013). Note that for each agent a at most one arc to one of $u_1^a, \dots, u_{q_a}^a$ is contained in E'. For the sake of contradiction, let us assume that there exist two vertices u_i^a and u_j^a with $i < j$ such that both the arc to u_i^a and u_j^a have been cut. Then, already cutting the arc to u_i^a destroys all (s,t)-paths visiting u_j^a, contradicting the minimality of the cut.

Using E', we construct a solution tuple as follows. For each agent $a \in A$, we set $d_a = 0$ if no arc to a vertex from $u_1^a, \dots, u_{q_a}^a$ has been cut. Otherwise, let u_i^a be the destination of the arc in the cut. We set $d_a = c(a, \beta_i^a(a))$. Note that the cost of the cut corresponds to the cost of the constructed solution.

Correctness. It remains to prove that the solution $(d_{a_1}, \dots, d_{a_{2n}})$ computed by our algorithm is indeed a solution to the given Constructive-Exists-Swap instance and that no solution of smaller cost exists. To prove the first part, for the sake of contradiction, let us assume that there exists a pair $\beta = \{m, w\} \in \mathrm{bp}(M^*, \mathcal{I})$ that is still blocking, i.e., it holds that $d_m < c(m, w)$ and $d_w < c(w, m)$. However, this implies that the graph G' arising from G through the deletion of the edges from E' still contains a path from s to $u^m_{\mathrm{id}(m,\beta)}$ and from $u^w_{\mathrm{id}(w,\beta)}$ to t: No arcs on the unique path from s to $u^m_{\mathrm{id}(m,\beta)}$ were cut by E', as they were all of cost greater than $c(m, w)$. A symmetric argument shows that no edges on the path from $u^w_{\mathrm{id}(w,\beta)}$ to t are contained in E'. Moreover, there exists an arc of infinite cost from $u^m_{\mathrm{id}(m,\beta)}$ to $u^w_{\mathrm{id}(w,\beta)}$ which implies the existence of an (s,t)-path in G'. This leads to a contradiction.

To prove the second part, let us assume that there exists a solution $(d'_{a_1}, \ldots, d'_{a_{2n}})$ of smaller cost. However, one can construct from this a cut E'' of smaller cost than the computed minimum cut E', a contradiction. For each agent $a \in A$, we include in the cut E'' the arc to the vertex from $u_1^a, \ldots, u_{q_a}^a$ of maximum index i such that $c(a, \beta_i^a(a)) \leq d'_a$. Clearly, E'' has the same cost as $(d'_{a_1}, \ldots, d'_{a_{2n}})$ and is therefore cheaper than the computed minimum cut E', so it remains to show that E'' is indeed an (s,t)-cut, i.e., after deleting all arcs from E'', there is no (s,t)-path. For the sake of contradiction, let us assume that there exists an (s,t)-path after deleting the arcs from E''. Then there exist $m \in U$ and $w \in W$ with $i \in [q_m]$ and $j \in [q_w]$ such that this path includes the arc (u_i^m, u_j^w). However, as it needs to hold that $d'_m \geq c(m, \beta_i^m(m))$ or $d'_w \geq c(w, \beta_j^w(w))$ (as otherwise $\{m, w\}$ would block M^* after the bribery), either an arc from the unique path from s to u_i^m or an arc from the unique path from u_j^w to t is part of E''. This leads to a contradiction. □

Having seen that Exact-Exists-Swap/DeleteAcceptability/Reorder are solvable in polynomial time, it is natural to ask whether these tasks remain tractable if we instead of specifying a full matching only specify a set of edges that should be made part of a stable matching. Note that we have seen in Section 3.1 that Constructive-Exists-Swap/DeleteAcceptability/Reorder, where the goal is to make just one edge part of a stable matching, are NP-hard. Looking now at cases in between these two extremes, i.e., an arbitrary number j of edges that should be included in some stable matching is given, it is straightforward to come up with an FPT-algorithm with respect to the parameter $n - j$.

Proposition 4. *For a given SM instance $\mathcal{I}$ and partial matching $\widetilde{M} \subseteq U \times W$ of size j, one can decide in $(n-j)!n^{\mathcal{O}(1)}$ time whether it is possible to modify $\mathcal{I}$ using Swap/ DeleteAcceptability/ Reorder actions such that $\widetilde{M}$ is part of some stable matching.*

Proof. Let $\mathcal{X} \in \{\textit{Swap}, \textit{DeleteAcceptability}, \textit{Reorder}\}$. The idea is to brute-force over all possibilities $\widetilde{M}'$ of matching the remaining $2(n-j)$ agents not included in $\widetilde{M}$ to each other. There are $(n-j)!$ such possibilities (fix an ordering of men and iterate over all possible orderings of women and match two agents at the same position in the orderings to each other). For each possibility, we employ the algorithm for Exact-Exists-$\mathcal{X}$ to decide whether the complete matching $\widetilde{M} \cup \widetilde{M}'$ can be made stable using at most ℓ manipulative actions of type $\mathcal{X}$. □

4.2 *Delete* and *Add*

In this section, we turn to the manipulative actions *Delete* and *Add* for which we needed to adapt the definition of *Exact-Exists*. Recall that in the context of these manipulative actions we are given a complete matching M^* involving all agents in the instance and the goal is to modify the instance such that there exists a stable matching M' with $M' \subseteq M^*$. First, we show that for *Add* the problem can be solved in linear time. Second, we argue that, in contrast to all other manipulative actions, the *Exact-Exists* question is computationally hard for the manipulative action *Delete*. This is at first sight surprising since *Delete* is the only manipulative action for which the *Constructive-Exists* question is solvable in polynomial time. However, it can be easily explained by the fact that we need to use the adapted definition of the *Exact-Exists* problem here.

Data: An SM instance $\mathcal{I}$, a complete matching M^*, a budget ℓ, and two sets U_{add} and W_{add}.

1 Set $X_A := \{w \in W_{\text{add}} : \exists m \in U \setminus U_{\text{add}} \text{ with } M^*(u) = w\}$;
2 **while** *there exists a lonely woman* $w \in W_{\text{orig}} \cup X_A$ *and some man* $m \in U_{\text{orig}} \cup X_A$ *with* $w \succ_m M^*(m)$ **do**
3 | Add $M^*(w)$ to X_A;
4 **if** $M^*|_{U_{\text{orig}} \cup W_{\text{orig}} \cup X_A}$ *is stable and* $|X_A| \leq \ell$ **then**
5 | **return** X_A;
6 **else**
7 | **return** False;

Algorithm 1: Linear-time algorithm for Exact-Exists-Add.

We start by showing that Exact-Exists-Add is solvable in linear time in the input size. On an intuitive level, this is due to the fact that, for *Add*, it is already determined by the instance which agents we have to insert to allow for the existence of a stable matching $M' \subseteq M^*$, as in case some agent a blocks the matching M' in the instance consisting of agents $U_{\text{orig}} \cup W_{\text{orig}} \cup X_A$ for some $X_A \subseteq U_{\text{add}} \cup W_{\text{add}}$ the only possibility to resolve this is to add $M'(a)$ to X_A. Following this idea, we prove that Exact-Exists-Add is linear-time solvable.

Proposition 5. Exact-Exists-Add *can be solved in* $\mathcal{O}(n^2)$ *time.*

Proof. Assume we are given an instance of Exact-Exists-Add consisting of an SM instance $(U, W, \mathcal{P})$ together with two subsets $U_{\text{add}} \subseteq U$ and $W_{\text{add}} \subseteq W$, a matching M^*, and budget ℓ. For a set $X_A \subseteq U_{\text{add}} \cup W_{\text{add}}$, we call an agent $a \in U_{\text{orig}} \cup W_{\text{orig}} \cup X_A$ *lonely* if $M^*(a) \notin U_{\text{orig}} \cup W_{\text{orig}} \cup X_A$. Note that for a solution X_A of agents to be added, there cannot exist both a lonely man and a lonely woman, as they otherwise would form a blocking pair. We assume without loss of generality that the instance admits a solution X_A without a lonely man or no solution at all (we can do this by applying the following algorithm twice (once with the role of men and women switched) and then taking the smaller solution). We show that Algorithm 1 solves the problem in $\mathcal{O}(n^2)$ time. As there exists no lonely man, for each man $m \in U_{\text{orig}} \cup X_A$ also $M^*(m)$ needs to be contained in $W_{\text{orig}} \cup X_A$. Thus, every woman added to X_A in Line 1 needs to be contained in every solution. Moreover, there cannot exist a lonely woman w and a man $m \in U_{\text{orig}} \cup X_A$ which prefers w to $M^*(m)$, as otherwise w and m would form a blocking pair. This implies that all agents added to X_A in Line 3 are necessary to create a stable matching which is a subset of M^*. By adding more agents to X_A, it is never possible to resolve any blocking pairs for $M^*|_{U_{\text{orig}} \cup W_{\text{orig}} \cup X_A}$, as we have already ensured that no lonely woman is part of a blocking pair. Thereby, if the instance admits a solution, then X_A computed by Algorithm 1 is a solution of minimum size. This proves the correctness of the algorithm.

All parts of Algorithm 1 except for the **while**-loop can be clearly performed in $\mathcal{O}(n^2)$ time. To see that the **while**-loop can be executed in $\mathcal{O}(n^2)$ time overall, we compute the set of lonely women once before entering the **while**-loop. The **while**-loop can be executed at most n times since there are only n women. In each execution of the **while**-loop, we update the set of critical lonely women in $\mathcal{O}(n)$ time by checking for each woman $w' \in$

$W_{\text{orig}} \cup (X_A \cap W)$ whether the man $M^*(w)$ added in the last execution of the **while**-loop prefers w' to $M^*(M^*(w))$ and adding w' to the set of lonely women if this is the case. $\square$

In contrast to this, for *Delete*, our modified goal definition allows for more flexibility in the problem to encode computationally hard problems, as it is possible to decide which agents one wants to delete from the instance to resolve all initially present blocking pairs. The intuitive reason why this problem is NP-hard is the following: First, one can ensure that for each deleted agent a, agent $M^*(a)$ needs to be deleted as well. By ensuring this, selecting the $n - \ell$ agents that remain after the modifications (and thereby also implicitly the ℓ agents to be deleted) corresponds to finding an independent set of size $n - \ell$ in the "underlying graph", where each vertex corresponds to a pair in M^* and two vertices are connected if the two corresponding pairs cannot be part of the same stable matching. Note that in contrast to Proposition 3, this graph is no longer bipartite. As Independent Set is NP-complete (Karp, 1972), it follows that Exact-Exists-Delete (and, in fact, also Exact-Unique-Delete, where the problem is to make a given matching the unique stable matching) is also NP-complete. In the following, we present this hardness result in more detail before showing that Exact-Exists-Delete parameterized by the budget ℓ is fixed-parameter tractable.

Proposition 6. Exact-Exists/Unique-Delete *is NP-complete. This also holds if one is only allowed to delete pairs from the given matching M^*.*

Proof. For Exact-Exists-Delete, membership in NP is obvious, as it is possible to determine in polynomial time whether a matching is stable. Further, for Exact-Unique-Delete, membership in NP follows from the fact that it is possible to determine in polynomial time whether a stable matching is unique, e.g., by running the Gale-Shapley algorithm to compute a stable matching M and afterwards the Gale-Shapley algorithm with roles of women and men swapped to compute a stable matching M' and checking whether M and M' are identical; then and only then M is the unique stable matching.

We show the NP-hardness of Exact-Exists/Unique-Delete by a reduction from the NP-complete Independent Set problem (Karp, 1972). Given an undirected graph G and an integer k, Independent Set asks whether there are k pairwise non-adjacent vertices in G. Given an Independent Set instance $G = (V, E)$ and integer k, we denote by $u_v^1, \dots u_v^{d_v}$ the list of all neighbors of a vertex $v \in V$. The general idea of the reduction is to introduce for each vertex $v \in V$ a man-woman pair who are matched to each other in the given matching M^* and a penalizing gadget that ensures that if one of the two agents from this pair is deleted, then the other one needs to be deleted as well. We construct the preferences of the agents in such a way that for every edge $\{v, v'\} \in E$, the agents corresponding to v prefer the agent corresponding to v' of opposite gender to its partner in M^*. Thus, two pairs from M^* can be part of the same stable matching if and only if they are non-adjacent in the given graph. Hence, finding a solution to the manipulation problem of size ℓ corresponds to finding an independent set of size $|V| - \ell$.

Formally, the construction of the corresponding Exact-Exists-Delete instance works as follows. In the SM instance $\mathcal{I}$, we introduce for each $v \in V$ a gadget consisting of one vertex man m_v, one vertex woman w_v, and dummy men and women $\widetilde{m}_v^i$ and $\widetilde{w}_v^i$ for $i \in [2|V|]$.

For all $v \in V$, the vertex man m_v and the vertex woman w_v have the preferences

$$m_v \colon w_{u_v^1} \succ \cdots \succ w_{u_v^{d_v}} \succ w_v \succ \widetilde{w}_v^1 \succ \cdots \succ \widetilde{w}_v^{2|V|} \succ \overset{\text{(rest)}}{\cdots},$$
$$w_v \colon m_{u_v^1} \succ \cdots \succ m_{u_v^{d_v}} \succ m_v \succ \widetilde{m}_v^1 \succ \cdots \succ \widetilde{m}_v^{2|V|} \succ \overset{\text{(rest)}}{\cdots}$$

and the dummy men and women $\widetilde{m}_v^i$ and $\widetilde{w}_v^i$ for $i \in [2|V|]$ have the preferences

$$\widetilde{m}_v^i \colon w_v \succ \widetilde{w}_v^i \succ \overset{\text{(rest)}}{\cdots}, \qquad \widetilde{w}_v^i \colon m_v \succ \widetilde{m}_v^i \succ \overset{\text{(rest)}}{\cdots}.$$

We set $M^* := \{\{m_v, w_v\} : v \in V\} \cup \{\{\widetilde{m}_v^i, \widetilde{w}_v^i\} : v \in V, i \in [2|V|]\}$ and $\ell := 2(|V| - k)$. We now prove the correctness of our construction.

($\Rightarrow$) Let $V' \subseteq V$ be an independent set of size k in G. We claim that deleting the agent set $A = \{\{m_v, w_v\} : v \in V \setminus V'\}$, which is of size $2(|V| - k) = \ell$, is a solution to the constructed Exact-Exists-Delete instance, i.e., $M' = M^* \setminus \{\{m_v, w_v\} : v \in V \setminus V'\}$ is a stable matching in the resulting instance. For the sake contradiction, assume that there exists a blocking pair for M' in $\mathcal{I} \setminus A$. However, no vertex man or vertex woman can be part of such a blocking pair, as they are all matched to their top-choice among the remaining agents. Every dummy agent is matched to the best non-vertex agent and thus does not form a blocking pair for M'. Thus, M' is stable (in fact, M' is even the unique stable matching in $\mathcal{I} \setminus A$).

($\Leftarrow$) Assume that there exists a subset $A' \subseteq U \cup W$ of agents of size at most $\ell = 2(n-k)$ such that some matching $M' \subseteq M^*$ is stable in $\mathcal{I} \setminus A'$. First of all note that it is never possible to delete for some $v \in V$ all corresponding dummy men or to delete all dummy women, as the number of both dummy men and dummy women for each vertex exceeds the given budget. From this it follows that if A' contains m_v for some $v \in V$, then it also has to contain w_v. The reason for this is that otherwise w_v together with some non-deleted dummy agent $\widetilde{m}_v^j$ forms a blocking pair for M'. Similarly, if A' contains w_v for some $v \in V$, then it also contains m_v. We now claim that $V' := \{v \in V : m_v \notin A'\}$ forms an independent set of size at least k. First of all note that V' has size at most k, as there exist n vertices, $|A'| \leq 2(|V| - k) = \ell$, and $m_v \in A'$ implies $w_v \in A'$. For all $v, v' \in V'$ we have that m_v and $w_{v'}$ are still present in $\mathcal{I} \setminus A'$ by the definition of V'. Thus, $\{v, v'\} \notin E$ for all $v, v' \in V'$, as otherwise $\{m_v, w_{v'}\}$ forms a blocking pair for M' in $\mathcal{I} \setminus A'$. $\square$

In contrast to the W[1]-hardness results for the other manipulative actions for *Constructive-Exists*, Exact-Exists-Delete parameterized by ℓ is fixed-parameter tractable. The algorithm is based on a simple search tree. We pick a blocking pair and branch over which endpoint of the blocking pair gets deleted. After deleting the selected endpoint, we recompute the set of blocking pairs and decrease ℓ by 1 (see Algorithm 2).

Proposition 7. Exact-Exists-Delete *can be solved in* $\mathcal{O}(n^2 2^\ell)$ *time.*

Proof. We claim that Algorithm 2 solves Exact-Exists-Delete in the stated running time. The correctness follows directly from the fact that for each blocking pair one of its endpoints needs to be deleted. The running time follows from the fact that the set of blocking pairs can be determined in $O(n^2)$ time and the search tree has depth ℓ and branches into two children at each node. $\square$

Data: An SM instance $\mathcal{I}$, a complete matching M^*, and a budget ℓ.

Set $A' = \emptyset$ and let P be the set of blocking pairs for M^* in $\mathcal{I}$;
while *there exists a pair in* P **do**
    if $\ell \leq 0$ **then**
        return False;
    Pick a pair $\{w, m\} \in P$ and branch over its endpoints $v = m$ or $v = w$;
    Set $A' := A' \cup \{v\}$ and $\ell := \ell - 1$;
    Set P to be the set of blocking pairs for $M^* \setminus \{e : e \cap A' \neq \emptyset\}$ in $\mathcal{I} \setminus A'$;
return A';

Algorithm 2: FPT algorithm wrt. ℓ for Exact-Exists-Delete.

5. *Exact-Unique*

In this section, we turn from the task of making a given matching stable to the task of making the given matching the unique stable matching. We show that this change makes the considered computational problems significantly more demanding in the sense that the *Exact-Unique* problem is W[2]-hard with respect to ℓ for *Reorder* and *Add* and NP-complete for *Swap*. In contrast, the problem for *DeleteAcceptability* is solvable in polynomial time. Recall that we have already proven in Proposition 6 that *Exact-Unique* is NP-complete for *Delete*.

5.1 Hardness Results

Both the W[2]-hardness result for the manipulative action *Reorder* and the NP-completeness for *Swap* follow from the same parameterized reduction from the NP-complete and W[2]-complete Hitting Set problem parameterized by solution size (Downey & Fellows, 2013) with small modifications. In an instance of Hitting Set, we are given a universe Z, a family $\mathcal{F} = \{F_1, \ldots, F_p\}$ of subsets of Z, and an integer k, and the task is to decide whether there exists a hitting set of size at most k, i.e., a set $X \subseteq Z$ with $|X| \leq k$ and $X \cap F \neq \emptyset$ for all $F \in \mathcal{F}$. The general idea of the construction is as follows: For each set $F \in \mathcal{F}$, we add a *set gadget* consisting of two men and two women, and, for each element $z \in Z$, we add an *element gadget* consisting of a man-woman pair. We connect all set gadgets to the element gadgets corresponding to the elements in the set. The preferences are constructed in a way such that in each set gadget where none of the element gadgets connected to it is manipulated, the two women can switch their partners and the resulting matching is still stable. In contrast, when an element gadget connected to the set gadget is manipulated, then this switch creates a blocking pair and every stable matching contains the same edges in this gadget. Thereby, the given matching M^* is the unique stable matching in the altered instance if and only if the manipulated element-gadgets form a hitting set. Note that in the following reduction, rather unintuitively, we manipulate agents to rank their partner in M^* worse to make M^* the unique stable matching.

Theorem 6. Exact-Unique-Reorder *parameterized by ℓ is W[2]-hard, and this also holds if the given matching M^* is already stable in the original instance and one is only allowed to modify the preferences of agents of one gender.*

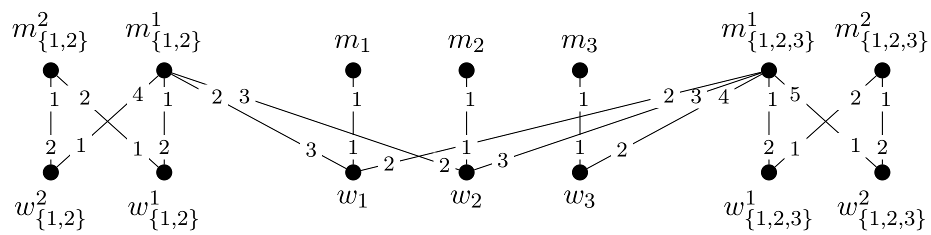


Figure 8: Example of the hardness reduction from Theorem 6 for the Hitting Set instance $Z = \{1, 2, 3\}$, $\mathcal{F} = \{\{1, 2, 3\}, \{1, 2\}\}$.

Proof. We give a parameterized reduction from Hitting Set, which is known to be W[2]-complete parameterized by the solution size k (Downey & Fellows, 2013). Given a Hitting Set instance $((Z, \mathcal{F} = \{F_1, \dots, F_q\}), k)$, for each element $z \in Z$, we add a man m_z and a woman w_z, which are the top-choices of each other (the preferences of both m_z and w_z are extended arbitrarily to include all agents of opposite sex). For each set $F = \{z_1, \dots, z_q\} \in \mathcal{F}$, we add two men m^1_F and m^2_F and two women w^1_F and w^2_F with the following preferences:

$$m^1_F : w^1_F \succ w_{z_1} \succ w_{z_2} \succ \cdots \succ w_{z_q} \succ w^2_F \succ \overset{\text{(rest)}}{\cdots}, \qquad m^2_F : w^2_F \succ w^1_F \succ \overset{\text{(rest)}}{\cdots},$$
$$w^1_F : m^2_F \succ m^1_F \succ \overset{\text{(rest)}}{\cdots}, \qquad w^2_F : m^1_F \succ m^2_F \succ \overset{\text{(rest)}}{\cdots}.$$

We set $M^* := \{\{m_z, w_z\} : z \in Z\} \cup \{\{m^1_F, w^1_F\}, \{m^2_F, w^2_F\} : F \in \mathcal{F}\}$ to be the man-optimal matching, and $\ell := k$ (see Figure 8 for a visualization).

We now show that the given Hitting Set instance admits a solution of size k if and only if it is possible to make M^* the unique stable matching in the constructed SM instance by reordering the preferences of at most ℓ agents.

($\Rightarrow$) Let $X \subseteq Z$ be a hitting set. For $z \in X$, we modify the preferences of w_z to be the following: $w_z : m^1_{F_1} \succ m^1_{F_2} \succ \cdots \succ m^1_{F_q} \succ m_z \succ \overset{\text{(rest)}}{\cdots}$. Matching M^* is still a stable matching, as each man is matched to his top-choice. To show that M^* is the unique stable matching, we utilize that if a second stable matching M' exists, then the union $M^* \cup M'$ needs to contain at least one cycle which consists of alternating edges from the two matchings. Moreover, as M^* is man-optimal (as every man is matched to his top-choice), each woman contained in the cycle needs to prefer the man matched to her in M' to the man matched to her in M^*. We now argue that such a cycle cannot exist and thereby that M^* is the unique stable matching. First of all, note that this cycle cannot contain an agent m_z (and thus neither w_z) for some $z \in Z$, as no woman prefers such a man to her partner in M^*. Thus, M' contains $\{m_z, w_z\}$ for every $z \in Z$. Next, we show that the cycle does not contain agent m^1_F for all $F \in \mathcal{F}$. Because X contains at least one $z_F \in F$, the preferences of at least one woman w_{z_F} with $z_F \in F$ have been reordered such that w_{z_F} now prefers m^1_F to m_{z_F}. As no woman w_z for $z \in Z$ can be part of a cycle in $M^* \cup M'$, a cycle in $M^* \cup M'$ containing m^1_F would imply that M' matches m^1_F to some woman from a set-gadget that is not his top-choice. However, then M' is blocked by the pair $\{m^1_F, w_z\}$, a contradiction. Hence, the cycle cannot contain an agent m^1_F. Therefore, the cycle contains an agent m^2_F. Since m^1_F and therefore also $M^*(m^1_F) = w^1_F$ are not contained in the cycle, this implies that M' matches m_F^2 to a woman to which he prefers w_F^1. Thus, M' is blocked by the pair $\{m_F^2, w_F^1\}$ in this case, a contradiction. As we have exhausted all cases, it follows that M^* is the unique stable matching after the bribery.

($\Leftarrow$) Let S be the set of at most ℓ agents such that modifying their preferences can make M^* the unique stable matching. From this, we construct a solution X to the given Hitting Set problem as follows. For each agent m_z or w_z contained in S, we add the element z to X, and for each agent m_F^i or w_F^i contained in S, we add an arbitrary element $z \in F$ to X. Clearly, $|X| \leq |S| \leq \ell = k$, so it remains to show that X is a hitting set.

Assume that $F \in \mathcal{F}$ does not intersect X. Then the preferences of m_F^i, w_F^i, and all w_z for $z \in F$ are unchanged after the bribery. We claim that $M' := \big(M^* \setminus \{\{m_F^1, w_F^1\}, \{m_F^2, w_F^2\}\}\big) \cup \{\{m_F^1, w_F^2\}, \{m_F^2, w_F^1\}\}$ is then a stable matching, contradicting the fact that M^* is the unique stable matching.

As M^* is stable and M^* and M' only differ in m_F^1, w_F^2, m_F^2, and w_F^1, any blocking pair for M' must contain an agent m_F^i or w_F^i for some $i \in \{1, 2\}$. Agents w_F^1 and m_F^2 are matched to their top-choice and thus are not part of a blocking pair. The only agent which m_F^2 prefers to w_F^1 is w_F^2. However, w_F^2 is matched to her top-choice in M'. Similarly, as after the bribery all agents which m_F^1 prefers to w_F^2 are matched to their top-choices, and thus do not participate in a blocking pair. Thus, no blocking pair for M' exists. $\square$

We now adapt the parameterized reduction presented in the proof of Theorem 6 to show that the *Exact-Unique* problem parameterized by ℓ is also W[2]-hard for the manipulative action *Add*.

Proposition 8. Exact-Unique-Add *parameterized by ℓ is W[2]-hard, and this also holds if the given matching M^* is already stable in the instance restricted to $U_{\text{orig}} \cup W_{\text{orig}}$ and one is only allowed to add agents of one gender.*

Proof. We show Proposition 8 by adapting the reduction from the proof of Theorem 6. We adapt the reduction slightly by changing the preferences of w_z for all $z \in Z$ to $w_z\colon m_{F_1}^1 \succ m_{F_2}^1 \succ \cdots \succ m_{F_p}^1 \succ m_z \succ \overset{\text{(rest)}}{\ldots}$. Moreover, we set $M^* := \{\{m_z, w_z\} : z \in Z\} \cup \{\{m_F^1, w_F^1\}, \{m_F^2, w_F^2\} : F \in \mathcal{F}\}$, we set $\ell := k$, and the agents that can be added to $U_{\text{add}} := \emptyset$ and $W_{\text{add}} := \{w_z : z \in Z\}$.

($\Rightarrow$) Let $X \subseteq Z$ be a hitting set. We set $X_A = \{w_z : z \in X\}$ to be the set of added agents. We claim that $M' = \{\{m_z, w_z\} : z \in X\} \cup \{\{m_F^1, w_F^1\}, \{m_F^2, w_F^2\} : F \in \mathcal{F}\} \subseteq M^*$ is the unique stable matching after adding the agents from X_A. First of all note that M' is stable, as all matched men are matched to their top choice and no woman prefers one of the unassigned man to their assigned partner. The argument why M' is the unique stable matching is analogous to the proof of Theorem 6.

($\Leftarrow$) Let X_A be the set of agents added in a solution to the constructed Exact-Unique-Add instance and let $M' := M^*|_{U_{\text{orig}} \cup W_{\text{orig}} \cup X_A}$ be the unique stable matching. We claim that $X = \{z \in Z : w_z \in X_A\}$ is a hitting set.

Assume that $F \in \mathcal{F}$ does not intersect X. Then $M'' := \big(M' \setminus \{\{m_F^1, w_F^1\}, \{m_F^2, w_F^2\}\}\big) \cup \{\{m_F^1, w_F^2\}, \{m_F^2, w_F^2\}\}$ is also a stable matching by an argument analogous to the proof of Theorem 6, contradicting the fact that M^* is the unique stable matching. $\square$

Finally, we adapt the reduction from Theorem 6 to prove NP-hardness for the manipulative action *Swap*. Here, we utilize the fact that *Reorder* operations can be modeled by (up to $n^2 + n$) *Swap* operations (see Section 2.5 for a high-level discussion of this fact). To do so, we adapt the reduction such that it is only possible to modify the preferences of women w_z and add an "activation cost" to modifying the preferences of w_z such that only the preferences of a fixed number of women can be modified but for these we can modify them arbitrarily.

Proposition 9. Exact-Unique-Swap *is NP-complete, and this also holds if the given matching M^* is already stable in the original instance and we are only allowed to modify the preferences of agents of one gender.*

Proof. Membership in NP is obvious, as it is possible to determine in polynomial time whether a stable matching is unique (as discussed in Proposition 6).

We adapt the reduction from Hitting Set to Exact-Unique-Reorder from Theorem 6 as follows. Let $(Z, \mathcal{F}, k)$ be an instance of Hitting Set, and let $(\mathcal{I}, M^*, \ell)$ be the instance of Exact-Unique-Reorder constructed in the reduction described in the proof of Theorem 6. We assume that $\ell \leq 2n$, as otherwise $\mathcal{I}$ is a trivial YES-instance. Furthermore, we assume $n \geq 3$, since otherwise the Hitting Set instance can be solved by brute force. In the following we modify $(\mathcal{I} = (U, W, \mathcal{P}), M^*, \ell)$ as follows. We add n^5 men m_1^d, ..., $m_{n^5}^d$ and n^5 women w_1^d, ..., $w_{n^5}^d$, where, for all $i \in [n^5]$, the preferences of m_i^d and w_i^d are as follows (indices are taken modulo n^5):

$$m_i^d : w_i^d \succ w_{i+1}^d \succ w_{i+2}^d \succ \cdots \succ w_{n^5+i-1}^d \succ \overset{\text{(rest)}}{\cdots},$$
$$w_i^d : m_i^d \succ m_{i+1}^d \succ m_{i+2}^d \succ \cdots \succ m_{n^5+i-1}^d \succ \overset{\text{(rest)}}{\cdots}.$$

Now, for each $z \in Z$, we add in w_z's preference list n^2 dummy men after m_z, and the remaining dummy men at the end of w_z's preference list. For all other women $w \in W \setminus \{w_z : z \in Z\}$, we insert n^4 dummy men between each two neighboring agents in w's preferences. For each men $m \in U$, we insert n^4 dummy women between any two neighboring agents in m's preferences. We set $M' := M^* \cup \{\{m_i^d, w_i^d\} : i \in [n^5]\}$ and the overall budget to $\ell' := k(n^2 + n)$. The reduction clearly runs in polynomial time, so it remains to show its correctness.

($\Rightarrow$) Let X be a hitting set. For each $z \in X$, we change the preferences of w_z by swapping m_z down $n^2 + n$ times such that after the modification w_z prefers, for all $F \in \mathcal{F}$, the agent m_F^1 to m_z. Thus, the overall number of performed swaps is at most ℓ'. By Lemma 7, any stable matching in the modified instance contains the edges $\{m_i^d, w_i^d\}$ for each $i \in [n^5]$. It follows by the same arguments as in the proof of Theorem 6 that M' is the unique stable matching after the bribery.

($\Leftarrow$) Since $\ell' < n^4$, we can swap pairs containing two non-dummy agents only for agents w_z for some $z \in Z$. Note that swapping the agent m_z with any non-dummy agent in the preferences of w_z requires at least n^2 swaps, and thus, this happens for at most k such agents. The corresponding elements of Z now form a hitting set by arguments analogous to the proof of Theorem 6. □

Remark. In *Exact-Unique*, we are given a matching M^* and want to make this matching stable. However, another possible objective might be to just ensure that there exists a unique stable matching in the manipulated instance, irrespective of which matching is the stable one. We remark that the reductions presented in this section also show W[2]-hardness parameterized by the budget respectively NP-hardness for this objective.

5.2 Algorithms

Contrasting the hardness results for all other manipulative actions, Exact-Unique-DeleteAcceptability turns out to be solvable in polynomial time. On an intuitive level, one reason for this is that we can only delete agents from the preferences of other agents, but we cannot swap the order of agents in the preferences. In particular, we cannot change whether an agent a is before or after $M^*(a')$ in the preferences of a'. Recall that the hardness reductions for *Reorder* and *Swap* crucially rely on this feature. We start this subsection by giving definitions and facts about rotations (Gusfield & Irving, 1989) that we will use afterwards to construct a polynomial-time algorithm for Exact-Unique-DeleteAcceptability and an XP-algorithm for Exact-Unique-Reorder parameterized by ℓ. For more details on rotations, we refer to the monograph of Gusfield and Irving (1989).

For a stable matching M and a man $m \in U$, let $s_M(m)$ denote the first woman w succeeding $M(m)$ in m's preference list who prefers m to $M(w)$. If no such woman exists, then we set $s_M(m) := \emptyset$. A *rotation exposed in a stable matching* M is a sequence $\rho = (m_1, w_1), \ldots, (m_r, w_r)$ such that for each $k \in [r]$ it holds that $\{m_k, w_k\} \in M$ and $w_{k+1} = s_M(m_k)$, where indices are taken modulo r. We call such a rotation a *man-rotation* and $s_M(m)$ the *rotation successor* of m. There is a close relation between rotations and stable matchings (see e.g. Gusfield & Irving, 1989). It is easy to see that given a rotation $(m_1, w_1), \ldots, (m_r, w_r)$ exposed in a stable matching M, matching $M' := \big(M \setminus \{\{m_k, w_k\} : k \in [r]\}\big) \cup \{\{m_k, w_{k+1}\} : k \in [r]\}$ is again a stable matching. We will mainly use the "reverse direction" of this statement, namely that the abscence of rotations can be used to prove the uniqueness of a stable matching.

Example 7. *Consider the SM instance depicted in Fig. 8 in Section 5.1 and let M be the matching in which every man is matched to his top-choice. Then, the rotation successor of $m^1_{\{1,2\}}$ in M is $w^2_{\{1,2\}}$, i.e., $s_M(m^1_{\{1,2\}}) = w^2_{\{1,2\}}$. Moreover, the rotation successor of $m^2_{\{1,2\}}$ is $w^1_{\{1,2\}}$, i.e., $s_M(m^2_{\{1,2\}}) = w^1_{\{1,2\}}$. Thus, $(m^1_{\{1,2\}}, w^1_{\{1,2\}}), (m^2_{\{1,2\}}, w^2_{\{1,2\}})$ is a rotation exposed in the stable matching M, which proves that M is not the unique stable matching.*

We define the rotation successor $s_W(w)$ of $w \in W$ analogously and call a rotation where the roles of men and women are switched *woman-rotation*. As a matching is unique if and only if it exposes neither a man-rotation nor a woman-rotation (Gusfield & Irving, 1989), we can reformulate the goal of Exact-Unique-DeleteAcceptability: Modify the given SM instance by deleting the acceptability of at most ℓ pairs such that neither a man-rotation nor a woman-rotation is exposed in M^*.

We start by making two straightforward observations:

Observation 2. *To determine whether a stable matching M is the unique stable matching, it is enough to know the rotation successors of all agents.*

This observation gives rise to a simple algorithm to check whether a stable matching M exposes a man-rotation. We create a sink t and for each man $m \in U$ a vertex v_m. We insert an arc from v_m to $v_{m'}$ if $M(m')$ is m's rotation successor in M, i.e., $M(m') = s_M(m)$, and an arc from v_m to t if m does not have a rotation successor. Then, checking whether M exposes a man-rotation reduces to checking whether there exists a cycle in the constructed directed graph.

In the following, for an agent $a \in A$, we refer to its preferences induced by the set of all agents it prefers to $M(a)$ as the *first part* of its preferences and to its preferences induced by the set of all agents to which it prefers $M(a)$ as the *second part* of its preferences. Using this notation, we can make the following observation, which holds also if the roles of women and men are switched:

Observation 3. *To determine whether a stable matching M exposes a man-rotation, it is enough to know the first part of the women's preferences and the second part of the men's preferences.*

A simple approach to solve the Exact-Unique-DeleteAcceptability problem would be to start by computing the set of rotations in the given matching and delete the acceptability of one pair in each rotation. However, thereby, we would change the rotation successors of some agents, which could lead to new rotations. That is why we need to be careful when choosing the pair within each rotation which one wants to delete.

To circumvent this issue, we first observe that, by Observation 3, it is never beneficial to delete the acceptability of some pair $\{m, w\}$ with $m \in U$ and $w \in W$ if m and w both appear in the same part of each others preferences, as this implies that none of them can be the rotation successor of the other. Moreover, it is possible to separately solve the problem of ensuring that the given matching M^* does not expose a man-rotation and the problem of ensuring that the given matching M^* does not expose a woman-rotation. To solve the former problem, we only care about the rotation successors of all men. Thereby, we only delete the acceptability of pairs $\{m, w\}$ where w appears in the second part of m's preferences and m appears in the first part of w's preferences. For woman-rotations, the situation is symmetric. We solve both problems by reducing them to the Minimum Weight Spanning Anti-Arborescence problem. In an instance of the Minimum Weight Spanning Anti-Arborescence problem, we are given a directed graph $G = (V, E)$ with arc costs and a budget $k \in \mathbb{N}$. The question is whether there exists a spanning anti-arborescence, i.e., an acyclic subgraph of G such that all vertices of G but one have out-degree exactly one, of cost at most k. Minimum Weight Spanning Anti-Arborescence can be solved in $\mathcal{O}(|E| + |V| \log |V|)$ time (Edmonds, 1967; Gabow, Galil, Spencer, & Tarjan, 1986).

Given an SM instance $\mathcal{I} = (U, W, \mathcal{P})$ and a matching M^*, the basic idea of the algorithm is the following (we present the algorithm for excluding man-rotations; excluding woman-rotations can be done symmetrically): For a set F of deleted acceptabilities, let $s^F_{M^*}(m)$ denote the rotation successor of m after the deletion of F. To ensure that there is no man-rotation, we need to find a set F of deleted acceptabilities such that the graph where the agents form the vertex set and the arc set is $\{(w, m) : m \in U \wedge w \in W \wedge \{m, w\} \in M^*\} \cup \{(m, s^F_{M^*}(m)) : m \in U \text{ with } s^F_{M^*}(m) \neq \emptyset\}$ is acyclic. Note that we can change $s_{M^*}(m)$ only by deleting the pair $\{m, s_{M^*}(m)\}$. In this case, the new rotation successor becomes the first woman w' succeeding $s_{M^*}(m)$ in m's preference list which prefers m to $M^*(w')$. Thus, the costs of making a woman w' the rotation successor of m is the number of women w such that m prefers $M^*(m)$ to w and w to w' and w prefers m to $M^*(w)$. We now argue that the problem of making $\{(w,m) : m \in U \wedge w \in W \wedge \{m,w\} \in M^*\} \cup \{(m, s^F_{M^*}(m)) : m \in U \text{ with } s^F_{M^*}(m) \neq \emptyset\}$ acyclic can be expressed as an instance of Minimum Weight Spanning Anti-Arborescence.

Theorem 7. Exact-Unique-DeleteAcceptability *can be solved in $\mathcal{O}(n^2)$ time.*

Proof. Clearly, any solution needs to delete all blocking pairs. Thus, we assume without loss of generality that M^* is a stable matching.

Given an instance $(\mathcal{I} = (U, W, \mathcal{P}), M^*, \ell)$ of Exact-Unique-DeleteAcceptability, we reduce the problem to two instances of the Minimum Weight Spanning Anti-Arborescence problem. The first instance of this problem that is responsible for deleting all man-rotations is constructed as follows. The graph contains a vertex v_m for each edge $\{m,w\} \in M^*$ as well as a sink t. We add an arc $(v_m, v_{m'})$ if $M^*(m')$ prefers m to m' and m prefers $M^*(m)$ to w'. The weight of this arc is the number of women w^* such that m prefers w^* to $M^*(m')$ and $M^*(m)$ to w^*, and w^* prefers m to $M^*(w^*)$ (i.e., the number of acceptabilities which need to be deleted to make $M^*(m')$ the rotation successor of m). Furthermore, there is an arc (v_m, t) for all $\{m,w\} \in M^*$. The weight of this arc is the number of women w^* such that m accepts w^* and prefers $M^*(m)$ to w^*, and w^* prefers m to $M^*(w^*)$ (i.e., the number of acceptabilities which need to be deleted to make $\emptyset$ the rotation successor of m). We call this graph H_U. Similarly, we construct a graph H_W (where the roles of men and women are exchanged).

We claim that M^* can be made the unique stable matching after the deletion of ℓ pairs if and only if the minimum weight anti-arborescences in H_U and H_W together have weight at most ℓ.

($\Rightarrow$) Let $F \subseteq \{\{m,w\} : m \in U, w \in W\}$ be a set of at most ℓ pairs whose deletion make M^* the unique stable matching. Let $F_W := \{\{m,w\} \in F : w \succ_m M^*(m)\}$ and $F_U := \{\{m,w\} \in F : m \succ_w M^*(w)\}$. For any man m, let $s_m := v_{M^*(w')}$, where w' is the woman best-ranked by m succeeding $M^*(m)$ such that w prefers m to $M^*(w)$ after the manipulation, i.e,. w' is the rotation successor of m after the manipulation. If no such woman exists, then we set $s_m := t$. We construct an anti-arborescence in $\mathcal{A}_U$ of cost at most $|F_U|$ by adding for each pair $\{m,w\} \in M^*$ the arc (v_m, s_m) to the anti-arborescence. We claim that $\mathcal{A}_U$ is an anti-arborescence. Every vertex but t has exactly one outgoing arc, so it is enough to show that there does not exist a cycle. As we have inserted for each man an arc from the node including him to the node including his rotation successor, there cannot exist any cycle in the anti-arborescence, as such a cycle would induce an exposed man-rotation in the modified SM instance which would contradict the uniqueness of M^* in the modified SM instance.

In the same way one can construct an anti-arborescence of cost $|F_W|$ in H_W. The constructed anti-arborescences together have weight at most $|F_W| + |F_U| \leq |F| \leq \ell$, as any arc in $F_W \cap F_U$ would be a blocking pair for M^*.

($\Leftarrow$) Let $\mathcal{A}_U$ be an anti-arborescence in H_U, and $\mathcal{A}_W$ be an anti-arborescence in H_W. For every arc $(v_m, t) \in \mathcal{A}_U$, we delete the acceptability of all pairs $\{m, w'\}$ with m preferring $M^*(m)$ to w' and w' preferring m to $M^*(w')$. For every arc $(v_m, v_{\widetilde{m}}) \in \mathcal{A}_U$, we delete the acceptability of all pairs $\{m, w'\}$ with m preferring w' to $M^*(\widetilde{m})$, and $M^*(m)$

to w', and w' preferring m to $M^*(w')$. After these deletions, $\widetilde{w}$ is the rotation successor of m. Let F_U denote the set of pairs deleted. We proceed with $\mathcal{A}_W$ analogously, and denote as F_W the set of deleted pairs. By construction, $\mathcal{A}_U$ has cost $|F_U|$, and $\mathcal{A}_W$ has cost $|F_W|$.

Assume that M^* is not the unique stable matching after deleting the pairs from $F_U \cup F_W$. Then, without loss of generality, a man-rotation is exposed in M^*: $(m_{i_1}, w_{j_1}), \dots, (m_{i_r}, w_{j_r})$. As we already observed, the anti-aborescence $\mathcal{A}_U$ contains all arcs $(v_m, v_{\widetilde{m}})$ where $\widetilde{w}$ is m's rotation-successor (after the deletion of F_U). Thus, $\mathcal{A}_U$ contains the arcs $(v_{m_{i_k}}, v_{m_{i_{k+1}}})$ for all $k \in [r]$ (all indices are taken modulo r). This implies that $\mathcal{A}_U$ contains a cycle, a contradiction to $\mathcal{A}_U$ being an anti-arborescence. □

We conclude this section by constructing an XP algorithm for Exact-Unique-Reorder parameterized by ℓ which runs in $\mathcal{O}(2^\ell n^{2\ell+2})$ time. This algorithm is described in Algorithm 3 and requires as input an instance of Exact-Unique-Reorder consisting of an SM instance $\mathcal{I} = (U, W, \mathcal{P})$, a complete matching M^*, and a budget ℓ.

The algorithm starts by guessing the subsets of men $X_U \subseteq U$ and women $X_W \subseteq W$ of summed size ℓ whose preferences we reorder (Line 1). We reject a guess if there exists a blocking pair $\{m, w\} \in \mathrm{bp}(M^*, \mathcal{I})$ such that $m \notin X_U$ and $w \notin X_W$, as in this case it is not possible to resolve this blocking pair using the guessed agents (Line 5). Moreover, for each $m \in X_U$, we guess his rotation-successor $s^{\text{Reorder}}_{M^*}(m) \in W \cup \{\emptyset\}$ after the re-orderings and for each $w \in X_W$, we guess her rotation successor $s^{\text{Reorder}}_{M^*}(w) \in U \cup \{\emptyset\}$ (Lines 2 and 3). We reject the guess if it is impossible for an agent $a \in X_U \cup X_W$ to make $s^{\text{Reorder}}_{M^*}(a)$ a's rotation successor. This is the case if $s^{\text{Reorder}}_{M^*}(a) \notin X_U \cup X_W$ and $s^{\text{Reorder}}_{M^*}(a)$ prefers $M(s^{\text{Reorder}}_{M^*}(a))$ to a (Line 7). Further, if there exists some $m \in X_U$ for which we have guessed that $s^{\text{Reorder}}_{M^*}(m) = \emptyset$ and there exists some $w \in W \setminus X_W$ that prefers m to $M^*(w)$, then we reject the guess, as in this case if m ranks w above $M^*(m)$, we create a blocking pair and if m ranks $M^*(m)$ above w, then m has a rotation successor; the same holds with roles of women and men swapped (Line 9). We also check whether there is a man-woman pair (m, w) with $m \in X_U$ and $w \in X_W$ such that we guessed that they are their mutual rotation successor and reject a guess in this case (Line 11), because if w is the rotation successor of m, then m prefers $M^*(m)$ to w, and if m is the rotation successor of w, then m prefers w to $M^*(m)$ and these two conditions clearly cannot be satisfied at the same time.

In the end, for all agents $a \in X_U \cup X_W$, we will reorder their preferences such that their rotation successor is ranked directly after $M^*(a)$. The only influence that the preferences of a can have on the rotation successor of another agent a' is whether a prefers a' to $M^*(a)$ or not. By Observation 3, it follows that in order to make M^* the unique stable matching, we only need to decide which agents are in the first part of a's preferences (and can then order them arbitrarily before $M^*(a)$ in the preferences of a). Again, by Observation 3, this can be solved for the men in X_U and women in X_W separately (with the exception that we need to ensure that there is no man-woman pair which are mutual rotation successors; however, this may happen only if both agents are contained in $X_U \cup X_W$, and we exclude that this happens in Line 11): Selecting the first part of the preferences of $w \in X_W$ only influences the rotation successors of all men. Similarly, selecting the first part of the preferences of $m \in X_U$ only influences whether there exists a woman-rotation. Consequently, it is possible to split the problem into two parts. We describe how to determine the preferences

Input: An SM instance $\mathcal{I} = (U, W, \mathcal{P})$, a complete matching M^*, and a budget ℓ.

1 Guess sets $X_U \subseteq U$ and $X_W \subseteq W$ with $|X_U \cup X_W| \leq \ell$;

2 For each $m \in X_U$, guess $s^{\text{REORDER}}_{M^*}(m) \in W \cup \{\emptyset\}$;

3 For each $w \in X_W$, guess $s^{\text{REORDER}}_{M^*}(w) \in M \cup \{\emptyset\}$;

4 **if** *there exists a blocking pair* $\{m, w\}$ *for* M^* *with* $m \notin X_U$ *and* $w \notin X_W$ **then**

5   Reject this guess;

6 **if** *there exists an agent* $a \in X_U \cup X_W$ *with* $s^{\text{REORDER}}_{M^*}(a) \notin X_U \cup X_W$ *and* $s^{\text{REORDER}}_{M^*}(a)$ *preferring* $M(s^{\text{REORDER}}_{M^*}(a))$ *to* a **then**

7   Reject this guess;

8 **if** *there exists* $m \in X_U$ *and a woman* $w \in W \setminus X_W$ *such that* $s^{\text{REORDER}}_{M^*}(m) = \emptyset$ *and* w *prefers* m *to* $M^*(w)$ *or there exists* $w \in X_W$ *and* $m \in U \setminus X_U$ *such that* $s^{\text{REORDER}}_{M^*}(w) = \emptyset$ *and* m *prefers* w *to* $M^*(m)$ **then**

9   Reject the guess;

10 **if** *there exists* $m \in X_U$ *and* $w \in X_W$ *such that* $m = s^{\text{REORDER}}_{M^*}(w)$ *and* $w = s^{\text{REORDER}}_{M^*}(m)$ **then**

11   Reject the guess;

12 Let H be an empty directed graph;

13 Add a vertex t to H;

14 **foreach** $\{m, w\} \in M^*$ **do**

15   Add a vertex v_m to H;

16 **foreach** $m \in X_U$ **do**

17   Add arc (v_m, v) to H, where $v := t$ if $s^{\text{REORDER}}_{M^*}(m) = \emptyset$ and $v := v_{m'}$ with $m' := M^*(s^{\text{REORDER}}_{M^*}(m))$ otherwise;

18 **foreach** $m \in U \setminus X_U$ **do**

19   Let $\widetilde{s}_{M^*}(m) \in W \setminus X_W$ be the woman from $W \setminus X_W$ which m likes most such that m prefers $M^*(m)$ to $\widetilde{s}_{M^*}(m)$, and $\widetilde{s}_{M^*}(m)$ prefers m to $M^*\big(\widetilde{s}_{M^*}(m)\big)$;

20   **if** *no such* $\widetilde{s}_{M^*}(m)$ *exists* **then**

21   Add (v_m, t) to H;

22   **foreach** $w \in X_W$ *such that* m *prefers* $M^*(m)$ *to* w *and* $s^{\text{REORDER}}_{M^*}(w) \neq m$ **do**

23   Add arc $(v_m, v_{m'})$ to H, where $m' := M^*(w)$;

24   **else**

25   Add arc $(v_m, v_{m'})$ to H, where $m' := M^*(\widetilde{s}_{M^*}(m))$;

26   **foreach** $w \in X_W$ *such that* m *prefers* $M^*(m)$ *to* w *to* $\widetilde{s}_{M^*}(m)$ *and* $s^{\text{REORDER}}_{M^*}(w) \neq m$ **do**

27   Add arc $(v_m, v_{m'})$ to H, where $m' := M^*(w)$;

28 **if** H *does not contain a spanning anti-arborescence* **then**

29   Reject this guess;

30 Repeat Lines 12–29 with roles of women and men swapped;

31 Accept this guess;

Algorithm 3: An XP algorithm wrt. ℓ for EXACT-UNIQUE-REORDER.

of all $w \in X_W$, thereby, resolving all man-rotations. The woman-rotations can be resolved symmetrically (Line 30).

To determine how to reorder the preferences of all $w \in X_W$, we reduce the problem to an instance of Spanning Anti-Arborescence (Lines 12–27). We construct the directed graph as follows. For each pair $\{m, w\} \in M^*$, we introduce a vertex v_m (Line 15). Moreover, we add a sink t (Line 13). For all $m \in X_U$, we add an arc from v_m to the vertex corresponding to the man matched to the guessed rotation successor of m in M^*, i.e., $v_{M^*(s_{M^*}^{\text{Reorder}}(m))}$, or to t if $s_{M^*}^{\text{Reorder}}(m) = \emptyset$ (Line 17). Now, we add for each two vertices v_m and $v_{m'}$ with $m \in U \setminus X_U$ and $m' \in X_U$ an arc from v_m to $v_{m'}$ if we can reorder the guessed agents' preferences such that $M^*(m')$ is m's rotation successor (Lines 18–27).

More formally, for each $m \in U \setminus X_U$, we denote as $\widetilde{s}_{M^*}(m)$ agent m's most-preferred woman $w \in W \setminus X_W$ who prefers m to $M^*(w)$ and is ranked after $M^*(m)$ by m (independent of how the preferences of the guessed agents are reordered, m cannot have a rotation successor to which he prefers $\widetilde{s}_{M^*}(m)$). We deal with the case that $\widetilde{s}_{M^*}(m)$ is not well-defined separately below. Now, for each $w' \in X_W$, who is ranked between $M^*(m)$ and $\widetilde{s}_{M^*}(m)$ in the preferences of m and fulfills $s_{M^*}^{\text{Reorder}}(w') \neq m$, we add an arc from v_m to $v_{M^*(w')}$ (for those women, we can decide whether they rank m before or after $M^*(w')$ and thus whether they become m's rotation successor or not). Note that we require $s_{M^*}^{\text{Reorder}}(w') \neq m$ as otherwise w' must prefer $M^*(w')$ to m and thus cannot be the rotation successor of m. Moreover, we add an arc from v_m to $v_{M^*(\widetilde{s}_{M^*}(m))}$ (Lines 25 and 27). As mentioned above, it may happen that $\widetilde{s}_{M^*}(m)$ is undefined for some $m \in U \setminus X_U$. In this case, we add an arc between v_m and $v_{M^*(w')}$ for each $w' \in X_W$ which m ranks below $M^*(m)$ and fulfills $s_{M^*}^{\text{Reorder}}(w') \neq m$ and an arc from v_m to t (Lines 21 and 23). We call the resulting graph H_U and the graph constructed using the same algorithm with the roles of men and women switched H_W.

We compute an anti-arborescence in H_U. In the anti-arborescence, for each v_m, the end point of its outgoing arc corresponds to the rotation successor of m in the modified instance. To ensure this, we construct the preferences of all women $w \in X_W$ as follows. For each $w \in X_W$, we rank all men $m \in U$ such that there is an arc from v_m to $v_{M^*(w)}$ in the anti-arborescence in an arbitrary order before $M^*(w)$, while we place the guessed man $s_{M^*}^{\text{Reorder}}(w)$ directly after $M^*(w)$ and add the remaining agents in an arbitrary order after $s_{M^*}^{\text{Reorder}}(w)$. If $s_{M^*}^{\text{Reorder}}(w) = \emptyset$, then we place all men $m \in U$ such that there is an arc from v_m to $v_{M^*(w)}$ in the anti-arborescence before $M^*(w)$ and all other men in an arbitrary order after $M^*(w)$.

We use the same procedure to determine the preferences of all $m \in X_U$. Thereby, if there exist anti-arborescences in H_U and H_W, we are able to reorder the preferences of the guessed agents such that M^* becomes the unique stable matching. Thus, we return YES in this case. Otherwise, we reject this guess (Lines 29 and 30), continue with the next guess and return NO after rejecting the last guess.

It remains to prove the correctness of the algorithm:

Lemma 11. *If the algorithm accepts a guess, then there exists a solution to the given instance of* Exact-Unique-Reorder.

Proof. We now prove that for every pair $(\mathcal{A}_U, \mathcal{A}_W)$ of anti-arborescences found in the graphs H_U and H_W, the resulting reorderings of the preferences make M^* the unique stable matching. First we show that the preferences are well-defined: The only case in which we require for two agents a and a' that a prefers a' to $M^*(a')$ and a prefers $M^*(a')$ to a' is when a' is the rotation successor of a and a is the rotation successor of a'. However, due to the check in Line 11, this is not possible. For the sake of contradiction, let us assume that M^* is not the unique stable matching. There are two possibilities, either M^* is not a stable matching or M^* is not the unique stable matching.

First, we show that M^* is stable. Since we rejected each guess containing a blocking pair $\{m, w\}$ with $m \in U \setminus X_U$ and $w \in W \setminus X_W$, each blocking pair involves at least one agent from $X_U \cup X_W$. Fix a blocking pair $\{m, w\}$. We assume without loss of generality that $m \in X_U$. The algorithm constructs the preferences of m such that m only prefers a woman w to $M^*(m)$ if the arc from v_w to $v_{M^*(m)}$ is part of the anti-arborescence $\mathcal{A}_W$. However, such an arc only exists in H_W if w ranks m below $M^*(w)$, which implies that $\{m, w\}$ cannot be blocking.

Now we show that M^* is the unique stable matching. To do so, we show that for each man m with $(v_m, v_{M^*(w)}) \in \mathcal{A}_U$, the rotation successor of m is the women w. If $\mathcal{A}_U$ contains the arc (v_m, t), then m has no rotation successor. A symmetric statement also holds for each woman. From this, the uniqueness of M^* easily follows, as for any rotation (without loss of generality a man-rotation) $(m_{i_1}, w_{j_1}), \ldots, (m_{i_r}, w_{j_r})$ exposed in M^*, the anti-arborescence $\mathcal{A}_U$ then contains the arcs $(v_{m_{i_j}}, v_{m_{i_{j+1}}})$ for all $j \in [r]$, and therefore contains a cycle, a contradiction to $\mathcal{A}_U$ being an anti-arborescence.

So consider a man m with $(v_m, v_{M^*(w)}) \in \mathcal{A}_U$. By the definition of H_U, man m prefers $M^*(m)$ to w. We claim that for every woman $w' \in W$ whom m ranks between $M^*(m)$ and w, it holds that w' prefers $M^*(w')$ to m after the modifications. If $w' \in X_W$, then we reorder the preferences of w' in this way; otherwise, this follows since H_U contains edge $(v_m, v_{M^*(w)})$. It remains to show that w prefers m to $M^*(w)$. If $w \in X_W$, then we reordered the preferences of w such that w prefers m to $M^*(w)$. Otherwise, woman w prefers m to $M^*(w)$ since H_U contains edge $(v_m, v_{M^*(w)})$. If $(v_m, t) \in \mathcal{A}_U$, then analogous arguments show that any woman w after $M^*(m)$ in m's preferences does not prefer m to $M^*(w)$. □

Lemma 12. *If the algorithm rejects every guess, then there exists no solution to the given instance of* Exact-Unique-Reorder.

Proof. For the sake of contradiction, let us assume that the given instance of Exact-Unique-Reorder admits a solution. We claim that there exists a guess of $X_U \cup X_W$ and their rotation successors for which H_U and H_W both admit anti-arborescences. This leads to a contradiction, thereby proving the lemma.

Assume that there exists a set $Y_A = Y_U \cup Y_W$ with $Y_U \subseteq U$ and $Y_W \subseteq W$ of ℓ agents and a reordering of the preferences of these agents such that M^* is the unique stable matching in the resulting instance. Let $\mathcal{I}'$ denote the manipulated instance and let $s^{\mathcal{I}'}_{M^*}(m)$ denote the rotation successor of some $m \in U$ in $\mathcal{I}'$ and $s^{\mathcal{I}'}_{M^*}(w)$ the rotation successor of some $w \in W$. Then, there exists a guess where $X_U = Y_U$, $X_W = Y_W$, and for all $m \in X_U$, his rotation successor is $s^{\mathcal{I}'}_{M^*}(m)$, i.e., $s^{\text{Reorder}}_{M^*}(m) = s^{\mathcal{I}'}_{M^*}(m)$, and for all $w \in X_U$, her rotation successor is $s^{\mathcal{I}'}_{M^*}(w)$. First of all note that the guess is not immediately rejected, as for each blocking pair one of the involved agents needs to be part of $X_U \cup X_W$, and no agent from X_A without a rotation successor can be preferred by an unmodified agent of opposite gender to its partner in M^*. Furthermore, there cannot be a man-woman pair (m, w) with $m \in X_U$ and $w \in X_W$ such that they are to be their mutual rotation successor: If w is the rotation successor of m, then m prefers $M^*(m)$ to w, and if m is the rotation successor of w, then m prefers w to $M^*(m)$ and these two conditions clearly cannot be satisfied at the same time.

We describe how to construct an anti-arborescence $\mathcal{A}_U$ for H_U while the construction for $\mathcal{A}_W$ works analogously. For each $m \in U$, we include the arc $(v_m, v_{M^*(s^{\mathcal{I}'}_{M^*}(m))})$ in $\mathcal{A}_U$ and the arc (v_m, t) if $s^{\mathcal{I}'}_{M^*}(m) = \emptyset$. As there is no rotation exposed in M^* in $\mathcal{I}'$, the resulting graph $\mathcal{A}_U$ is acyclic and every vertex but t has out-degree exactly 1, i.e., $\mathcal{A}_U$ is indeed an anti-arborescence.

It remains to show that $\mathcal{A}_U$ is a subgraph of H_U. Fix an arc $(v_m, v_{M^*(s^{\mathcal{I}'}_{M^*}(m))}) \in \mathcal{A}_U$ (where $v_{M^*(\emptyset)} := t$). For all $m \in X_U$, this arc is contained in H_U, as we have already guessed $s^{\mathcal{I}'}_{M^*}(m)$ and added the arc $(v_m, v_{M^*(s^{\mathcal{I}'}_{M^*}(m))})$ to H_U. For all $m \in U \setminus X_U$, the woman $s^{\mathcal{I}'}_{M^*}(m)$ needs to be ranked below $M^*(m)$ in m's preferences. Moreover, by definition, $s^{\mathcal{I}'}_{M^*}(m)$ is the first woman after $M^*(m)$ in m's preferences who prefers m over $M^*(w)$. Thereby, w cannot be ranked after the first woman $\widetilde{s}_{M^*}(m) \in W \setminus X_W$ who prefers m to $M^*(\widetilde{s}_{M^*}(m))$. Thus, if w is not $\widetilde{s}_{M^*}(m)$, then w is contained in X_W. In fact, for all such women w there exists an arc from v_m to $v_{M^*(w)}$ in H_U. □

The developed algorithm runs in $\mathcal{O}(2^\ell n^{2\ell+2})$ time since we iterate over up to $\binom{2n}{\ell}$ guesses for X_A and for each of these guesses, we iterate over $\mathcal{O}(n^\ell)$ guesses for the rotation successors. For each guess of X_A and the rotation successors, graph H and the anti-arborescence can be computed in $\mathcal{O}(n^2)$ time. Altogether, the following theorem results from Lemma 11 and Lemma 12:

Theorem 8. Exact-Unique-Reorder *is solvable in $\mathcal{O}(2^\ell n^{2\ell+2})$ time.*

6. Conclusion

We provided a first comprehensive study of the computational complexity of several manipulative actions and goals in the context of the Stable Marriage problem. Our diverse set of computational complexity results is surveyed in Table 1.

Several challenges for future research remain. In contrast to the constructive setting considered here, there is also a destructive view on manipulation, where the goal is to prevent a certain constellation (see Section 3.3.2 for a discussion which of our results translate). Moreover, for the *Constructive-Unique* scenario not presented here (where one edge shall be contained in every stable matching), our hardness results for *Constructive-Exists* carry over, as the stable matchings constructed in the proofs of Theorems 1 to 3 are indeed unique. However, the algorithm for Constructive-Exists-Delete as well as the 2-approximation algorithm for Constructive-Exists-Reorder do not work for *Constructive-Unique*. A very specific open question is whether the Exact-Unique-Swap problem is fixed-parameter tractable when parameterized by the budget. Additionally, there is clearly a lot of room for investigating more manipulative actions. For instance, a manipulator might be able to divide the set of agents into two parts, where a separate matching for each part needs to be found and agents from different parts cannot form a blocking pair. Further, in the presence of ties, assuming a less powerful manipulator, a manipulator might only be able to break ties in the preferences (whether this manipulation can have an impact depends however on the concrete stability concept and the manipulation goal considered). Notably, in several matching mechanisms used in practice, e.g., in the context of matching students to schools in Estonia (Triin, Põder, & Veski, 2014) and in several larger US cities (Erdil & Ergin, 2008) and in the context of assigning residents to hospitals in Scotland (Irving, 2011), ties in the agent's preferences are broken uniformly at random. Clearly, one may also extend the study of external manipulation to stable matching problems beyond Stable Marriage. A further natural extension of our work would be to consider weighted manipulation, where one assumes that each possible manipulative action comes at a specific cost. All our results for the two *Exact* settings as well as the hardness results for *Constructive* carry over to the weighted case in a straightforward way, while the polynomial-time algorithm for Constructive-Exists-Delete does not work anymore.

Lastly, it might also be interesting to analyze the power of different manipulative actions in real-world scenarios. We already did some very preliminary experiments for all polynomial-computable cases on synthetic data having between 30 and 200 agents, where the preferences of agents were drawn uniformly at random from all possible preferences. The manipulation goal was also set uniformly at random. The following two observations were particularly surprising to us: In the *Constructive-Exists* setting, *Delete* operations seem to be quite powerful, as most of the time deleting a moderately low number of agents (around 10%) sufficed. In the *Exact-Exists* setting, *Reorder* operations are not as powerful as one might intuitively suspect, as, on average, close to half of the agents needed to be modified—note that there always exists a trivial solution where the preferences of all agents from one gender are reordered.

Acknowledgments

An extended abstract of the paper appeared in the proceedings of the *13th International Symposium on Algorithmic Game Theory (SAGT 2020)*, Springer LNCS 12283, pages 163–177, 2020. Main work done while Robert Bredereck was with Technische Universität Berlin. Niclas Boehmer was supported by the DFG project MaMu (NI 369/19). Klaus Heeger was supported by the DFG Research Training Group 2434 "Facets of Complexity".

References

Anwar, A. A., & Bahaj, A. S. (2003). Student project allocation using integer programming. *IEEE Transactions on Education*, *46*(3), 359–367.

Aziz, H., Seedig, H. G., & von Wedel, J. K. (2015). On the susceptibility of the deferred acceptance algorithm. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*, pp. 939–947.

Bartholdi III, J. J., Tovey, C. A., & Trick, M. A. (1992). How hard is it to control an election?. *Mathematical and Computer Modelling*, *16*(8-9), 27–40.

Boehmer, N., Bredereck, R., Faliszewski, P., & Niedermeier, R. (2021). Winner robustness via swap- and shift-bribery: Parameterized counting complexity and experiments. In

Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI '21), pp. 52–58. ijcai.org.

Boehmer, N., Bredereck, R., Knop, D., & Luo, J. (2020). Fine-grained view on bribery for group identification. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI '20)*, pp. 67–73.

Cechlárová, K., & Fleiner, T. (2005). On a generalization of the stable roommates problem. *ACM Transactions on Algorithms*, *1*(1), 143–156.

Chen, J., Skowron, P., & Sorge, M. (2019). Matchings under preferences: Strength of stability and trade-offs. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC '19)*, pp. 41–59.

Clark, S. (2006). The uniqueness of stable matchings. *Contributions in Theoretical Economics*, *6*(1), 8.

Consuegra, M. E., Kumar, R., & Narasimhan, G. (2013). Comment on "on the uniqueness of stable marriage matchings" [Economic Letters 69(1):1–8, 2000]. *Economics Letters*, *121*(3), 468.

Cseh, Á., & Heeger, K. (2020). The stable marriage problem with ties and restricted edges. *Discrete Optimization*, *36*, 100571.

Downey, R. G., & Fellows, M. R. (2013). *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer.

Drgas-Burchardt, E. (2013). A note on the uniqueness of stable marriage matching. *Discussiones Mathematicae. Graph Theory*, *33*(1), 49–55.

Edmonds, J. (1967). Optimum branchings. *Journal of Research of the National Bureau of Standards B*, *71*(4), 233–240.

Eeckhout, J. (2000). On the uniqueness of stable marriage matchings. *Economics Letters*, *69*(1), 1–8.

Erdil, A., & Ergin, H. (2008). What's the matter with tie-breaking? Improving efficiency in school choice. *American Economic Review*, *98*(3), 669–89.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. A. (2009). How hard is bribery in elections?. *Journal of Artificial Intelligence Research*, *35*, 485–532.

Faliszewski, P., & Rothe, J. (2016). Control and bribery in voting. In *Handbook of Computational Social Choice*, pp. 146–168. Cambridge University Press.

Faliszewski, P., Skowron, P., & Talmon, N. (2017). Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17)*, pp. 6–14.

Gabow, H. N., Galil, Z., Spencer, T. H., & Tarjan, R. E. (1986). Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, *6*(2), 109–122.

Gai, A., Mathieu, F., de Montgolfier, F., & Reynier, J. (2007). Stratification in P2P networks: Application to BitTorrent. In *Proceedings of the 27th IEEE International Conference on Distributed Computing Systems (ICDCS '07)*.

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

Gelain, M., Pini, M. S., Rossi, F., Venable, K. B., & Walsh, T. (2011). Male optimal and unique stable marriages with partially ordered preferences. In *Proceedings of the First and Second International Workshop on Collaborative Agents, Research and Development (CARE '09/'10)*, pp. 44–55.

Gonczarowski, Y. A. (2014). Manipulation of stable matchings using minimal blacklists. In *Proceedings of the 15th ACM Conference on Economics and Computation (EC '14)*, p. 449.

Gusfield, D. (1987). Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, *16*(1), 111–128.

Gusfield, D., & Irving, R. W. (1989). *The Stable Marriage Problem - Structure and Algorithms*. Foundations of Computing Series. MIT Press.

Heyneman, S., Anderson, K., & Nuraliyeva, N. (2008). The cost of corruption in higher education. *Comparative Education Review*, *52*(1), 1–25.

Hopcroft, J. E., & Karp, R. M. (1973). An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, *2*(4), 225–231.

Hosseini, H., Umar, F., & Vaish, R. (2021). Accomplice manipulation of the deferred acceptance algorithm. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI '21)*, pp. 231–237. ijcai.org.

Hussain, S., Gamage, K. A., Sagor, M. H., Tariq, F., Ma, L., & Imran, M. A. (2019). A systematic review of project allocation methods in undergraduate transnational engineering education. *Education Sciences*, *9*(4), 258.

Irving, R. (2011). Matching practices for entry-labor markets – Scotland.. `https://www.matching-in-practice.eu/wp-content/uploads/2013/05/MiP_-Profile_No.3.pdf`.

Irving, R. W., Manlove, D., & Scott, S. (2003). Strong stability in the hospitals/residents problem. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS '03)*, pp. 439–450.

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center*, pp. 85–103.

Kazakov, D. (2001). Coordination of student project allocation.. `https://www-users.cs.york.ac.uk/kazakov/papers/proj.pdf`.

King, V., Rao, S., & Tarjan, R. E. (1994). A faster deterministic maximum flow algorithm. *Journal of Algorithms*, *17*(3), 447–474.

Knuth, D. E. (1976). *Mariages stables et leurs relations avec d'autres problèmes combinatoires*. Les Presses de l'Université de Montréal, Montreal, Que. Introduction à l'analyse mathématique des algorithmes, Collection de la Chaire Aisenstadt.

Lebedev, D., Mathieu, F., Viennot, L., Gai, A., Reynier, J., & de Montgolfier, F. (2007). On using matching theory to understand P2P network design. In *Proceedings of the International Network Optimization Conference 2007 (INOC '07)*.

Liu, Q., & Peng, Y. (2015). Corruption in college admissions examinations in China. *International Journal of Educational Development*, *41*, 104–111.

Mai, T., & Vazirani, V. V. (2018a). Finding stable matchings that are robust to errors in the input. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA '18)*, pp. 60:1–60:11.

Mai, T., & Vazirani, V. V. (2018b). Stable matchings, robust solutions, and finite distributive lattices. *arXiv preprint*, *arXiv:1804.05537 [cs.DM]*.

Manlove, D., Irving, R. W., Iwama, K., Miyazaki, S., & Morita, Y. (2002). Hard variants of stable marriage. *Theoretical Computer Science*, *276*(1-2), 261–279.

Manlove, D. F. (2013). *Algorithmics of Matching Under Preferences*, Vol. 2 of *Series on Theoretical Computer Science*. WorldScientific.

Orlin, J. B. (2013). Max flows in $O(nm)$ time, or better. In *Proceedings of the 45th ACM Symposium on Theory of Computing (STOC '13)*, pp. 765–774.

Pini, M. S., Rossi, F., Venable, K. B., & Walsh, T. (2011). Manipulation complexity and gender neutrality in stable marriage procedures. *Autonomous Agents and Multi-Agent Systems*, *22*(1), 183–199.

Reny, P. J. (2021). A simple sufficient condition for a unique and student-efficient stable matching in the college admissions problem. *Economic Theory Bulletin*, *9*, 7–9.

Roth, A. E. (1982). The economics of matching: Stability and incentives. *Mathematics of Operations Research*, *7*(4), 617–628.

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

Shen, W., Tang, P., & Deng, Y. (2018). Coalition manipulation of Gale-Shapley algorithm. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI '18)*, pp. 1210–1217.

Teo, C., Sethuraman, J., & Tan, W. (2001). Gale-Shapley stable marriage problem revisited: Strategic issues and applications. *Management Science*, *47*(9), 1252–1267.

Triin, L., Põder, K., & Veski, A. (2014). Matching practices for elementary schools – Estonia.. `https://www.matching-in-practice.eu/wp-content/uploads/2014/02/MiP_-Profile_No.18.pdf`.