Deepening the (Parameterized) Complexity Analysis of Incremental Stable Matching Problems

Niclas Boehmer ✉
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Germany

Klaus Heeger ✉
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Germany

Rolf Niedermeier
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Germany

Abstract

When computing stable matchings, it is usually assumed that the preferences of the agents in the matching market are fixed. However, in many realistic scenarios, preferences change over time. Consequently, an initially stable matching may become unstable. Then, a natural goal is to find a matching which is stable with respect to the modified preferences and as close as possible to the initial one. For Stable Marriage/Roommates, this problem was formally defined as Incremental Stable Marriage/Roommates by Bredereck et al. [AAAI '20]. As they showed that Incremental Stable Roommates and Incremental Stable Marriage with Ties are NP-hard, we focus on the parameterized complexity of these problems. We answer two open questions of Bredereck et al. [AAAI '20]: We show that Incremental Stable Roommates is W[1]-hard parameterized by the number of changes in the preferences, yet admits an intricate XP-algorithm, and we show that Incremental Stable Marriage with Ties is W[1]-hard parameterized by the number of ties. Furthermore, we analyze the influence of the degree of "similarity" between the agents' preference lists, identifying several polynomial-time solvable and fixed-parameter tractable cases, but also proving that Incremental Stable Roommates and Incremental Stable Marriage with Ties parameterized by the number of different preference lists are W[1]-hard.

2012 ACM Subject Classification Theory of computation → W hierarchy; Theory of computation → Fixed parameter tractability; Theory of computation → Design and analysis of algorithms; Theory of computation → Algorithmic game theory

Keywords and phrases Stable Marriage, Stable Roommates, adapting to changing preferences, NP-hardness, W[1]-hardness, XP, FPT, master lists, incremental algorithms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.21

Related Version *Full Version*: https://arxiv.org/abs/2208.01563

Acknowledgements NB was supported by the DFG project MaMu (NI 369/19) and by the DFG project ComSoc-MPMS (NI 369/22). KH was supported by the DFG Research Training Group 2434 "Facets of Complexity" and by the DFG project FPTinP (NI 369/16).

1 Introduction

Efficiently adapting solutions to changing inputs is a core issue in modern algorithmics [3, 6, 9, 16, 27]. In particular, in *incremental* combinatorial problems, roughly speaking, the goal is to build new solutions incrementally while adapting to changes in the input. Typically, one wants to avoid (if possible) too radical changes in the solution relative to perhaps moderate changes in the input. The corresponding study of incremental algorithms attracted research on numerous problems and scenarios [24], including among many others shortest path computation [40], flow computation [31], clustering problems [9, 35], and graph coloring [28].

In this paper, we study the problem of adapting stable matchings under preferences to change. Consider for instance the following two scenarios: First, as reported by Feigenbaum et al. [18], school seats in public schools are centrally assigned in New York. Ahead of the

47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022).
Editors: Stefan Szeider, Robert Ganian, and Alexandra Silva; Article No. 21; pp. 21:1–21:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

start of the new year, all interested students are asked to submit their preferences over public schools. Then, a stable matching of students to public schools is computed and transmitted. However, in the past, shortly before the start of the new year typically around 10% of students changed their preferences and decided to attend a private school instead, leaving the initially implemented matching unstable and triggering lengthy decentralized ad hoc updates. Second, consider the assignment of freshmen to double bedrooms in college accommodation. After the orientation weeks, it is quite likely that students got to know each other (and in particular their roommates) better and thus their initially uninformed preferences changed, making the matching unstable.

In our work, we focus on the problem of finding a stable matching after the "change" that is as close as possible to a given initially stable matching. The closeness condition here is due to the fact that in most applications reassignments come at some cost which we want to minimize (e.g., in the above New York example, reassigning students might make it necessary for the family to reallocate within the city). We build upon the work of Bredereck et al. [7], who performed a first systematic study of incremental versions of stable matching problems, and the recent (partially empirical) follow-up work by Boehmer et al. [5], who proved among others that different types of changes are "equivalent" to each other. The central focus of our studies lies on the Stable Roommates (SR) problem: given a set of agents with each agent having preferences over other agents, the task is to find a stable matching, i.e., a matching so that there are no two agents preferring each other to their assigned partner. We also consider a famous special case of SR, namely Stable Marriage (SM), where the set of agents is partitioned into two sets, and each agent may only be matched to an agent from the other set. Formally, in the incremental versions of SR and SM, called Incremental Stable Roommates (ISR) and Incremental Stable Marriage (ISM), we are given two preference profiles containing the preferences of each agent before and after the "change" and a matching that is stable in the preference profile before the change. Then, the task is to find a matching that is stable after the change and as close as possible to the given matching, i.e., has a minimum symmetric difference to it.

Related Work. Bredereck et al. [7] formally introduced Incremental Stable Marriage [with ties] (ISM/[ISM-T]) and Incremental Stable Roommates [with ties] (ISR/[ISR-T]). They showed that ISM without ties (in the preference lists) is solvable in polynomial time by a simple reduction to finding a stable matching maximizing the weight of the included agent pairs (which is solvable in polynomial time [17]). In contrast to this, ISR is NP-complete [12, Theorem 4.2], yet admits an FPT-algorithm for the parameter k, that is, the maximum allowed size of the symmetric difference between the two matchings [7]. With ties, Bredereck et al. [7] showed that ISM-T and ISR-T are NP-complete and W[1]-hard for k even if the two preference profiles only differ in a single swap in some agent's preference list. As ISR-T can be considered as a generalization of ISM-T, their results motivate us to focus on the NP-hard ISR and ISM-T problems (which are somewhat incomparable problems). Recently, Boehmer et al. [5] followed up on the work of Bredereck et al. [7], proving that different types of changes such as deleting agents or performing swaps of adjacent agents in some preference list are "equivalent". Moreover, they introduced incremental variants of further stable matching problems and performed empirical studies.

More broadly considered, matching problems involving preferences in the presence of change are of high current interest in several application domains. Many such works fall into the category of "dynamic matchings" [1, 2, 14, 15, 34]. However, different from our work, there the focus is typically on adapting classic stability notions to dynamic settings while we rather aim for "reestablishing" (classic) stability at minimal change cost.

Table 1 Overview of our main results where each row contains results for one parameterization. Note that ISM is polynomial-time solvable as proven by Bredereck et al. [7].

	ISR	ISM-T
$\lvert\mathcal{P}_1 \oplus \mathcal{P}_2\rvert$	W[1]-h. (Th. 1) & XP (Th. 2)	NP-h. for $\lvert\mathcal{P}_1 \oplus \mathcal{P}_2\rvert = 1$ (Th. 5)
#ties+k	FPT wrt. k (Th. 1 in [7])	W[1]-h. even if $\lvert\mathcal{P}_1 \oplus \mathcal{P}_2\rvert = 1$ (Th. 5) XP (even for parameter #agents with ties) (Pr. 6)
#outliers	FPT (Th. 10)	?
#master lists	W[1]-h. even for complete preferences (Th. 11/12)	

Closer to our work, there are several papers on adapting a given matching to change (while minimizing the number of reassignments): First, Gajulapalli et al. [20] designed a polynomial-time (and incentive-compatible) algorithm for an incremental variant of the many-to-one version of Stable Marriage (known as Hospital Residents) where new agents are added. Second, Feigenbaum et al. [18] considered an incremental variant of Hospital Residents where some agents may leave the system. They designed a "fair", Pareto-efficient, and strategy-proof algorithm for finding a matching before and after the change. Both these works are closest related to the polynomial-time solvable ISM problem, which we do not study. Third, Bhattacharya et al. [3] studied one-to-one matching markets where agents are added and deleted over time and for some agents the set of acceptable partners may change. Their focus is on updating the matching in each step such that the number of reassignments is small while maintaining a small unpopularity factor. So in contrast to our work, they do not maintain that the matching is stable but (close to) popular.

Also motivated by temporally evolving preferences, several papers study the robustness of stable matchings subject to changing preferences [4, 11, 21, 22, 23, 36]. By selecting a robust initial stable matching, one can increase the odds that it remains stable after some changes.

Our Contributions. Focusing on the two NP-hard problems ISR and ISM-T, we significantly extend the work of Bredereck et al. [7] on incremental stable matchings, particularly answering their two main open questions. Moreover, we strengthen several of their results. In addition, we analyze the impact of the degree of "similarity" between the agents' preference lists. Doing so, from a conceptual perspective, we complement work of Meeks and Rastegari [38]. They studied the influence of the number of agent types on the computational complexity of stable matching problems (two agents are of the same type if they have the same preferences and all other agents are indifferent between them). By way of contrast, we consider the smaller so far unstudied parameter "number of different preference lists".

Next, we present a brief summary of the structure of the paper (for each section marking the main studied problem(s)) and our main contributions (see Table 1 for an overview):

Section 3 (ISR). Motivated by the observation that ISM-T is NP-hard even if just one swap has been performed, Bredereck et al. [7] asked for the parameterized complexity of ISR with respect to the difference $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ between the two given preference profiles. We design and analyze an involved algorithm solving ISR in polynomial time if $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ is constant (in other words, this is an XP-algorithm). Our algorithm relies on the observation that if we know how certain agents are matched in the matching to be constructed and we adapt the given matching accordingly, then we can find an optimal solution by propagating these changes through the matching until a new stable matching is reached; a general approach that might be of independent interest. We complement this result by proving that ISR parameterized by $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ is W[1]-hard.

Section 4 (ISM-T). Bredereck et al. [7] considered the total number of ties to be a promising parameter to potentially achieve fixed-parameter tractability results. We prove that this is not the case as ISM-T is W[1]-hard with respect to k *plus* the total number of ties even if $|\mathcal{P}_1 \oplus \mathcal{P}_2| = 1$. Notably, this result strengthens the W[1]-hardness with respect to k for $|\mathcal{P}_1 \oplus \mathcal{P}_2| = 1$ of Bredereck et al. [7] for ISM-T, while presenting a fundamentally different yet less technical proof. On the positive side, we devise an XP-algorithm for the number of agents with at least one tie in their preferences.

Section 5 (ISR; ISM-T). We study different cases where the agents have "similar" preferences. For instance, we consider the case where all but s agents have the same preference list (we call these s agents *outliers*), or the case where each agent has one out of only p different *master preference lists*. We devise an algorithm that enumerates all stable matchings in an SR instance in FPT time with respect to s, implying an FPT algorithm for ISR parameterized by s. In contrast to this and to a simple FPT algorithm for the number of agent types [5], we prove that ISR and ISM-T are W[1]-hard with respect to the number p of different preference lists.

2 Preliminaries

The input of Stable Roommates with Ties (SR-T) is a set $A = \{a_1, \dots, a_{2n}\}$ of agents. Each agent $a \in A$ has a subset $\mathrm{Ac}(a) \subseteq A \setminus \{a\}$ of agents it *accepts* and a preference relation $\succsim_a$ which is a weak order over the agents $\mathrm{Ac}(a)$. Without loss of generality, we assume that acceptance is symmetric, i.e., for two agents $a, a' \in A$, $a' \in \mathrm{Ac}(a)$ implies $a \in \mathrm{Ac}(a')$. We collect the preferences of all agents in a *preference profile* $\mathcal{P}$. For two agents $a', a'' \in \mathrm{Ac}(a)$, agent a *weakly prefers* a' to a'' if $a' \succsim_a a''$. If a weakly prefers a' to a'' but does not weakly prefer a'' to a', then a *strictly prefers* a' to a'', and we write $a' \succ_a a''$. If a weakly but not strictly prefers a' to a'', then a is *indifferent* between a' and a'' and we write $a' \sim_a a''$; in other words, a' and a'' are *tied*. If a strictly prefers a' to a'' or $a' = a''$ holds, then we write $a' \succeq_a a''$. We say that an agent a has strict preferences, which we denote as $\succ_a$, if $\succsim_a$ is a strict order, and, in this case, we use the terms "strictly prefer" and "prefer" interchangeably. For two preference relations $\succsim$ and $\succsim'$ defined over the same set, the swap distance between $\succsim$ and $\succsim'$ is the number of agent pairs that are ordered differently by the two relations, i.e., $|\{(a,b) : a \succ b \wedge b \succsim' a\}| + |\{(a,b) : a \sim b \wedge \neg(a \sim' b)\}|$; for two preference relations over different sets, we define the swap distance to be infinite. For two preference profiles $\mathcal{P}_1$ and $\mathcal{P}_2$ containing the preferences of the same agents, $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ denotes the total swap distance between the two preference relations of an agent summed over all agents.[1]

A *matching* M is a set of pairs $\{a, a'\}$ with $a \neq a' \in A$, $a \in \mathrm{Ac}(a')$, and $a' \in \mathrm{Ac}(a)$, where each agent appears in at most one pair. In a matching M, an agent a is *matched* if a is part of one pair from M; otherwise, a is *unmatched*. A *perfect matching* is a matching in which all agents are matched. For a matching M and an agent $a \in A$, we denote by $M(a)$ the partner of a in M, i.e., $M(a) = a'$ if $\{a, a'\} \in M$ and $M(a) := \square$ if a is unmatched in M. All agents $a \in A$ strictly prefer any agent from $\mathrm{Ac}(a)$ to being unmatched, i.e., $a' \succ_a \square$ for all $a \in A$ and $a' \in \mathrm{Ac}(a)$.

[1] Notably, by the equivalence theorem of Boehmer et al. [5, Theorem 1], all our results (except for Theorem 5 where the constant $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ increases by a small number) still hold if $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ instead denotes the number of agents whose preferences changed, the number of deleted agents (i.e., the number of agents with empty preferences in $\mathcal{P}_2$ and non-empty preferences in $\mathcal{P}_1$), or the number of added agents (i.e., the number of agents with empty preferences in $\mathcal{P}_1$ and non-empty preferences in $\mathcal{P}_2$).

Two agents $a \neq a' \in A$ *block* a matching M if a and a' accept each other and strictly prefer each other to their partners in M, i.e., $a \in \mathrm{Ac}(a')$, $a' \in \mathrm{Ac}(a)$, $a' \succ_a M(a)$, and $a \succ_{a'} M(a')$. A matching M is *stable* if it is not blocked by any agent pair.[2] An agent pair $\{a, a'\}$ is called a *stable pair* if there is a stable matching M with $\{a, a'\} \in M$. For two matchings M and M', we denote by $M \triangle M'$ the set of pairs that appear only in M or only in M', i.e., $M \triangle M' = \{\{a, a'\} \mid (\{a, a'\} \in M \wedge \{a, a'\} \notin M') \vee (\{a, a'\} \notin M \wedge \{a, a'\} \in M')\}$. The incremental variant of STABLE ROOMMATES [WITH TIES] is defined as follows.

INCREMENTAL STABLE ROOMMATES [WITH TIES] (ISR/[ISR-T])
Input: A set A of agents, two preference profiles $\mathcal{P}_1$ and $\mathcal{P}_2$ containing the strict [weak] preferences of all agents, a stable matching M_1 in $\mathcal{P}_1$, and an integer k.
Question: Is there a matching M_2 that is stable in $\mathcal{P}_2$ such that $|M_1 \triangle M_2| \leq k$?

We also consider the incremental variant of STABLE MARRIAGE. Instances of STABLE MARRIAGE are instances of STABLE ROOMMATES where the set of agents is partitioned into two sets U and W such that agents from one of the sets only accept agents from the other set, i.e., $\mathrm{Ac}(m) \subseteq W$ for all $m \in U$ and $\mathrm{Ac}(w) \subseteq U$ for all $w \in W$. Following traditional conventions, we refer to the agents from U as men and to the agents from W as women. All definitions from above still analogously apply to STABLE MARRIAGE. Thus, in INCREMENTAL STABLE MARRIAGE [WITH TIES] (ISM/[ISM-T]), we are given a set $A = U \cupdot W$ of agents and two preference profiles $\mathcal{P}_1$ and $\mathcal{P}_2$ containing the strict [weak] preferences of all agents, where each $m \in U$ accepts only agents from W and the other way round.

Lastly, in STABLE ROOMMATES, the preferences of an agent $a \in A$ are *complete* if $\mathrm{Ac}(a) = A \setminus \{a\}$. In STABLE MARRIAGE, the preferences of an agent $a \in U \cupdot W$ are *complete* if $\mathrm{Ac}(a) = W$ for $a \in U$ or if $\mathrm{Ac}(a) = U$ for $a \in W$. If the preferences of an agent are not complete, then they are *incomplete*.

We defer the proofs (or their completions) of all results marked by ($\star$) to a full version.

3 Incremental Stable Roommates Parameterized by $|\mathcal{P}_1 \oplus \mathcal{P}_2|$

Bredereck et al. [7] showed that ISR-T and ISM-T are NP-hard even if $\mathcal{P}_1$ and $\mathcal{P}_2$ differ only by a single swap. While Bredereck et al. showed that ISR (without ties) is NP-hard, they asked whether it is fixed-parameter tractable parameterized by $|\mathcal{P}_1 \oplus \mathcal{P}_2|$. We show that ISR is W[1]-hard with respect to $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ (Section 3.1), yet admits an intricate polynomial-time algorithm for constant $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ (Section 3.2), thus still clearly distinguishing it from the case with ties.

3.1 W[1]-Hardness

This section is devoted to proving that ISR with respect to $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ is W[1]-hard:

▶ **Theorem 1** ($\star$). *ISR parameterized by $|\mathcal{P}_1 \oplus \mathcal{P}_2|$ is W[1]-hard.*

To prove the theorem, we reduce from the W[1]-hard MULTICOLORED CLIQUE problem parameterized by the solution size ℓ [39]. In MULTICOLORED CLIQUE, we are given an ℓ-partite graph $G = (V^1 \cupdot V^2 \cupdot \cdots \cupdot V^\ell, E)$ and the question is whether there is a clique X

[2] This definition of stability in the presence of ties is the by far most frequently studied variant known as *weak stability*. *Strong stability* and *super stability* are the two most popular alternatives. Notably, ISM-T (as defined later) becomes polynomial-time solvable for both strong and super stability, as for these two stability notions a stable matching maximizing a given weight function on all pairs of agents can be found in polynomial time [19, 32, 33].

of size ℓ in G with $X \cap V^c \neq \emptyset$ for all $c \in [\ell]$. To simplify notation, we assume that $V^c = \{v_1^c, \ldots, v_\nu^c\}$ for all $c \in [\ell]$ and that the given graph G is r-regular for some $r \in \mathbb{N}$. We refer to the elements of $[\ell]$ as *colors* and say that a vertex v has color $c \in [\ell]$ if $v \in V^c$. The structure of the reduction is as follows. For each color $c \in [\ell]$, there is a vertex-selection gadget, encoding which vertex from V^c is part of the multicolored clique. Furthermore, there is one edge gadget for each edge. Unless both endpoints of an edge are selected by the corresponding vertex-selection gadgets, the matching M_2 selected in the edge gadget contributes to the difference $M_1 \triangle M_2$ between M_1 and M_2. We set k (that is, the upper bound on $|M_1 \triangle M_2|$) such that at least $\binom{\ell}{2}$ edges need to have both endpoints in the selected set of vertices, implying that the selected set of vertices forms a clique.

Vertex-Selection Gadget. For each color $c \in [\ell]$, we add a vertex selection gadget. For each vertex $v_i^c \in V^c$, we add one 4-cycle consisting of agents $a_{i,1}^c$, $a_{i,2}^c$, $a_{i,3}^c$, and $a_{i,4}^c$. Further, in $\mathcal{P}_2$, two agents s^c and $\bar{s}^c$ are "added" to the gadget (more formally, s^c and $\bar{s}^c$ are matched to dummy agents t^c and $\bar{t}^c$ in all stable matching in $\mathcal{P}_1$ but cannot be matched to t^c and $\bar{t}^c$ in a stable matching in $\mathcal{P}_2$). We construct the vertex-selection gadget such that the agents s^c and $\bar{s}^c$ have to be matched to agents from the same 4-cycle in a stable matching in $\mathcal{P}_2$. This encodes the selection of the vertex corresponding to this 4-cycle to be part of the multicolored clique. Lastly, we add a second 4-cycle consisting of agents $\bar{a}_{i,1}^c$, $\bar{a}_{i,2}^c$, $\bar{a}_{i,3}^c$, and $\bar{a}_{i,4}^c$ for each vertex $v_i^c \in V^c$ to achieve that $M_1 \triangle M_2$ contains the same number of pairs from the vertex-selection gadget, independent of which vertex is selected to be part of the clique.

Apart from agents s^c and $\bar{s}^c$, all agents from the vertex-selection gadget only find agents from the gadget acceptable, while s^c and $\bar{s}^c$ also find agent $a_{e,1}$ (this agent will be introduced in the next paragraph "Edge Gadget") for each edge e incident to a vertex from V^c acceptable. For $v_i^c \in V^c$, let $A_{\delta(v_i^c),1}$ denote the set of agents $a_{e,1}$ such that e is an edge incident to v_i^c, i.e., $A_{\delta(v_i^c),1} := \{a_{e,1} \mid e \in E \wedge e \cap v_i^c \neq \emptyset\}$, and let $[A_{\delta(v_i^c),1}]$ denote an arbitrary strict order of $A_{\delta(v_i^c),1}$. For all $c \in [\ell]$ and $i \in [n]$, each vertex-selection gadget contains the following agents with the indicated preferences in $\mathcal{P}_1$:

$$s^c : t^c \succ a_{1,1}^c \succ \bar{a}_{1,1}^c \succ [A_{\delta(v_1^c),1}] \succ a_{2,1}^c \succ \bar{a}_{2,1}^c \succ [A_{\delta(v_2^c),1}] \succ \cdots \succ a_{n,1}^c$$
$$\succ \bar{a}_{n,1}^c \succ [A_{\delta(v_1^n),1}]$$
$$\bar{s}^c : \bar{t}^c \succ a_{n,4}^c \succ \bar{a}_{n,4}^c \succ [A_{\delta(v_n^c),1}] \succ a_{n-1,4}^c \succ \bar{a}_{n-1,4}^c \succ [A_{\delta(v_{n-1}^c),1}] \succ \cdots \succ a_{1,4}^c$$
$$\succ \bar{a}_{1,4}^c \succ [A_{\delta(v_1^c),1}]$$
$$t^c : s^c \succ u^c, \qquad \bar{t}^c : \bar{s}^c \succ \bar{u}^c, \qquad u^c : t^c, \qquad \bar{u}^c : \bar{t}^c$$
$$a_{i,1}^c : a_{i,2}^c \succ s^c \succ a_{i,4}^c, \quad a_{i,2}^c : a_{i,3}^c \succ a_{i,1}^c, \quad a_{i,3}^c : a_{i,4}^c \succ a_{i,2}^c, \quad a_{i,4}^c : a_{i,1}^c \succ \bar{s}^c \succ a_{i,3}^c$$
$$\bar{a}_{i,1}^c : \bar{a}_{i,2}^c \succ s^c \succ \bar{a}_{i,4}^c, \quad \bar{a}_{i,2}^c : \bar{a}_{i,3}^c \succ \bar{a}_{i,1}^c, \quad \bar{a}_{i,3}^c : \bar{a}_{i,4}^c \succ \bar{a}_{i,2}^c, \quad \bar{a}_{i,4}^c : \bar{a}_{i,1}^c \succ \bar{s}^c \succ \bar{a}_{i,3}^c$$

In $\mathcal{P}_2$, only the preferences of agents t^c and $t^{\bar{c}}$ change to $u^c \succ s^c$, respectively, $\bar{u}^c \succ \bar{s}^c$. See Figure 1 for an example. Notably, in each of the added 4-cycles, there exist two matchings of the four agents that are stable within the cycle in both $\mathcal{P}_1$ and $\mathcal{P}_2$ (i.e., $\{\{a_{i,1}^c, a_{i,2}^c\}, \{a_{i,3}^c, a_{i,4}^c\}\}$ or $\{\{a_{i,1}^c, a_{i,4}^c\}, \{a_{i,2}^c, a_{i,3}^c\}\}$ and $\{\{\bar{a}_{i,1}^c, \bar{a}_{i,2}^c\}, \{\bar{a}_{i,3}^c, \bar{a}_{i,4}^c\}\}$ or $\{\{\bar{a}_{i,1}^c, \bar{a}_{i,4}^c\}, \{\bar{a}_{i,2}^c, \bar{a}_{i,3}^c\}\}$ for $c \in [\ell]$ and $i \in [\nu]$). Matching M_1 contains for the 4-cycles consisting of agents $\{a_{i,t}^c \mid t \in [4]\}$ the edges $\{a_{i,1}^c, a_{i,2}^c\}$ and $\{a_{i,3}^c, a_{i,4}^c\}$ and for the 4-cycles consisting of agents $\{\bar{a}_{i,t}^c \mid t \in [4]\}$ the edges $\{\bar{a}_{i,1}^c, \bar{a}_{i,4}^c\}$ and $\{\bar{a}_{i,2}^c, \bar{a}_{i,3}^c\}$.

Edge Gadget. For each edge $e = \{v_i^c, v_j^{\hat{c}}\}$, we add an edge gadget. This gadget consists of a 4-cycle with agents $a_{e,1}$, $a_{e,2}$, $a_{e,3}$, and $a_{e,4}$, admitting two different matchings that are stable *within* the gadget in both $\mathcal{P}_1$ and $\mathcal{P}_2$. The matching M_1 contains $\{a_{e,1}, a_{e,4}\}$ and $\{a_{e,3}, a_{e,2}\}$

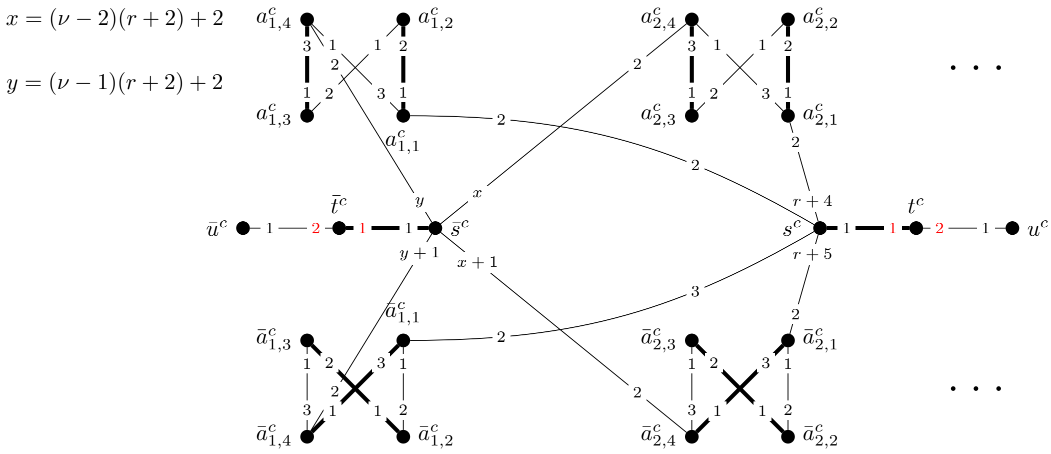

Figure 1 An example for the vertex-selection gadget from Theorem 1. For an edge between two agents a and a', the number x closer to agent a means that a ranks a' at position x, i.e., there are $x-1$ agents which a prefers to a'. For example, the preferences of $a^c_{1,4}$ are $a^c_{1,1} \succ \bar{s}^c \succ a^c_{1,3}$. The depicted preferences are those in $\mathcal{P}_1$. The preferences in $\mathcal{P}_2$ arise from swapping the red numbers. The matching M_1 is marked in bold.

in this 4-cycle and remains stable in M_2 if all of s^c, $\bar{s}^c$, $s^{\hat{c}}$, and $\bar{s}^{\hat{c}}$ are matched at least as good as the respective agents corresponding to v^c_i and $v^{\hat{c}}_j$. This notably only happens if the vertex-selection gadgets of V^c and $V^{\hat{c}}$ "select" the endpoints of e. Otherwise, the agents in this component need to be matched as $\{a_{e,1}, a_{e,2}\}$ and $\{a_{e,3}, a_{e,4}\}$ in M_2, thereby contributing four pairs to $M_1 \triangle M_2$. For each $e = \{v^c_i, v^{\hat{c}}_j\} \in E$, the agent's preferences are as follows:

$$a_{e,1} : a_{e,2} \succ s^c \succ \bar{s}^c \succ s^{\hat{c}} \succ \bar{s}^{\hat{c}} \succ a_{e,4}, \qquad a_{e,2} : a_{e,3} \succ a_{e,1},$$

$$a_{e,3} : a_{e,4} \succ a_{e,2}, \qquad a_{e,4} : a_{e,1} \succ a_{e,3}.$$

The Reduction. To complete the description of the parameterized reduction, we set $M_1 := \{\{s^c, t^c\}, \{\bar{s}^c, \bar{t}^c\} \mid c \in [\ell]\} \cup \{\{a^c_{i,1}, a^c_{i,2}\}, \{a^c_{i,3}, a^c_{i,4}\}, \{\bar{a}^c_{i,1}, \bar{a}^c_{i,4}\}, \{\bar{a}^c_{i,3}, \bar{a}^c_{i,2}\} \mid c \in [\ell], i \in [\nu]\} \cup \{\{a_{e,1}, a_{e,4}\}, \{a_{e,3}, a_{e,2}\} \mid e \in E\}$ and $k := \ell \cdot (4\nu + 5) + 4(|E| - \binom{\ell}{2})$.

For the correctness of the reduction one can show that in M_2 for each $c \in [\ell]$ there is some $i^* \in [\nu]$ such that the matching M_2 contains pairs $\{s^c, a^c_{i^*,1}\}$, $\{\bar{s}^c, a^c_{i^*,4}\}$ (this corresponds to selecting vertex $v^c_{i^*}$ for color c). Then, the only agents $a_{e,1}$ for an edge $e \in E$ incident to some vertex from V^c that both s^c and $\bar{s}^c$ do not prefer to their partner in M_2 are those in $A_{\delta(v^c_{i^*}),1}$. This implies that for all edges $e = \{v^c_i, v^{\hat{c}}_j\}$ with both endpoints selected we can match $a_{e,1}$ worse than s^c, $\bar{s}^c$, $s^{\hat{c}}$, and $\bar{s}^{\hat{c}}$. Thus, we can select $\{a_{e,1}, a_{e,4}\}, \{a_{e,2}, a_{e,3}\}$ as in M_1 in the respective edge gadget. In contrast, for all other edges we have to select the other matching in the edge gadget. To upper-bound the overall symmetric difference, one needs to further prove that for all $j < i^*$, matching M_2 contains $\{\{\bar{a}^c_{j,1}, \bar{a}^c_{j,2}\}, \{\bar{a}^c_{j,3}, \bar{a}^c_{j,4}\}\}$, and that for all $j > i^*$, matching M_2 contains $\{\{a^c_{j,1}, a^c_{j,4}\}, \{a^c_{j,2}, a^c_{j,3}\}\}$. Thus, independent of the selected vertex, each vertex-selection gadget contributes $4\nu + 5$ pairs to $M_1 \triangle M_2$.

3.2 XP-Algorithm

Complementing the above W[1]-hardness result, we now sketch an intricate XP-algorithm for ISR parameterized by $|\mathcal{P}_1 \oplus \mathcal{P}_2|$, resulting in the following theorem:

▶ **Theorem 2** ($\star$). *ISR can be solved in* $\mathcal{O}(2^{4|\mathcal{P}_1 \oplus \mathcal{P}_2|} \cdot n^{5|\mathcal{P}_1 \oplus \mathcal{P}_2|+3})$ *time.*

Our algorithm works in two phases: In the initialization phase, we make some guesses how the stable matching M_2 looks like and accordingly change the original stable matching M_1. These changes and guesses then impose certain constraints how good/bad some agents must be matched in M_2. Subsequently, in the propagation phase, we locally resolve blocking pairs caused by the initial changes by "propagating" these constraints through the instance until a new stable matching is reached. This matching is then guaranteed to be as close as possible to the original stable matching. We believe that our technique to propagate changes through a matching is also of independent interest and might find applications elsewhere. In the following, we sketch the main ingredients of our algorithm. We say that we *update a matching* M *to contain a pair* $e = \{a, b\}$ if we delete all pairs containing a or b from M and add pair e to M.

Initialization Phase (First Part of Description). Our algorithm maintains a matching M. At the beginning, we set $M := M_1$. Before we change M, we make some guesses on how the output matching M_2 shall look like. These guesses are responsible for the exponential part of the running time (the rest of our algorithm runs in polynomial time). The guesses result in some changes to M and, for some agents $a \in A$, in a "best case" and "worst case" to which partner a can be matched in M_2. Consequently, we will store in $\text{bc}(a)$ the *best case* how a may be matched in M_2, i.e., the most-preferred (by a in $\mathcal{P}_2$) agent b for which we cannot exclude that a is matched to b in a stable matching in $\mathcal{P}_2$ respecting the guesses. Similarly, $\text{wc}(a)$ stores the *worst case* to which a can be matched. We initialize $\text{bc}(a) = \text{wc}(a) = \perp$ for all $a \in A$, encoding that we do not know a best or worst case yet.

To be more specific, among others, in the initialization phase we guess for each agent $a \in A$ with modified preferences as well as for $M_1(a)$ how they are matched in M_2 and update M to include the guessed pairs. Moreover, as an unmatched agent a shall always have $\text{bc}(a) \neq \perp$ or $\text{wc}(a) \neq \perp$, we guess for all agents a that became unmatched by this whether they prefer $M_1(a)$ to $M_2(a)$ (in which case we set $\text{bc}(a) := M_1(a)$) or $M_2(a)$ to $M_1(a)$ (in which case we set $\text{wc}(a) := M_1(a)$). Our algorithm also makes further guesses in the initialization phase. However, in order to understand the purpose of these additional guesses, it is helpful to first understand the propagation phase in some detail. Thus, we postpone the description of the additional guesses to the end of this section.

Propagation Phase. After the initialization phase, blocking pairs for the current matching M force the algorithm to further change M and force a propagation of best and worst cases through the instance until a stable matching is reached. As our updates to M are in some sense "minimally invasive" and exhaustive, once M is stable in $\mathcal{P}_2$, it is guaranteed to be the stable matching in $\mathcal{P}_2$ which is closest to M_1 among all matchings respecting the initial guesses. At the core of the technique lies the simple observation that in an SR instance for each stable pair $\{c, d\}$ and each stable matching N not including $\{c, d\}$ exactly one of c and d prefers the other to its partner in N:

▶ **Lemma 3** ([26, Lemma 4.3.9]). *Let* N *be a stable matching and* $e = \{c, d\} \notin N$ *be a stable pair in an SR instance. Then either* $N(c) \succ_c d$ *and* $c \succ_d N(d)$ *or* $d \succ_c N(c)$ *and* $N(d) \succ_d c$.

From this we can draw conclusions in the following spirit: Assuming that for a stable pair $\{c, d\}$ in $\mathcal{P}_2$ we have that $\text{wc}(c) \succ_c d$, i.e., c is matched better than d in M_2, it follows from Lemma 3 that d is matched worse than c in M_2, implying that we can safely set $\text{bc}(d) = c$.

Algorithm 1 Simplified propagation step performed for a pair $\{a, b\}$ of two matched agents blocking M with $\text{bc}(b) = M(b)$ or for an unmatched agent a with $\text{wc}(a) \neq \bot$.

1: **if** a is unmatched **then** Let $e = \{a, c\}$ be the stable pair in $\mathcal{P}_2$ such that $c \succ_a^{\mathcal{P}_2} \text{wc}(a)$ and $\text{bc}(c) \succ_c^{\mathcal{P}_2} a$ (or $\text{bc}(c) = \bot$) and c is worst-ranked by a among all such pairs.
2: **else** Let $e = \{a, c\}$ be the stable pair in $\mathcal{P}_2$ such that $c \succ_a^{\mathcal{P}_2} M(a)$ and $\text{bc}(c) \succ_c^{\mathcal{P}_2} a$ (or $\text{bc}(c) = \bot$) and c is worst-ranked by a among all such pairs.
3: **if** no such pair exists **then** Reject this guess.
4: **else** Update M such that it contains e, set $\text{wc}(a) := c$ and $\text{bc}(c) := a$.
5:     **if** $M(a) \neq \square$ **then** $\text{bc}(M(a)) := a$.
6:     **if** $M(c) \neq \square$ **then** $\text{wc}(M(c)) := c$.

In the following, we will now explain simplified versions of some parts of the propagation phase, while leaving out others (in fact, for the full algorithm and proof of correctness an extensive analysis is needed; see our full version).

Assume for a moment that the current matching M is perfect and that there is a blocking pair for M in $\mathcal{P}_2$ (see Algorithm 1 for a pseudocode-description of the following procedure). Because M_1 is stable in $\mathcal{P}_1$, all pairs that currently block M_1 either involve an agent with changed preferences or resulted from previous changes made to M. Using this, one can show that at least one of the two agents from a blocking pair $\{a, b\}$, say b, will have $a \succ_b \text{bc}(b) = M(b)$. Thus, we know that b is matched worse than a in any stable matching in $\mathcal{P}_2$ respecting our current guesses. Accordingly, for $\{a, b\}$ not to block M_2, agent a has to be matched to b or better and, in particular, better than $M(a)$ in the solution. As a consequence, we update the worst case of a to be the next agent c which a prefers to $M(a)$ such that $\{a, c\}$ is a stable pair in $\mathcal{P}_2$, i.e., we set $\text{wc}(a) := c$ (see Line 4). This change is then further propagated through the instance. Note that from Lemma 3 it follows that if $\{a', a''\}$ is a stable pair in $\mathcal{P}_2$ and agent a'' is the worst possible partner of a' in a stable matching in $\mathcal{P}_2$ (or a' prefers its worst possible partner to a''), then agent a'' cannot be matched better than agent a' in a stable matching in $\mathcal{P}_2$. Thus, by setting $\text{wc}(a') := a''$ we also get $\text{bc}(a'') := a'$. Consequently, applying this to our previous update $\text{wc}(a) = c$, we can also set $\text{bc}(c) := a$ (see Line 4). Moreover, recall that a prefers c to a's current partner $M(a)$ in M. Thus, assuming that in a stable matching M^* in $\mathcal{P}_2$ one of a and $M(a)$ prefers the other to its partner in M^* and the other prefers its partner in M^*, we can use the same argument again and set $\text{bc}(M(a)) := a$ (see Line 5; we will discuss in the last paragraph in this section why this assumption can be made). For the update in Line 6 a similar reasoning applies.

So far we assumed that all agents are matched (which indeed needs to be the case for M_2 because we can delete all agents not matched by M_2 in a preprocessing step). Using this, whenever there is an unmatched agent a, one can show that it cannot be matched to $\text{bc}(a)$ or $\text{wc}(a)$. Thus, if $\text{wc}(a) \neq \bot$, then we match a to the next-better agent c before $\text{wc}(a)$ in its preferences such that $\{a, c\}$ is a stable pair in $\mathcal{P}_2$ and set $\text{wc}(a) = c$. Subsequently, we propagate this change as in the above described case of a blocking pair (see Algorithm 1). Otherwise, we have $\text{bc}(a) \neq \bot$ and we match a to the next-worse agent b after $\text{bc}(a)$ in the preferences of a such that $\{a, b\}$ is a stable pair in $\mathcal{P}_2$. Here, a slightly more complicated subsequent propagation step is needed (as described in our full version).

Repeating these steps, i.e., matching so far unmatched agents and resolving blocking pairs, eventually either results in a conflict (i.e., an agent preferring its worst case to its best case, or changing a pair which we guessed to be part of M_2) or in a stable matching. In the first case, we conclude that no stable matching obeying our guesses exists, while in the latter case, we found a stable matching obeying the guesses and minimizing the symmetric difference to M_1 among all such matchings. The reason for the optimality of this matching is that every matching obeying our initial guesses has to obey the best and worst cases at the termination of the algorithm, and the computed matching M contains all pairs from M_1 which comply with the best and worst cases.

Initialization Phase (Second Part of Description). In addition to the guesses described above, the algorithm guesses the set F of pairs from M_1 for which both endpoints prefer M_2 to M_1. Similarly, the algorithm guesses the set H of pairs from M_2 for which both endpoints prefer M_1 to M_2. Notably, one can prove that the cardinality of both F and H can be upper-bounded by $|\mathcal{P}_1 \oplus \mathcal{P}_2|$, ensuring XP-running time. The reason why we need to guess the set F is that pairs $\{a, b\}$ from M_1 may not be stable pairs in $\mathcal{P}_2$. In this case, a preferring $M(a)$ to b does not imply that b prefers a to $M(b)$. Thus, if we would treat the pairs from F as "normal" pairs, we would propagate an incorrect worst case in Line 5. Note that all pairs from $M \setminus M_1$ are stable pairs in $\mathcal{P}_2$, as we only add pairs that are stable over time (see Line 4). The reason why we need to guess the set H is more subtle but also due to fact that pairs from H might cause problems for our propagation step (see our full version). To incorporate our guesses, for each $\{a, b\} \in F$, we delete $\{a, b\}$ from M and set $\mathrm{wc}(a) = b$ and $\mathrm{wc}(b) = a$, while for each $\{a, b\} \in H$ we update M to include H. We remark that from the proof of Theorem 1 it follows that ISR is NP-complete even if we know for each agent a whose preferences changed as well as $M_1(a)$ how they are matched in M_2 and the set of pairs $F \subseteq M_1$ for which both endpoints prefer M_2 to M_1. This indicates that guessing the set H might be necessary for an XP-algorithm.

4 Incremental Stable Marriage with Ties Parameterized by the Number of Ties

Bredereck et al. [7] raised the question how the total number of ties influences the computational complexity of ISM-T. Note that the number of ties in a preference relation is the number of equivalence classes of the relation containing more than one agent. For instance the preference relation $a \sim b \sim c \succ d \sim e \succ f$ contains two ties, where the first tie has size three and the second tie has size two. In this section, following a fundamentally different and significantly simpler path than Bredereck et al., we show that their W[1]-hardness result for ISM-T parameterized by k for $|\mathcal{P}_1 \oplus \mathcal{P}_2| = 1$ still holds if we parameterize by k *plus* the number of ties. To prove this, we introduce a natural extension of ISM called Incremental Stable Marriage with Forced Edges (ISMFE). ISMFE differs from ISM in that as part of the input we are additionally given a subset $Q \subseteq M_1$ of the initial matching, and the question is whether there is a stable matching M_2 for the changed preferences with $|M_1 \triangle M_2| \leq k$ containing all pairs from Q, i.e., $Q \subseteq M_2$.

We first show that ISMFE with ties is intractable even if $|Q| = 1$ by reducing from a W[1]-hard local search problem related to finding a perfect stable matching with ties [37]:

▶ **Proposition 4** ($\star$). *ISMFE with ties parameterized by k and the summed number of ties in $\mathcal{P}_1$ and $\mathcal{P}_2$ is W[1]-hard, even if $|\mathcal{P}_1 \oplus \mathcal{P}_2| = 1$ and $|Q| = 1$ and only women have ties in their preferences.*

Second, we reduce ISMFE with ties to ISM-T. The general idea of this parameterized reduction is to replace a forced pair $\{m, w\} \in Q$ by a gadget consisting of $6(k + 1)$ agents. In M_1, we match the agents from the gadget in a way such that if m and w are matched differently in M_2, then, compared to M_1, the matching in the whole gadget needs to be changed, thereby exceeding the given budget k. This reduction implies:

▶ **Theorem 5** (⋆). *ISM-T parameterized by k and the summed number of ties in $\mathcal{P}_1$ and $\mathcal{P}_2$ is W[1]-hard, even if $|\mathcal{P}_1 \oplus \mathcal{P}_2| = 1$ and only women have ties in their preferences.*

We remark that our reduction also implies the W[1]-hardness of ISM-T parameterized by k if each tie has size at most two and $|\mathcal{P}_1 \oplus \mathcal{P}_2| = 1$.

On the algorithmic side, parameterized by the number of agents with at least one tie in their preferences in $\mathcal{P}_2$, ISM-T lies in XP. The idea of our algorithm is to first guess the partners of all agents in M_2 with a tie in their preferences in $\mathcal{P}_2$ and subsequently reduce the problem to an instance of Weighted Stable Marriage, which is polynomial-time solvable [17]. Note that parameterizing by the summed size of all ties results in fixed-parameter tractability, as we can iterate over all possibilities of breaking the ties and subsequently apply the algorithm for ISM.

▶ **Proposition 6** (⋆). *ISM-T parameterized by the number of agents with at least one tie in their preferences in $\mathcal{P}_2$ lies in XP. ISM-T parameterized by the summed size of all ties in $\mathcal{P}_2$ is fixed-parameter tractable.*

5 Master Lists

After having shown in the previous section that ISM-T and ISR mostly remain intractable even if we restrict several problem-specific parameters, in this section we analyze the influence of the structure of the preference profiles by considering what happens if the agents' preferences are similar to each other. The arguably most popular approach in this direction is to assume that there exists a single central order (called master list) and that all agents derive their preferences from this order. This approach has already been applied to different stable matching problems in the quest for making them tractable [8, 13, 29, 30]. Specifically, we analyze in Section 5.1 the case where the preferences of all agents follow a single master list, in Section 5.2 the case where all but few agents have the same preference list, and in Section 5.3 the case where each agent has one of few different preference lists (which generalizes the setting considered in Section 5.2).

5.1 One Master List

In an instance of Stable Marriage/Roommates with agent set A, we say that the preferences of agent $a \in A$ can be *derived* from some preference list $\succsim^*$ over agents A if the preferences of a are $\succsim^*$ restricted to $\mathrm{Ac}(a)$. If the preferences of all agents in $\mathcal{P}_2$ can be derived from the same strict preference list (which is typically called *master list*), then there is a unique stable matching in $\mathcal{P}_2$ which iteratively matches the so-far unmatched top-ranked agent in the master list to the highest ranked agent it accepts:

▶ **Observation 7.** *If all preferences in $\mathcal{P}_2$ can be derived from the same strict preference list, then ISR can be solved in linear time.*

This raises the question what happens when the master list is not a strict but a weak order. If the preferences of the agents may be incomplete, then reducing from the NP-hard Weakly Stable Pair problem (the question is whether there is a stable matching in an SM-T/SR-T instance containing a given pair [29, Lemma 3.4]), one can show that even assuming that all preferences are derived from a weak master list is not sufficient to make ISM-T or ISR-T polynomial-time solvable.

▶ **Observation 8** (⋆). *ISM-T and ISR-T are NP-hard even if all preferences in $\mathcal{P}_1$ and $\mathcal{P}_2$ can be derived from the same weak preference list.*

In contrast to this, if we assume that the preferences of agents in $\mathcal{P}_2$ are complete and derived from a weak master list, then we can solve ISM-T and ISR-T in polynomial time. While for ISM-T this follows from a characterization of stable matchings in such instances as the perfect matchings in a bipartite graph due to Irving et al. [29, Lemma 4.3], for ISR-T this characterization does not directly carry over. Thus, we need a new algorithm: Assume that the master list consists of q ties (possibly containing just one agent) and let $A_i \subseteq A$ be the set of agents from the ith tie for $i \in [q]$. Distinguishing between several cases, we build the matching M_2 by dealing for increasing $i \in [q]$ with each tie separately while greedily maximizing the overlap of the so-far constructed matching with M_1. Our algorithm exploits the observation that in a stable matching, for $i \in [q]$, all agents from A_i are matched to agents from A_i except if (i) $|\bigcup_{j \in [i-1]} A_j|$ is odd in which case one agent from A_i is matched to an agent from A_{i-1}, or (ii) if $|\bigcup_{j \in [i]} A_j|$ is odd in which case one agent from A_i is matched to an agent from A_{i+1}.

▶ **Proposition 9** (⋆). *If the preferences of agents in $\mathcal{P}_2$ are complete and derived from a weak master list, then ISM-T/ISR-T can be solved in polynomial time.*

5.2 Few Outliers

Next, we consider the case that almost all agents derive their complete preferences from a single strict preference list (we will call these agents *followers*), while the remaining agents (we will call those agents *outliers*) have arbitrary preferences. ISR is fixed-parameter tractable with respect to the number of outliers, as we show that all stable matchings in a Stable Roommates instance can be enumerated in FPT time with respect to this parameter:

▶ **Theorem 10** (⋆). *Given a Stable Roommates instance $(A, \mathcal{P})$ and a partitioning $F \cup S$ of the agents A such that all agents from F have complete preferences that can be derived from the same strict preference list, one can enumerate all stable matchings in $(A, \mathcal{P})$ in $\mathcal{O}(n^2 \cdot |S|^{|S|})$ time. Consequently, ISR is solvable in $\mathcal{O}(n^2 \cdot |S|^{|S|})$ time, where $|S|$ is the number of outliers in $\mathcal{P}_2$.*

If the master list may contains ties, then enumerating stable matchings becomes a lot more complicated, as here we have much more flexibility on how the agents are matched. Thus, we leave it open whether there exists a similar fixed-parameter tractability result for a weak master list (both in the roommates and marriage setting).

5.3 Few Master Lists

Motivated by the positive result from Section 5.2, in this section we consider the smaller parameter "number of different preference lists". Recall that Observation 7 states that if the preference lists of all agents are derived from a strict master list in a Stable Roommates instance, then there exists only one stable matching (even if the preferences of the agents may be incomplete). This raises the question what happens if there exist "few" master lists and each agent derives its preferences from one of the lists. To the best of our knowledge, the parameter "number of master lists" has not been considered before. However, it nicely complements (and lower-bounds) the parameter "number of agent types" as studied by Meeks and Rastegari [38]. Two agents are of the same type if they have the same preferences and all other agents are indifferent between them. Notably, Boehmer et al. [5, Proposition 5] proved that ISM-T is fixed-parameter tractable with respect to the number of agent types. Their algorithm also works for ISR-T.

If the preferences of agents are incomplete, then as proven in Observation 8, ISM-T is already NP-hard for just one weak master list. Moreover, note that a reduction of Cseh and Manlove [12, Theorem 4.2] implies that ISR with incomplete preferences is NP-hard even if the preferences of each agent are derived from one of two weak preference lists. Consequently, in this subsection we focus on the case with complete preferences.

In contrast to the two fixed-parameter tractability results for the number of outliers (Theorem 10) and the number of agent types [5], we show that parameterized by the number p of master lists, ISR is W[1]-hard even if the preferences of agents are complete:

▶ **Theorem 11** (⋆). *ISR is W[1]-hard parameterized by the minimum number p such that in $\mathcal{P}_2$ the preferences of each agent can be derived from one of p strict preference lists, even if in $\mathcal{P}_1$ as well as in $\mathcal{P}_2$ all agents have complete preferences.*

Containment of this problem in XP is an intriguing open question; in other words, is there a polynomial-time algorithm if the number of master lists is constant?

Recalling that ISM-T is polynomial-time solvable if agents have complete preferences derived from one weak master list (Proposition 9), we now ask the same question for ISM-T. Using a similar but slightly more involved reduction than for Theorem 11, we show that this problem is W[1]-hard with respect to the number of master lists.

▶ **Theorem 12** (⋆). *ISM-T is W[1]-hard parameterized by the minimum number p such that in $\mathcal{P}_2$ the preferences of each agent can be derived from one of p weakly ordered preference lists, even if in $\mathcal{P}_1$ as well as in $\mathcal{P}_2$ all agents have complete preferences.*

Again, it remains open whether ISM-T for a constant number of master lists is polynomial-time solvable or NP-hard.

6 Conclusion

Among others, answering two open questions of Bredereck et al. [7], we have contributed to the study of the computational complexity of adapting stable matchings to changing preferences. From a broader algorithmic perspective, in particular, the "propagation" technique from our XP-algorithm for the number of swaps, and the study of the number of different preference lists/master lists as a new parameter together with the needed involved constructions for the two respective hardness proofs could be of interest.

There are several possibilities for future work. As direct open questions, for the parameterization by the number of outliers, we do not know whether ISM-T or ISR-T are fixed-parameter tractable. Moreover, it remains open whether ISR or ISM-T with complete preferences is polynomial-time solvable for a constant number of master lists.

Finally, it would also be interesting to analyze a variation of ISR or ISM where the matching in $\mathcal{P}_1$ is not given, i.e., we have to find two matchings M_1 and M_2 with $|M_1 \triangle M_2| \leq k$ such that M_1 is stable in $\mathcal{P}_1$ and M_2 is stable in $\mathcal{P}_2$. Notably, this is a special case of a multistage [16, 25] variant of stable matching problems and Chen et al. [10] already proved that this problem is NP-hard for $k = 0$ in the bipartite case.

References

1 Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Thickness and information in dynamic matching markets. *J. Political Econ.*, 128(3):783–815, 2020. 2

2 Mariagiovanna Baccara, SangMok Lee, and Leeat Yariv. Optimal dynamic matching. *Theor. Econ.*, 15(3):1221–1278, 2020. 2

3 Sayan Bhattacharya, Martin Hoefer, Chien-Chung Huang, Telikepalli Kavitha, and Lisa Wagner. Maintaining near-popular matchings. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP '15)*, pages 504–515. Springer, 2015. 1, 3

4 Niclas Boehmer, Robert Bredereck, Klaus Heeger, and Rolf Niedermeier. Bribery and control in stable marriage. *J. Artif. Intell. Res.*, 71:993–1048, 2021. 3

5 Niclas Boehmer, Klaus Heeger, and Rolf Niedermeier. Theory of and experiments on minimally invasive stability preservation in changing two-sided matching markets. *CoRR*, abs/2112.05777, 2021. Accepted at *AAAI'22*. arXiv:2112.05777. 2, 4, 12, 13

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

7 Robert Bredereck, Jiehua Chen, Dušan Knop, Junjie Luo, and Rolf Niedermeier. Adapting stable matchings to evolving preferences. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI '20)*, pages 1830–1837. AAAI Press, 2020. Full version available at arxiv.org/abs/1907.01375. 2, 3, 4, 5, 10, 13

8 Robert Bredereck, Klaus Heeger, Dušan Knop, and Rolf Niedermeier. Multidimensional stable roommates with master list. In *Proceedings of the 16th International Conferenc of the Web and Internet Economics (WINE '20)*, pages 59–73. Springer, 2020. 11

9 Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM J. Comput.*, 33(6):1417–1440, 2004. 1

10 Jiehua Chen, Rolf Niedermeier, and Piotr Skowron. Stable marriage with multi-modal preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC '18)*, pages 269–286. ACM, 2018. 13

11 Jiehua Chen, Piotr Skowron, and Manuel Sorge. Matchings under preferences: Strength of stability and tradeoffs. *ACM Trans. Economics and Comput.*, 9(4):20:1–20:55, 2021. 3

12 Ágnes Cseh and David F. Manlove. Stable marriage and roommates problems with restricted edges: Complexity and approximability. *Discret. Optim.*, 20:62–89, 2016. 2, 13

13 Lin Cui and Weijia Jia. Cyclic stable matching for three-sided networking services. *Comput. Networks*, 57(1):351–363, 2013. 11

14 Ettore Damiano and Ricky Lam. Stability in dynamic matching markets. *Games Econ. Behav.*, 52(1):34–53, 2005. 2

15 Laura Doval. Dynamically stable matching. *CoRR*, abs/1906.11391v5, 2021. arXiv:1906.11391. 2

16 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP '14)*, pages 459–470. Springer, 2014. 1, 13

17 Tomás Feder. A new fixed point approach for stable networks and stable marriages. *J. Comput. Syst. Sci.*, 45(2):233–284, 1992. 2, 11

18 Itai Feigenbaum, Yash Kanoria, Irene Lo, and Jay Sethuraman. Dynamic matching in school choice: Efficient seat reallocation after late cancellations. *Manag. Sci.*, 66(11):5341–5361, 2020. 1, 3

19 Tamás Fleiner, Robert W. Irving, and David F. Manlove. Efficient algorithms for generalized stable marriage and roommates problems. *Theor. Comput. Sci.*, 381(1-3):162–176, 2007. 5

20 Karthik Gajulapalli, James A. Liu, Tung Mai, and Vijay V. Vazirani. Stability-preserving, time-efficient mechanisms for school choice in two rounds. In *Proceedings of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '20)*, pages 21:1–21:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. 3

21 Begum Genc, Mohamed Siala, Barry O'Sullivan, and Gilles Simonin. Finding robust solutions to stable marriage. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI '17)*, pages 631–637. ijcai.org, 2017. 3

22 Begum Genc, Mohamed Siala, Barry O'Sullivan, and Gilles Simonin. Robust stable marriage. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI '17)*, pages 4925–4926. AAAI Press, 2017. 3

23 Begum Genc, Mohamed Siala, Gilles Simonin, and Barry O'Sullivan. Complexity study for the robust stable marriage problem. *Theor. Comput. Sci.*, 775:76–92, 2019. 3

24 Michel X. Goemans and Francisco Unda. Approximating incremental combinatorial optimization problems. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX/RANDOM 2017)*, pages 6:1–6:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. 1

25 Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing bases: Multistage optimization for matroids and matchings. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP '14)*, pages 563–575. Springer, 2014. 13

26 Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem – Structure and Algorithms*. Foundations of computing series. MIT Press, 1989. 8

27 Kathrin Hanauer, Monika Henzinger, and Christian Schulz. Recent advances in fully dynamic graph algorithms (invited talk). In *Proceedings of the 1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND '22)*, pages 1:1–1:47. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. 1

28 Sepp Hartung and Rolf Niedermeier. Incremental list coloring of graphs, parameterized by conservation. *Theor. Comput. Sci.*, 494:86–98, 2013. 1

29 Robert W. Irving, David F. Manlove, and Sandy Scott. The stable marriage problem with master preference lists. *Discret. Appl. Math.*, 156(15):2959–2977, 2008. 11, 12

30 Naoyuki Kamiyama. Many-to-many stable matchings with ties, master preference lists, and matroid constraints. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19)*, pages 583–591. IFAAMAS, 2019. 11

31 S. Kumar and P. Gupta. An incremental algorithm for the maximum flow problem. *J. Math. Model. Algorithms*, 2(1):1–16, 2003. 1

32 Adam Kunysz. An algorithm for the maximum weight strongly stable matching problem. In *Proceedings of the 29th International Symposium on Algorithms and Computation (ISAAC '18)*, pages 42:1–42:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. 5

33 Adam Kunysz, Katarzyna E. Paluch, and Pratik Ghosal. Characterisation of strongly stable matchings. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '16)*, pages 107–119. SIAM, 2016. 5

34 Ce Liu. Stability in repeated matching markets. *CoRR*, abs/2007.03794v2, 2021. `arXiv:2007.03794`. 2

35 Junjie Luo, Hendrik Molter, André Nichterlein, and Rolf Niedermeier. Parameterized dynamic cluster editing. *Algorithmica*, 83(1):1–44, 2021. 1

36 Tung Mai and Vijay V. Vazirani. Finding stable matchings that are robust to errors in the input. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA '18)*, pages 60:1–60:11. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. 3

37 Dániel Marx and Ildikó Schlotter. Parameterized complexity and local search approaches for the stable marriage problem with ties. *Algorithmica*, 58(1):170–187, 2010. 10

38 Kitty Meeks and Baharak Rastegari. Solving hard stable matching problems involving groups of similar agents. *Theor. Comput. Sci.*, 844:171–194, 2020. 3, 12

39 Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *J. Comput. Syst. Sci.*, 67(4):757–771, 2003. 5

40 G. Ramalingam and Thomas W. Reps. An incremental algorithm for a generalization of the shortest-path problem. *J. Algorithms*, 21(2):267–305, 1996. 1