Control and Bribery in Stable Marriage and Stable Roommates: A Complete Complexity Landscape

Jiehua Chen[1] and Ildikó Schlotter[2,3]

[1] TU Wien, Austria, jiehua.chen@ac.tuwien.ac.at
[2] ELTE Centre for Economic and Regional Studies, Hungary, schlotter.ildiko@krtk.elte.hu
[3] Budapest University of Technology and Economics, Hungary

Abstract. We study control and bribery problems for stable matchings: a central authority (the *controller*, resp., *briber*) may add agents, delete agents, delete acceptable pairs, swap two adjacent agents in some agent's preference list, or arbitrarily reorder some agent's preference list, in an instance of Stable Marriage or Stable Roommates. We extend previous work on control and bribery in stable matchings by Boehmer et al. [8]. We consider goals capturing individual and pair inclusion, stability, and uniqueness requirements: matching a designated agent (MA), matching a designated pair (MP), realizing a stable matching consistent with a given matching (MS), making a given matching the unique stable matching (USM), or guaranteeing that a stable (resp., perfect and stable) matching exists (∃SM/∃PSM). We provide a unified complexity map for all non-trivial action–goal combinations in both settings, consolidating known results and extending the study to the roommates model, where stable matchings need not exist.

1 Introduction

Stable Roommates deals with the division of human or non-human agents into pairs, where agents have preference lists describing their desires about whom they want to be matched with. More precisely, an instance of Stable Roommates consists of a set U of agents, with each agent $u \in U$ having a strict preference list over the subset of the remaining agents that u finds *acceptable*. The goal is to decide whether there is a *stable matching*, that is, a set M of pairwise disjoint pairs of agents for which *no* blocking pair exists. Herein, a *blocking pair* for M is a pair of distinct agents u and w who are not matched together by M such that

(i) u is either unmatched by M or prefers w to its partner in M and
(ii) w is either unmatched by M or prefers u to its partner in M.

Note that each Stable Roommates instance induces an *acceptability graph* where the vertices correspond to the agents and there is an edge between two vertices if both consider each other acceptable. A well-known special case of

Stable Roommates called Stable Marriage is the *bipartite* variant where agents are partitioned into two disjoint subsets such that each agent's preference list consists of agents of the opposite subset, and each agent can only be assigned a partner from that subset. The acceptability graph of an instance of Stable Marriage is then bipartite. Due to a celebrated result by Gale and Shapley [18], we know that in the marriage setting a stable matching always exists and can be found in time linear in the input size. However, for the roommates setting, a stable matching is not guaranteed to exist and, more than two decades after the seminal paper by Gale and Shapley [18], Irving [22] provided a polynomial-time algorithm to decide whether a given instance of Stable Roommates admits a stable matching.

Bartholdi III et al. [4, 3] introduced a framework for considering questions of manipulation and control in the context of elections, and their approach was adopted by Boehmer et al. [8] to investigate issues of manipulation, control, and bribery in the context of Stable Marriage. In their framework, an external agent (the *controller*, also called the *briber* when modifications affect preferences directly) aims to achieve a specific *goal* by applying as few *actions* as possible.

Example 1.1 ([13]). Suppose that Anna (a), Belle (b), Chris (c), and David (d) are attending a dance class, and need to form opposite-sex couples. Their preferences are as follows:

$$a\colon c \succ d, \qquad b\colon c \succ d, \qquad c\colon a \succ b, \qquad d\colon b \succ a.$$

The only stable matching in this instance is $\{\{a,c\},\{b,d\}\}$. However, Belle's friend, evil Eve, is among the teachers of the class, and her goal is to match Belle with her top-choice partner, Chris. She considers three options to achieve her goal: (1) declaring that Chris is too short for Anna and hence cannot dance with her, (2) stepping on Anna's toes with her high heels, thereby sending her off to ER, or (3) inviting her attractive friend, Frodo (f), whom Anna prefers to Chris. She decides on option (3), and obtains the following instance:

$$a\colon f \succ c \succ d, \quad b\colon c \succ f \succ d, \quad c\colon a \succ b, \quad d\colon b \succ a, \quad f\colon b \succ a.$$

Yet, Eve is not entirely satisfied, as now there are two stable matchings, namely $M_1 = \{\{a,f\},\{b,c\}\}$ and $M_2 = \{\{a,c\},\{b,f\}\}$. Hence, she declares that Frodo is too tall for Belle, ensuring that M_1 becomes the only stable matching.

Example 1.1 illustrates a few of the manipulative actions and goals examined in the literature on control and bribery for stable matchings. However, despite the apparent simplicity of the underlying space of action–goal–setting combinations, *previous work has so far only addressed disjoint fragments of it*: Boehmer et al. [8] consider five manipulative actions, but only three goals, and exclusively in the bipartite Stable Marriage setting; Bérczi et al. [5] consider two further (existence-based) goals, but only for the action of deleting agents and in Stable Roommates; and individual action–goal cases have appeared under different names in work on almost-stable [1, 7], maximum-cardinality stable [26, 28], and covering-constraint stable [25] matchings. The result is a *patchwork* of partial results, with no unified picture of which combinations are tractable, which are hard, and how the marriage and roommates settings relate.

In this paper, we close this gap. We *unify* these strands into a single coherent framework, considering *all* five manipulative actions of Boehmer et al. under *six* goals, in *both* the marriage and the non-bipartite roommates settings. Following the terminology of Boehmer et al., we organize the five actions into two complementary paradigms.

Control actions. In many matching markets, an organizer, platform, or regulator can change *participation* and *feasibility* without rewriting preferences: participants may enter or leave the market, and certain pairs may be ruled out (e.g., due to conflicts of interest or policy restrictions); moreover, unlike direct preference manipulation, such interventions can often be justified by external rules. We therefore consider the following control actions, showcased in Example 1.1:

- AddAg: adding agents (e.g., inviting Frodo),
- DelAg: deleting agents (e.g., removing Anna from the class), and
- DelAcc: deleting acceptability (e.g., declaring constraints on who cannot dance with whom).

As in classic control models in computational social choice, we minimize the number of actions, capturing intervention cost and market disruption.

Bribery actions. A complementary type of intervention models agents who can be *bribed*, i.e., convinced to misreport their preferences; it is captured by the following two actions of Boehmer et al. [8]:

- Swap: swapping two adjacent agents in some agent's preference list (modeling a briber who pays per "small" modification),
- Reorder: arbitrarily permuting some agent's preference list (modeling a briber who pays per fully "bought" agent).

As with control actions, we minimize the number of bribery operations.

Goals. The goals we study correspond to common individual and pair inclusion, stability, and uniqueness requirements. Formally, given an instance of Stable Roommates, we consider the goals:

- MA: a given agent is matched in some stable matching,
- MP: a given pair is contained in some stable matching,
- MS: there exists a stable matching consistent with a given matching M (i.e., using only pairs from M),
- USM: a given matching M is the unique stable matching,
- ∃SM: a stable matching exists, or
- ∃PSM: a perfect and stable matching exists.

Our Problems. For each action $\mathcal{A} \in \{\mathsf{AddAg}, \mathsf{DelAg}, \mathsf{DelAcc}, \mathsf{Swap}, \mathsf{Reorder}\}$ and each goal $\mathcal{G} \in \{\mathsf{MA}, \mathsf{MP}, \mathsf{MS}, \mathsf{USM}, \exists\mathsf{SM}, \exists\mathsf{PSM}\}$, the Control for Stable Roommates-$\mathcal{A}$-$\mathcal{G}$ (abbreviated as CSR-$\mathcal{A}$-$\mathcal{G}$) problem asks for a minimum number of actions of type $\mathcal{A}$ necessary to achieve the goal $\mathcal{G}$ in a given Stable Roommates instance. When restricted to the bipartite marriage setting, we call these problems Control for Stable Marriage-$\mathcal{A}$-$\mathcal{G}$ (abbreviated as CSM-$\mathcal{A}$-$\mathcal{G}$).

Table 1: Complexity results for Control for Stable Marriage (or Roommates)-$\mathcal{A}$-$\mathcal{G}$ problems. The problems CSM-$\mathcal{A}$-∃SM, marked "—", are trivial, as a stable matching always exists in an instance of SM. "NPc" stands for NP-complete. Hardness results for CSR that carry the mark of a result shown for CSM follow from that result, since Stable Marriage is a special case of Stable Roommates (membership in NP is straightforward in each such case). Additionally, we use the following notation: [♦] and [♦] mark results by Boehmer et al. [8] and easy consequences thereof, respectively; [†] marks results by Tan [26, 28]; [♡] marks results by Mnich and Schlotter [25]; [⋆] marks results by Abraham et al. [1]; [♠] marks results by Biró et al. [7]; [∗∗] marks results by Chen, Skowron, and Sorge [15]. Previously known results are typeset in gray; results new to this paper are typeset in **bold** black.

	AddAg		DelAg		DelAcc		Reorder		Swap	
	CSM	CSR	CSM	CSR	CSM	CSR	CSM	CSR	CSM	CSR
MA	**NPc** [T3.2]	**NPc** [C3.3]	in P♦ [T3.1]	**in P** [T3.1]	NPc♡	NPc♡	**NPc** [T4.1]	**NPc** [T4.1]	**NPc** [P4.2]	**NPc** [P4.2]
MP	NPc♦	NPc♦	in P♦	**in P** [T3.1]	NPc♦	NPc♦	NPc♦	NPc♦	NPc♦	NPc♦
MS	in P♦	**NPc** [T3.4]	NPc♦	NPc♦	in P♦	in P♦ [P3.5]	in P♦	**NPc** [P4.5]	in P♦	**NPc** [T4.4]
USM	NPc♦	NPc♦	NPc♦	NPc♦	in P♦	**NPc** [T3.7]	NPc♦	NPc♦	NPc♦	NPc♦
∃SM	—	**NPc** [T3.6]	—	in P†	—	NPc⋆	—	**NPc** [T4.6]	—	**NPc** [T4.6]
∃PSM	**NPc** [T3.2]	**NPc** [T3.6]	in P†	in P†	NPc♠	NPc♠	NPc♦	NPc♦	NPc∗∗	NPc∗∗

Our contributions. Table 1 summarizes the computational complexity of all non-trivial action–goal–setting combinations. The majority of these results are not new: they either appear explicitly in the prior literature on stable-matching control/manipulation, or they follow via transfer arguments (in particular, from CSM to CSR or by standard reductions from known "almost stable" and deletion problems). Our main technical contributions are (i) introducing and studying the control goal MA (ensuring that a designated agent is matched in some stable outcome), which was not part of the previously studied stable-matching control goal sets, and (ii) settling the remaining cases not implied by existing results or immediate transfers (highlighted in Table 1), yielding a complete and coherent landscape for these actions and goals in both CSM and CSR. For the tractable new cases we provide polynomial-time algorithms, and for the intractable ones we give matching NP-hardness proofs.

Outline. Section 2 introduces notation and formalizes our control problems. Sections 3 and 4 contain our results for control actions (AddAg, DelAg, and

DelAcc), and for bribery actions (Reorder and Swap), respectively. We conclude in Section 5. Proofs of results marked by ($\star$) are deferred to the full version [14].

Related work. Control was introduced in computational social choice for elections by Bartholdi III et al. [3] and brought to stable matchings by Boehmer et al. [8], who study multiple actions/goals (including unique stability) for Stable Marriage. A broader unifying perspective is provided by the IJCAI 2025 survey of Chen et al. [13], which reviews control across COMSOC (including voting, fair allocation, cooperative games, and matching under preferences) and helps situate stable-matching control among related structural-intervention models.

Several earlier lines of work coincide with special cases of our action/goal combinations under different motivations. Tan [26, 28] studies maximum stable matchings for Stable Roommates, which corresponds to deleting as few agents as possible so that the remaining instance admits a perfect and stable matching (i.e., CSR-DelAg-∃PSM). Problems of the form CSR-DelAcc-$\mathcal{G}$ are closely related to *almost stable* matchings: Deleting k acceptable pairs can be viewed as removing up to k potential blocking pairs. Abraham et al. [1] prove strong inapproximability for minimizing blocking pairs in Stable Roommates, and Biró et al. [7] show even stronger inapproximability for maximizing size while minimizing blocking pairs; their hardness already applies to the marriage setting with a perfect target, implying intractability for CSM-DelAcc-∃PSM. Mnich and Schlotter [25] study covering distinguished agents subject to a bound on blocking pairs; in particular, their hardness for covering even a single distinguished agent implies hardness for CSM-DelAcc-MA. Further intractability and parameterized results include Chen et al. [12] (parameterized analysis for DelAcc variants), Gupta and Jain [20] (weighted/destructive variants), and Kamiyama [23] (agent deletion to achieve super-stability with ties).

Finally, control by adding/deleting agents has recently been extended to coalition formation: Chen et al. [11] initiate the study of control in hedonic games under several stability notions and goals. This underlines that participation-based interventions form a common algorithmic theme across COMSOC models; our results give a complete picture for the classical stable matching setting.

2 Preliminaries

For a positive integer t, we write $[t] \coloneqq \{1, \ldots, t\}$. In the following, we first introduce the basic concepts of the standard model for Stable Roommates, and then define the control problems we study.

2.1 Model for Stable Roommates

Formally, an instance I of Stable Roommates consists of a set U of agents and a collection $(\succ_u)_{u \in U}$ of preferences where each preference list $\succ_u$ is a strict linear order over the set $A(u) \subseteq U \setminus \{u\}$ of agents that u finds *acceptable*. We assume that the acceptability relation is symmetric, meaning that x finds y acceptable if and only if y finds x acceptable. For the acceptability relation of each instance I, there exists an *associated undirected graph* $G = (U, E)$ over the set of agents such that two agents are connected by an edge of E if and only if they find each other acceptable. We use $E(I)$ to denote the set of all acceptable pairs in I.

A *matching* in I is a subset $M \subseteq E$ of pairwise disjoint edges. For a pair $\{u, v\} \in M$, we will use the notation $M(u) = v$ and $M(v) = u$ to denote the partner of u and v, respectively. Consequently, we say that M *covers* u and v. It is *perfect* for I if every agent in I is covered. We write $M(u) = \emptyset$ to indicate that u is *unmatched* by M, that is, M does not cover agent u. A *blocking pair* for some matching M in I is a pair $\{u, v\} \in E$ of agents such that

(i) either $M(u) = \emptyset$ or $v \succ_u M(u)$, and similarly,
(ii) either $M(v) = \emptyset$ or $u \succ_v M(v)$.

A matching is *stable* if it admits no blocking pair.

Given an instance $I = (U, (\succ_u)_{u\in U})$ of Stable Roommates and a subset W of agents, $I - W$ is the instance obtained by *deleting the agents* in W from I, that is, we let $I - W := (U \setminus W, (\succ'_u)_{u\in U\setminus W})$ where $\succ'_u$ is the restriction of $\succ_u$ to $U \setminus W$. Similarly, given a subset $F \subseteq E$ of agent pairs, the instance obtained from I by *deleting the acceptability* for the pairs in F is $I - F := (U, (\succ'_u)_{u\in U})$, where for each agent $u \in U$, the set of acceptable agents becomes $A'(u) := A(u) \setminus \{v : \{u, v\} \in F\}$ and thus $\succ'_u$ is the restriction of $\succ_u$ to $A'(u)$.

Stable Marriage. The Stable Marriage model is the bipartite special case of Stable Roommates: The agent set is partitioned into two sides, and acceptability edges only go across the partition. Following tradition, we refer to agents on the two sides as *men* and *women*. Each agent ranks its acceptable partners on the opposite side in strict order, and a matching pairs men and women. Stability is defined identically (no blocking pair of mutually preferred partners), but unlike in Stable Roommates, a stable matching always exists in Stable Marriage and can be found by the Gale–Shapley algorithm [18].

2.2 Problem definitions

We now formally define problem CSR-$\mathcal{A}$-$\mathcal{G}$ for each action $\mathcal{A} \in \{\mathsf{AddAg}, \mathsf{DelAg}, \mathsf{DelAcc}, \mathsf{Swap}, \mathsf{Reorder}\}$ and each goal $\mathcal{G} \in \{\mathsf{MA}, \mathsf{MP}, \mathsf{MS}, \mathsf{USM}, \exists\mathsf{SM}, \exists\mathsf{PSM}\}$.

Adding agents. We first focus on the control action of agent addition AddAg. For this control action, the input not only contains the *original* set U of agents but also a set U' of *addable* agents, and the preferences of the agents are given for the whole market over $U \cup U'$. For the control goals ∃SM and ∃PSM, the definitions are as follows:

CSR-AddAg-∃SM (resp. CSR-AddAg-∃PSM)
Input: An instance $I = (U \cup U', (\succ_u)_{u\in U\cup U'})$ of Stable Roommates, where U denotes the set of *original* agents and U' the set of *addable* agents, and an integer *budget* ℓ.
Question: Is there a subset $W \subseteq U'$ with $|W| \le \ell$ such that $I - (U' \setminus W)$ admits a stable (resp., *perfect* and stable) matching?

For the control goals MA and MP, we require an additional input:

CSR-AddAg-MA (resp. CSR-AddAg-MP)
Input: An instance $I = (U \cup U', (\succ_u)_{u \in U \cup U'})$ of Stable Roommates, an integer *budget* ℓ, and an agent $u^* \in U$ (resp., a pair $e^* \in E(I)$ of agents).
Question: Is there a subset $W \subseteq U'$ with $|W| \leq \ell$ such that $I - (U' \setminus W)$ admits a stable matching that covers u^* (resp., contains e^*)?

Finally, for the goals MS and USM, following Boehmer et al. [8], the additional input is a *perfect* matching, which covers not only the original agents but also the addable agents. The reason for this is that the control action AddAg may alter the set of matched agents in an instance.

CSR-AddAg-MS (resp. CSR-AddAg-USM)
Input: An instance $I = (U \cup U', (\succ_u)_{u \in U \cup U'})$ of Stable Roommates, an integer *budget* ℓ, and a perfect matching M of I.
Question: Is there a subset $W \subseteq U'$ with $|W| \leq \ell$ such that $I - (U' \setminus W)$ admits a stable matching M' with $M' \subseteq M$ (resp., a *unique* stable matching M' with $M' \subseteq M$)?

Deleting agents. The control problems regarding the control action of deleting agents are defined analogously, so we omit their definitions here, except for the goal MS, where we again assume the input matching to be perfect:

CSR-DelAg-MS
Input: An instance $I = (U, (\succ_u)_{u \in U})$ of Stable Roommates, an integer *budget* ℓ, and a perfect matching M of I.
Question: Is there a subset $W \subseteq U$ of at most ℓ agents such that $I - W$ admits a stable matching M' with $M' \subseteq M$?

Deleting acceptability. The control problems CSR-DelAcc-$\mathcal{G}$ are defined analogously: instead of a set of at most ℓ agents $W \subseteq U$, we ask for a set of at most ℓ edges $F \subseteq E(I)$ so that the instance $I - F$ satisfies our goals. For brevity, we omit these problem definitions. We remark that, contrasting the previous two control actions, the input matching for CSR-DelAcc-MS is simply a matching, because deleting acceptability does not change the set of agents:

CSR-DelAcc-MS
Input: An instance $I = (U, (\succ_u)_{u \in U})$ of Stable Roommates, an integer *budget* ℓ, and a matching M in I.
Question: Is there a subset $F \subseteq E(I)$ with $|F| \leq \ell$ such that $I - F$ admits a stable matching M' with $M' \subseteq M$?

Swapping agents in a preference list. We further consider the manipulative action Swap, introduced by Boehmer et al. [8] as a *bribery* action: in contrast to the three previous actions, Swap does not change the set of agents nor the acceptability relation, but instead modifies the preference orders themselves. Formally, given an instance $I = (U, (\succ_u)_{u \in U})$ of Stable Roommates, a *swap operation* is a triple $\sigma = (u, v, v')$ where $u \in U$ and $v, v' \in A(u)$ appear consecutively in $\succ_u$, that is, $v \succ_u v'$ and no $x \in A(u)$ satisfies $v \succ_u x \succ_u v'$. Applying σ to I yields the instance $I_\sigma = (U, (\succ'_u)_{u \in U})$ in which $\succ'_u$ is identical to $\succ_u$ except that the relative order of v and v' is reversed, while $\succ'_x = \succ_x$ holds for each $x \in U \setminus \{u\}$. Modeling a briber whose cost grows with the distance an agent must move in another agent's preferences, Swap is the most fine-grained of the bribery actions. Given a sequence S of swap operations, we write $I \oplus S$ for the instance obtained by applying the operations of S to I one after the other (each subsequent operation is interpreted in the already partially modified instance).

Reordering preference lists. The second bribery action we adopt from Boehmer et al. [8] is Reorder, which models a briber capable of fully "buying" an agent: a single operation may replace an agent's preference list with an arbitrary strict order over the same acceptable set. Formally, a *reorder operation* is a pair $\rho = (u, \succ^\circ_u)$ where $u \in U$ and $\succ^\circ_u$ is a strict linear order over $A(u)$. Applying ρ to I replaces $\succ_u$ with $\succ^\circ_u$ and leaves every other agent's preference list intact. Given a set R of reorder operations, with at most one per agent, we again write $I \oplus R$ for the resulting instance.

Restriction for the goals MA *and* MP*.* As already observed by Boehmer et al. [8, Section 2.4] for the goal corresponding to MP, allowing the briber to act on the distinguished agents themselves trivializes the problem (e.g., a constant number of reorders on u^* and v^* would already make them each other's top choice). They therefore explicitly forbid reordering the preference list of any agent in the distinguished pair, and we adopt the same restriction for the goal MP. For the goal MA, however, protecting only the distinguished agent u^* itself is not enough: a single reorder operation placing u^* at the top of the preference list of an arbitrary agent $w \in A(u^*)$ already forces u^* to be matched in every stable matching, as otherwise $\{u^*, w\}$ would form a blocking pair. To prevent the briber from promoting the distinguished agent indirectly in this way, we additionally forbid operations on the agents that u^* finds acceptable. Formally, given an instance I, a swap or reorder operation acting on an agent u is *admissible* if

- for the goal MA with distinguished agent $u^* \in U$: $u \neq u^*$ and $u \notin A(u^*)$;
- for the goal MP with distinguished pair $e^* = \{u^*, v^*\}$: $u \notin \{u^*, v^*\}$.

For the goals MS, $\exists$SM, $\exists$PSM, and USM, every swap and reorder operation is admissible.

For each bribery action $\mathcal{A} \in \{\text{Swap}, \text{Reorder}\}$ and each goal $\mathcal{G} \in \{\text{MA}, \text{MP}, \text{MS}, \exists\text{SM}, \exists\text{PSM}, \text{USM}\}$, the problem CSR-$\mathcal{A}$-$\mathcal{G}$ asks whether the goal can be achieved by at most ℓ admissible operations of type $\mathcal{A}$. We illustrate the formal definitions with two representative cases.

CSR-Swap-MA (resp. CSR-Swap-MP)
Input: An instance $I = (U, (\succ_u)_{u \in U})$ of Stable Roommates, an integer *budget* ℓ, and an agent $u^* \in U$ (resp., a pair $e^* \in E(I)$).
Question: Is there a sequence S of at most ℓ admissible swap operations on I such that $I \oplus S$ admits a stable matching that covers u^* (resp., contains e^*)?

CSR-Reorder-∃SM (resp. CSR-Reorder-∃PSM)
Input: An instance $I = (U, (\succ_u)_{u \in U})$ of Stable Roommates and an integer *budget* ℓ.
Question: Is there a set R of at most ℓ reorder operations on I such that $I \oplus R$ admits a stable (resp., *perfect* and stable) matching?

We omit the analogous definitions for the other action–goal combinations. The marriage variants of our problems are defined analogously, with the input containing an instance of Stable Marriage instead of Stable Roommates.

We conclude this section with the following simple observation.

Observation 2.1 **($\star$)** *For every control action $\mathcal{A} \in \{$AddAg, DelAg, DelAcc$\}$, there is a polynomial-time Turing reduction from CSR-$\mathcal{A}$-MA to CSR-$\mathcal{A}$-MP, and from CSM-$\mathcal{A}$-MA to CSM-$\mathcal{A}$-MP.*

We remark that Observation 2.1 does not extend directly to the bribery actions Swap and Reorder, since the admissibility restrictions of the two problems differ: an operation that is admissible for the goal MP with distinguished pair $\{u^*, w^*\}$ may act on an agent in $A(u^*) \setminus \{w^*\}$ and is then inadmissible for the goal MA with distinguished agent u^*.

3 Control actions

We first focus on control by adding or deleting agents, or deleting acceptability. Within the section, we organize our results based on the goal we aim to achieve.

Matching an agent or a pair of agents. We start with our main algorithmic result which shows that even in an instance of Stable Roommates, we can efficiently decide whether, by deleting at most ℓ agents, we can obtain an instance in which some stable matching pairs two designated agents together. Naturally, this also implies an algorithm for ensuring via agent deletions that a given agent is matched in some stable matching—it suffices to try all its possible partners.

Boehmer et al. [8] showed that, in the marriage setting, it is polynomial-time solvable to make a given pair part of some stable matching. From this, the polynomial-time solvability of CSM-DelAg-MA follows using Observation 2.1. We now extend the approach by Boehmer et al. [8] to the roommates setting. The crucial idea is to use the concept of stable partitions by Tan [28] and a result by Tan and Hsu [29] regarding deleting an agent; see also [24, Sections 4.3.2 and 4.3.4] for the necessary background on stable partitions.

Theorem 3.1. *CSR-DelAg-MP and CSR-DelAg-MA are in* P*.*

Proof. We present a polynomial-time algorithm for the CSR-DelAg-MP problem. By Observation 2.1, this implies the polynomial-time solvability of CSR-DelAg-MA as well. As already mentioned, we will use the structural properties of stable partitions and ideas from the algorithm given by Boehmer et al. [8] for the bipartite version of our problem, CSM-DelAg-MP.

ALGORITHM 1: Algorithm for solving CSR-DelAg-MP

Input: An instance I of Stable Roommates, a budget ℓ, and a pair of agents $\{a, b\}$

Output: “yes” if there exists a set S of at most ℓ agents such that $I - S$ admits a stable matching that contains $\{a, b\}$; otherwise “no”

1 Compute $A^\star$, $B^\star$, and F as defined in (1)
2 Compute instance $I^\star \coloneqq I - F$
3 Compute a stable partition Π for $I^\star$
4 Let U_0 denote the set of singleton parties in Π consisting of an agent in $A^\star \cup B^\star$
5 Let r denote the number of odd parties of size at least 3 in Π
6 Return “yes” if $r + |U_0| \leq \ell$; otherwise, return “no.”

Let $\mathcal{J} = (I, \ell, \{a, b\})$ be our input instance where I is an instance of Stable Roommates, ℓ is our budget, i.e., the number of agents we are allowed to delete, and $\{a, b\}$ is the (acceptable) pair in I that we aim to include in the desired matching. We say that a set S of agents is a *solution* for $\mathcal{J}$ if removing S from I yields an instance, denoted by $I - S$, where some stable matching contains $\{a, b\}$.

Let $A^\star$ and $B^\star$ be the sets of agents which are preferred by a to b, or by b to a, respectively (note that $A^\star$ and $B^\star$ need not be disjoint). Furthermore, let the edge set F contain those pairs $\{x, y\}$, where

(i) $x \in A^\star$, and $y = a$ or x prefers a to y, or
(ii) $x \in B^\star$, and $y = b$ or x prefers b to y.

Formally, let

$$\begin{aligned} A^\star &= \{x : x \succ_a b\}, \qquad & F &= \{\{x, y\} : x \in A^\star, (y = a \text{ or } a \succ_x y)\} \\ B^\star &= \{x : x \succ_b a\}, & &\cup \{\{x, y\} : x \in B^\star, (y = b \text{ or } b \succ_x y)\}. \end{aligned} \tag{1}$$

Note that no pair in F can be included in a stable matching containing $\{a, b\}$, because all such pairs either match a or b (but not via $\{a, b\}$), or result in a blocking pair formed by one of the agents in the pair and an agent in $\{a, b\}$.

Let $I^\star$ denote the instance obtained from I by deleting all pairs in F from the set of acceptable pairs, i.e., $I^\star \coloneqq I - F$.

Our algorithm is depicted in Algorithm 1. Note that line 3 can be performed in linear time using Tan’s algorithm [27] for computing a stable partition. Hence, the algorithm runs in polynomial time. It remains to show its correctness. We start with the following claim.

Claim 3.1 *A set S of agents with $S \cap \{a, b\} = \emptyset$ is a solution for $\mathcal{J}$ if and only if $I^\star - S$ admits a stable matching that matches every agent in $(A^\star \cup B^\star) \setminus S$.*

Proof. Assume that S is a solution for $\mathcal{J}$. Then there exists a stable matching M in $I - S$ containing $\{a, b\}$. First, we show that $M \cap F = \emptyset$. Suppose for the sake of contradiction that $\{x, y\} \in M \cap F$. Then $x \in A^\star \cup B^\star$; by symmetry we may assume $x \in A^\star$. By the definition of F, we infer that either $y = a$ or x prefers a to y. The former is not possible, because $M(a) = b \notin A^\star$; the latter is not possible either, because it would mean that $\{x, a\}$ blocks M: x prefers a to $y = M(x)$

and a prefers x to $M(a) = b$ by the definition of $A^\star$. This contradiction proves that M is a matching disjoint from F, i.e., it is a matching in $I^\star - S$.

It remains to show that M matches every agent in the set $(A^\star \cup B^\star) \setminus S$. Suppose for the sake of contradiction that $x \in (A^\star \cup B^\star) \setminus S$ is unmatched by M. By symmetry, we may assume $x \in A^\star$. However, then the pair $\{x, a\}$ is a blocking pair, a contradiction. Thus, M has the desired property.

For the other direction, assume that for some agent set S with $S \cap \{a, b\} = \emptyset$, the instance $I^\star - S$ admits a stable matching M that matches every agent in the set $(A^\star \cup B^\star) \setminus S$. In this case, M matches a to b because they are each other's best choice in $I^\star$ due to the deletion of the acceptability of all pairs in F. Moreover, M is stable in $I - S$, because it is stable in $I^\star - S$ and, since all agents in $(A^\star \cup B^\star) \setminus S$ are matched in $I^\star - S$, no pair in F blocks M. $\lhd$

To show the correctness of our algorithm, assume first that it returns "yes". Let S contain an arbitrarily chosen agent from each odd party of size at least 3 in Π and all singleton parties consisting of an agent in $A^\star \cup B^\star$. Note that $a, b \notin S$: since a and b are each other's top choice in $I^\star$, they form a party of size two in Π, so neither belongs to an odd party. Then deleting S from $I^\star$ yields an instance I' whose stable partition Π' has no odd parties of size at least 3 and whose singleton parties are all disjoint from $A^\star \cup B^\star$, due to well-known properties of stable partitions [28] (see also [24, Proposition 4.11]). This means that in all stable matchings for $I^\star - S$, the set of unmatched agents (i.e., those agents that form singletons in Π') is disjoint from $A^\star \cup B^\star$. By Claim 3.1, S is therefore a solution, and since $|S| = r + |U_0| \leq \ell$ by line 6 of Algorithm 1, we obtain that $\mathcal{J}$ is a "yes"-instance.

It remains to prove that whenever S is a solution of size at most ℓ, then the algorithm returns "yes". Clearly, $a, b \notin S$, as some stable matching of $I - S$ must contain the pair $\{a, b\}$. Since S is a solution, by Claim 3.1 we know that $I^\star - S$ admits a stable matching that matches all agents in $(A^\star \cup B^\star) \setminus S$. This means that the stable partition Π_S of $I^\star - S$ contains no odd parties of size at least 3, and contains no singleton odd parties formed by some agent in $(A^\star \cup B^\star) \setminus S$; let us call such odd parties *forbidden*. However, it is known that deleting an agent from an instance of Stable Roommates results in an instance whose stable partitions contain all odd parties of the original instance except, possibly, for one (see [29, Corollary 3.9]). Since Π has exactly $|U_0| + r$ forbidden odd parties, deleting fewer than $r + |U_0|$ agents from $I^\star$ will result in an instance that has at least one forbidden odd party. This implies $\ell \geq |S| \geq r + |U_0|$, so the algorithm returns "yes" on line 6. $\square$

Let us turn our focus to control by adding agents. Our next result shows that making an agent matched (MA) in a stable matching is no easier than making a pair matched (MP). The reduction is from the NP-complete Clique problem [19] and is inspired by the one given by Boehmer et al. [8] for the MP case, and happens to prove hardness also for CSM-AddAg-∃PSM.

Theorem 3.2 ($\star$). *CSM-AddAg-MA and CSM-AddAg-∃PSM are NP-complete.*

Note that CSR-AddAg-MA is in NP, because even in an instance of Stable Roommates, all stable matchings cover the same set of agents [21, Theorem 4.5.2]. This leads to the following consequence of Theorem 3.2.

Corollary 3.3. *CSR*-AddAg-MA *is* NP*-complete.*

Stabilizing a matching. Having settled the complexity of using control actions to achieve the goals MA and MP, we next turn to the goal MS of stabilizing a prescribed matching. For Stable Marriage, AddAg-MS is solvable in polynomial time by Boehmer et al. [8]. The roommates setting is more brittle: even when we are given a perfect "plan" matching and are only allowed to keep pairs from that plan, deciding whether agent additions suffice becomes intractable.

Theorem 3.4 (⋆). *CSR*-AddAg-MS *is* NP*-complete.*

Boehmer et al. [8, Observation 1] observed that for the marriage case, the number of blocking pairs of a given matching M lower-bounds the number of acceptable pairs to delete to make M stable. Since this observation applies to the roommates case as well, the following holds.

Proposition 3.5 (⋆). *CSR*-DelAcc-MS *can be solved in linear time.*

Ensuring the existence of a (perfect) stable matching. We now move to the existence-oriented goals for AddAg in CSR, where no reference matching is given and the controller only aims to ensure that a stable (or perfect and stable) matching exists after adding agents. Even under this additional freedom, adding agents remains computationally hard in the roommates setting.

Theorem 3.6 (⋆). *CSR*-AddAg-∃SM *and CSR*-AddAg-∃PSM *are* NP*-complete.*

Next, we prove that the polynomial-time algorithm by Boehmer et al. [8] for CSM-DelAcc-USM does not generalize to the roommates setting; we present a reduction from the NP-complete Back and Forth Paths problem [17].

Theorem 3.7 (⋆). *CSR*-DelAcc-USM *is* NP*-complete, even on instances whose underlying graph can be made bipartite by deleting a single edge.*

Proof sketch. Whether an instance of Stable Roommates admits a unique stable matching can be decided in polynomial time by standard methods [21]. To show NP-hardness, we reduce from the following variant of the NP-complete problem Back and Forth Paths [17]: given a directed graph $D = (V, E)$ and two arcs (s, s') and (t, t') in D, decide if there exists a cycle through (s, s') and (t, t') in D. We may assume that every vertex in D has indegree at least 2, and that $(s, v) \in E$ for each $v \in V \setminus \{s, s'\}$; see the full version [14] for details.

For some $v \in V$, let d_v denote the indegree of v. We let $V' = V \setminus \{s', t'\}$. For each vertex v, we fix an ordering of the incoming arcs of v, and for each $i \in [\mathsf{d}_v]$ we denote by $\delta(v, i)$ the tail of the i-th arc in this ordering. For simplicity, we let (s,s') and (t,t') be the first arcs among the incoming edges of s' and of t', respectively, so $\delta(s',1)=s$ and $\delta(t',1)=t$.

Construction. We construct an instance of Stable Roommates as follows. For each vertex $v \in V$, we create two sets of agents, $A_v = \{a_v^i \colon i \in [\mathsf{d}_v]\}$ and $B_v = \{b_v^i \colon i \in [\mathsf{d}_v]\}$. The preferences of these agents are as follows; throughout the proof, we treat the superscripts of agents in $A_v \cup B_v$ modulo d_v:

$$
\begin{array}{lll}
a_v^1\colon & \{b_u^j : \delta(u,j)=v\} \succ b_v^{\mathsf{d}_v} \succ b_v^1 & \text{for all } v \in V, \\
a_v^i\colon & b_v^{i-1} \succ b_v^i & \text{for all } v \in V,\ 2 \le i \le \mathsf{d}_v, \\
b_v^i\colon & a_v^i \succ a_v^{i+1} \succ a_u^1 & \text{for all } v \in V',\ i \in [\mathsf{d}_v],\ \delta(v,i)=u, \\
b_{s'}^1\colon & a_{s'}^1 \succ a_{s'}^2 \succ b_{t'}^1 \succ a_s^1, & \\
b_{t'}^1\colon & a_{t'}^1 \succ a_{t'}^2 \succ b_{s'}^1 \succ a_t^1, & \\
b_x^i\colon & a_x^i \succ a_x^{i+1} \succ a_u^1 & \text{for } x \in \{s',t'\},\ 1 < i \le \mathsf{d}_x,\ \delta(x,i)=u.
\end{array}
$$

We set the target matching to $M = \{\{a_v^i, b_v^i\} : v \in V, i \in [\mathsf{d}_v]\}$ and the budget to $\ell = |V|$. Note that the instance I of Stable Roommates is almost bipartite: deleting the edge $\{b_{s'}^1, b_{t'}^1\}$ yields an instance of Stable Marriage.

Correctness. First assume that F is a set of at most ℓ edges in $E(I)$ such that M is the unique stable matching in $I' = I - F$. For each $v \in V$, let C_v denote the unique M-alternating cycle over agent set $A_v \cup B_v$. It is not hard to see that $M_v = M \Delta C_v$ (where Δ stands for symmetric difference) is a stable matching in I. Hence, M_v is also a stable matching in I' unless F contains an edge of $C_v \setminus M$. Due to $|F| \le |V|$, we know that F contains exactly one edge from each $C_v \setminus M$; let $\{b_v^{\sigma(v)}, a_v^{\sigma(v)+1}\}$ denote this edge for some $\sigma(v) \in [\mathsf{d}_v]$.

For each $v \in V$, consider the arc (u,v) in D such that $\delta(v,\sigma(v)) = u$; we will use the notation $\alpha(v) = u$. Define $E^\star = \{(\alpha(v), v) \colon v \in V\} \subseteq E$. We are going to show that $E^\star$ contains a cycle traversing both (s,s') and (t,t'). Clearly, $E^\star$ contains some directed cycle C, since each vertex has indegree exactly 1 in $E^\star$. We define an M-alternating cycle $\varphi(C)$ corresponding to C as the union of the paths $P_{(u,v)} = (a_v^1, b_v^1, \dots, b_v^{\sigma(v)}, a_u^1)$ for $u = \alpha(v)$, taken over each $v \in V(C)$. It can be shown that if C contains at most one arc from $\{(s,s'),(t,t')\}$, then the matching $M' = M \Delta \varphi(C)$ is stable in I', because each agent a_v^i for which $M'(a_v^i) \neq M(a_v^i)$ prefers M' to M, and each agent b_v^i for which $M'(b_v^i) \neq M(b_v^i)$ is matched by M' to its second best choice in I', except possibly for either $b_{s'}^1$ or $b_{t'}^1$; thus, the only possible blocking edge for M' is $e^\star = \{b_{s'}^1, b_{t'}^1\}$, but by our assumption, at least one endpoint of $e^\star$ prefers its partner both in M and in M' to the other endpoint of $e^\star$. Thus, M' is stable in I', a contradiction. This implies that C must be a cycle in D containing both (s,s') and (t,t').

Assume now that C is a cycle in D containing both (s,s') and (t,t'). Let $\alpha(v)$ be the vertex preceding v on C if $v \in V(C)$, and let $\alpha(v) = s$ otherwise; then $\{(\alpha(v), v) \colon v \in V\} \subseteq E$. We define $F = \{\{b_v^i, a_v^{i+1}\} \colon v \in V,\ \alpha(v) = \delta(v,i)\}$ and claim that M is the unique stable matching in $I' = I - F$. Since stability of M is clear, it remains to prove the uniqueness in I'.

Assume for contradiction that I' admits a stable matching other than M. By the work of Cheng and Rosenbaum [16], we know that there exists a stable matching M' in I'—namely, a stable matching in I' that is adjacent to M in the median graph of the instance I'—for which $M \mathbin{\Delta} M'$ is a single cycle C_Δ. Since deleting $\{b^1_{s'}, b^1_{t'}\}$ yields a bipartite instance, C_Δ cannot contain $\{b^1_{s'}, b^1_{t'}\}$.

We say that some $v \in V$ is *important*, if C_Δ contains a vertex in $A_v \cup B_v$. Note that $V(C_\Delta) \subseteq A_v \cup B_v$ is not possible, because $A_v \cup B_v$ induces a path in I'. Thus, if v is important, then C_Δ needs to contain at least two edges having exactly one endpoint in $A_v \cup B_v$, and at most one such edge can be incident to a^1_v. Thus, at least one edge on C_Δ connects some b^i_v to a^1_u for some $i \in [\mathsf{d}_v]$ and $u = \delta(v, i)$. However, then $\{b^i_v, a^{i+1}_v\}$ must be in F, as otherwise it blocks M'. This means that $\{b^i_v, a^{i+1}_v\} \in F$ and, therefore, $\alpha(v) = u = \delta(v, i)$.

Moreover, using the structure of our construction and the fact that C_Δ is M-alternating, it can be shown that the edges of C_Δ incident to vertices of $A_v \cup B_v$ are exactly the edges of $P_{(u,v)}$ for $\alpha(v) = u$ and the edge $\{a^1_v, b^j_z\} \in P_{(v,z)}$ where $\alpha(z) = v = \delta(z, j)$. Consequently, C_Δ must be the union of all paths $P_{(\alpha(v),v)}$ over all important vertices v. It follows that the arcs $(\alpha(v), v) \in E$ for each important vertex v form a single cycle in D which, thus, must be the cycle C; in fact, using the notation $\varphi(\cdot)$ for the first part of the proof, we have $\varphi(C) = C_\Delta$. In particular, $\{b^1_{s'}, a^1_s\}$ and $\{b^1_{t'}, a^1_t\}$ are both on $C_\Delta \cap M'$. However, then $\{b^1_{s'}, b^1_{t'}\}$ forms a blocking pair for M', a contradiction to the stability of M'. This proves that M is indeed the unique stable matching in $I' = I - F$, proving the correctness of our reduction. $\lhd$

4 Bribery actions

In this section, we focus on the bribery actions that manipulate agents' preferences via the Swap or the Reorder actions. Within the section, we again organize our results based on the goal we aim to achieve.

Matching an agent. Boehmer et al. [8] showed that, already in the marriage setting, it is NP-complete to decide whether reordering at most ℓ preference lists yields a stable matching that contains a given pair (MP). Their hardness for MP does not immediately imply hardness for MA: a straightforward reduction from MP to MA would require one of the agents of the distinguished pair to have only the other as its acceptable partner, since this is the only way to make "the distinguished agent is matched in some stable matching" equivalent to "the distinguished pair is contained in some stable matching". The construction of Boehmer et al. does not enjoy this property. We therefore provide a direct reduction for MA in the marriage setting. Note that in our construction the distinguished agent a^* has a unique acceptable partner b^*, so a stable matching covers a^* if and only if it contains the pair $\{a^*, b^*\}$; moreover, the admissible operations for the goal MA (which must avoid a^* and $A(a^*) = \{b^*\}$) coincide with those for the goal MP with distinguished pair $\{a^*, b^*\}$. Hence, our reduction also serves as a valid reduction for MP, and thus subsumes the hardness of CSM-Reorder-MP proved by Boehmer et al.

Theorem 4.1 (⋆). *CSM*-Reorder-MA *and CSR*-Reorder-MA *are* NP*-complete.*

The same construction with a larger budget yields hardness for Swap.

Proposition 4.2 (⋆). *CSM*-Swap-MA *and CSR*-Swap-MA *are* NP*-complete.*

Remark 4.3. The NP-hardness for the swap action in the marriage case can alternatively be derived from Theorem 3.2 via a generic reduction from AddAg to Swap due to Boehmer et al. [8]: for each addable agent p in the source instance, introduce two dummy agents p_d and q_d with preferences $p_d\colon p \succ q_d$ and $q_d\colon p_d$, and prepend p_d to the head of p's preference list; in addition, pad every other preference list with sufficiently many further dummies between consecutive entries so that no swap on any other list fits within the budget. A single swap on p_d's list (from $p \succ q_d$ to $q_d \succ p$) causes $\{p_d, q_d\}$ to be matched, releasing p to participate in the rest of the matching—corresponding exactly to adding p in the source instance, with the budgets coinciding.

Stabilizing a matching. Next, we consider the goal MS. Boehmer et al. [8] show that both bribery actions Swap and Reorder are polynomial-time solvable for the marriage case. We complement this result by showing that for the roommates setting, both CSR-Swap-MS and CSR-Reorder-MS become NP-complete.

Theorem 4.4 (⋆). *CSR*-Swap-MS *is* NP*-complete.*

Proof. Membership in NP is straightforward, since checking whether a given matching is stable can be done in polynomial time. To show NP-hardness, we reduce from the following NP-complete variant [6] of the 3SAT problem which we call (2P,2N)-3SAT: given a 3CNF Boolean formula ϕ (i.e., a set of clauses each containing exactly 3 literals) over a variable set X, such that each variable appears *exactly* twice unnegated and twice negated, decide if ϕ is satisfiable.

Let (X, ϕ) be an instance of (2P,2N)-3SAT with $X = \{x_1, \ldots, x_{\hat{n}}\}$ and $\phi = \{C_1, \ldots, C_{\hat{m}}\}$. For each clause C_j, we fix an arbitrary order of its three literals and write $C_j = (l_j^1 \vee l_j^2 \vee l_j^3)$. We construct an instance of CSR-Swap-MS as follows: for each variable $x_i \in X$, we introduce four *variable agents* $x_i, \overline{x}_i, y_i,$ and $\overline{y}_i$; for each clause $C_j \in \phi$ and each $z \in [3]$, we introduce two *clause agents* c_j^z and d_j^z. The preferences are as follows (indices for clause agents taken modulo 3):

$$
\begin{array}{lll}
x_i\colon \overline{x}_i \succ \{c_j^z\colon l_j^z = x_i\} \succ y_i, & y_i\colon x_i, & \forall i \in [\hat{n}];\\
\overline{x}_i\colon x_i \succ \{c_j^z\colon l_j^z = \overline{x}_i\} \succ \overline{y}_i, & \overline{y}_i\colon \overline{x}_i, & \forall i \in [\hat{n}];\\
c_j^z\colon c_j^{z+1} \succ c_j^{z+2} \succ l_j^z \succ d_j^z, & d_j^z\colon c_j^z, & \forall j \in [\hat{m}],\ z \in [3].
\end{array}
$$

Since each literal occurs in exactly two clauses, each set $\{c_j^z\colon l_j^z = l\}$ has size 2, so the variable agents' preference lists have exactly four entries each. The target matching is $M \coloneqq \{\{x_i, y_i\}, \{\overline{x}_i, \overline{y}_i\}\colon i \in [\hat{n}]\} \cup \{\{c_j^z, d_j^z\}\colon j \in [\hat{m}],\ z \in [3]\}$, and we set the budget to $\ell \coloneqq 3\hat{n} + 5\hat{m}$.

We now give some intuition on why (X, ϕ) is satisfiable if and only if at most ℓ swaps suffice to obtain an instance with a stable matching $M' \subseteq M$. First, for each $i \in [\hat{n}]$ we need to ensure that $\{x_i, \overline{x}_i\}$ is not a blocking pair, which requires three swaps in the preferences of either x_i or $\overline{x}_i$; this choice corresponds to setting x_i to true or to false. Next, for each $j \in [\hat{m}]$ we need to ensure that no pair $\{c_j^a, c_j^b\}$ for $a, b \in [3]$ is a blocking pair; this requires five swaps and, importantly, leaves the preferences of c_j^z for some $z \in [3]$ unchanged. Since these swaps already use up the budget, for each agent c_j^z with untouched preferences we must have that the corresponding variable agent's preference list must have been changed (so that $\{c_j^z, l_j^z\}$ is not blocking)—this requirement translates to each clause containing a true literal. For the complete proof of correctness, see the full version [14]. ☐

For the reorder action, the same hardness can be obtained via a reduction from Vertex Cover that re-uses part of the construction from the proof of Theorem 3.4.

Proposition 4.5 (⋆). *CSR*-Reorder-MS *is* NP*-complete.*

Ensuring the existence of a stable matching. Finally, we consider the bribery counterpart of the existence goal ∃SM. In the marriage case, the existence of a stable matching is always guaranteed, so the question is only interesting in the roommates setting.

Theorem 4.6 (⋆). *CSR*-Reorder-∃SM*, CSR*-Reorder-∃PSM*, CSR*-Swap-∃SM*, and CSR*-Swap-∃PSM *are all* NP*-complete.*

Remark 4.7. Hardness for the swap action with the ∃PSM goal is in fact already known in the strictly easier marriage setting: in the context of the trade-off between optimality (size of the matching) and stability, CSM-Swap-∃PSM has been shown to be W[1]-hard parameterized by the swap budget ℓ [15], and this hardness lifts to CSR-Swap-∃PSM since Stable Marriage is a special case of Stable Roommates. The argument above is nonetheless included because it yields the ∃SM variants and the reorder cases in the same construction.

5 Conclusion

We studied *control* and *bribery* for stable matchings under five manipulative actions and six goals, mapping the complete complexity landscape for both the marriage and the roommates settings (Table 1).

Regarding our NP-hardness results, it is important to keep in mind that our results do not necessarily offer a barrier against control in practice. Nevertheless, several of our reductions go beyond classical NP-hardness: those in Theorems 3.2, 3.6, 4.1 and 4.6, Corollary 3.3, and Proposition 4.2 reduce from Clique or Independent Set with budget ℓ polynomial in the solution size, yielding W[1]-hardness parameterized by ℓ. The remaining new hardness results do not yield such parameterized lower bounds: the reductions in Theorems 3.4, 3.7 and 4.4 have budgets that are not bounded by the source parameter, and Proposition 4.5

reduces from Vertex Cover, which is fixed-parameter tractable; hence, the parameterized complexity of these cases remains open.

Several directions remain open. First, since several of our intractable cases are already W[1]-hard parameterized by the number ℓ of applied actions, a natural next step is to identify alternative parameters that may yield fixed-parameter tractability, such as bounds on preference-list lengths, structural restrictions of the acceptability graph (e.g., bounded treewidth/vertex cover), distances to restricted domains, or combined parameters (e.g., ℓ together with such a structural bound); the parameterized complexity of the cases that our reductions leave open is itself a concrete target. Second, one can study restricted preference classes (e.g., single-peaked/single-crossing preferences [2, 9, 10]) that may restore tractability for some action–goal combinations. Finally, it would be interesting to extend the same action/goal palette to richer matching models (ties, many-to-one, matching with contracts, distributional constraints) and to compare systematically with emerging control landscapes in coalition formation settings.

Acknowledgment

We are grateful to Jörg Rothe for enjoyable and fruitful discussions, and for his invitation to prepare a survey on control problems in the area of matching under preferences; the work contained in this paper originated in our efforts to write that survey. J. Chen is supported by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18012]. I. Schlotter is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2021-2) and its János Bolyai Research Scholarship.

References

[1] Abraham, D.J., Biró, P., Manlove, D.F.: “Almost stable” matchings in the roommates problem. In: Proc. of the 3rd Workshop on Approximation and Online Algorithms (WAOA ’05). LNCS, vol. 3879, pp. 1–14 (2006)

[2] Ballester, M., Haeringer, G.: A characterization of the single-peaked domain. Social Choice and Welfare **36**(2), 305–322 (2011)

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

[4] Bartholdi III, J.J., Tovey, C.A., Trick, M.A.: The computational difficulty of manipulating an election. Social Choice and Welfare **6**(3), 227–241 (1989)

[5] Bérczi, K., Csáji, G., Király, T.: Manipulating the outcome of stable marriage and roommates problems. Games and Economic Behavior **147**, 407–428 (2024)

[6] Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of MAX-3SAT. Tech. rep. (2003), report no. 49

[7] Biró, P., Manlove, D.F., Mittal, S.: Size versus stability in the marriage problem. Theoretical Computer Science **411**(16–18), 1828–1841 (2010)

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

[9] Bredereck, R., Chen, J., Finnendahl, U.P., Niedermeier, R.: Stable roommate with narcissistic, single-peaked, and single-crossing preferences. Autonomous Agents and Multi-Agent Systems **34**(53), 1–29 (2020)

[10] Bredereck, R., Chen, J., Woeginger, G.J.: A characterization of the single-crossing domain. Social Choice and Welfare **41**(4), 989–998 (2013)
[11] Chen, J., Guttmann, J., Mustajbašić, M., Simola, S.: Control in hedonic games. In: Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '26). pp. 920–928 (2026)
[12] Chen, J., Hermelin, D., Sorge, M., Yedidsion, H.: How hard is it to satisfy (almost) all roommates? In: Proc. of the 45th International Colloquium on Automata, Languages, and Programming (ICALP '18). LIPIcs, vol. 107, pp. 35:1–35:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2018)
[13] Chen, J., Kaczmarek, J., Nüsken, P., Rothe, J., Schlotter, I., Seeger, T.: Control in computational social choice. In: Proc. of the 34th International Joint Conference on Artificial Intelligence (IJCAI '25). pp. 10391–10399 (2025), Survey Track.
[14] Chen, J., Schlotter, I.: Control and bribery in stable marriage and stable roommates: A complete complexity landscape (2025), arXiv:2502.01215 [cs.GT]
[15] Chen, J., Skowron, P., Sorge, M.: Matchings under preferences: Strength of stability and tradeoffs. ACM Trans. Econ. Comput. **9**(4) (2021)
[16] Cheng, C.T., Rosenbaum, W.: Bridging the gap between stable marriage and stable roommates: A parameterized algorithm for optimal stable matchings (2026), arXiv:2603.26943 [cs.DS]
[17] Fortune, S., Hopcroft, J., Wyllie, J.: The directed subgraph homeomorphism problem. Theoretical Computer Science **10**(2), 111–121 (1980)
[18] Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly **69**(1), 9–15 (1962)
[19] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company (1979)
[20] Gupta, S., Jain, P.: Manipulation with(out) money in matching market. In: Proc. of the 8th International Conference on Algorithmic Decision Theory (ADT '24). LNCS, vol. 15248, p. 273–287 (2025)
[21] Gusfield, D., Irving, R.W.: The stable marriage problem: Structure and algorithms. Foundations of Computing Series, MIT Press, Cambridge, MA (1989)
[22] Irving, R.W.: An efficient algorithm for the "stable roommates" problem. Journal of Algorithms **6**(4), 577–595 (1985)
[23] Kamiyama, N.: Modifying an instance of the super-stable matching problem. Information Processing Letters **189**, 106549 (2025)
[24] Manlove, D.F.: Algorithmics of Matching Under Preferences, Series on Theoretical Computer Science, vol. 2. World Scientific Publishing (2013)
[25] Mnich, M., Schlotter, I.: Stable marriage with covering constraints: A complete computational trichotomy. Algorithmica **82**(1), 1136–1188 (2020)
[26] Tan, J.J.: A maximum stable matching for the roommates problem. BIT Numerical Mathematics **30**, 631–640 (1990)
[27] Tan, J.J.: A necessary and sufficient condition for the existence of a complete stable matching. Journal of Algorithms **12**, 154–178 (1991)
[28] Tan, J.J.: Stable matchings and stable partitions. International Journal of Computer Mathematics **39**, 11–20 (1991)
[29] Tan, J.J., Hsueh, Y.C.: A generalization of the stable matching problem. Discrete Applied Mathematics **59**, 87–102 (1995)