Computational Complexity of k-stable Matchings

HARIS AZIZ, UNSW Sydney, Sydney, Australia
GERGELY CSÁJI, HUN-REN Centre for Economic and Regional Studies, Budapest, Hungary
ÁGNES CSEH, Universität Bayreuth, Bayreuth, Germany

We study deviations by a group of agents in the three main types of matching markets: the house allocation, the marriage, and the roommates models. For a given instance, we call a matching *k-stable* if no other matching exists that is more beneficial to at least k out of the n agents. The concept generalizes the recently studied majority stability [57]. We prove that whereas the verification of k-stability for a given matching is polynomial-time solvable in all three models, the complexity of deciding whether a k-stable matching exists depends on $\frac{k}{n}$ and is characteristic of each model.

CCS Concepts: • **Mathematics of computing** → *Matchings and factors*; • **Theory of computation** → *Algorithmic game theory;*

Additional Key Words and Phrases: Stable matching, popular matching, majority stability, algorithm, complexity

ACM Reference Format:
Haris Aziz, Gergely Csáji, and Ágnes Cseh. 2025. Computational Complexity of k-stable Matchings. *ACM Trans. Econ. Comput.* 13, 1, Article 5 (February 2025), 25 pages. https://doi.org/10.1145/3708507

1 Introduction

In matchings under preferences, agents seek to be matched among themselves or to objects. Each agent has a preference list of their possible partners. When an agent is asked to vote between two offered matchings, they vote for the one that allocates the more desirable partner to them. The goal of the mechanism designer is to compute a matching that guarantees some type of optimality. A rich literature has emerged from various combinations of input types and optimality conditions. In our article, we study three classic input types together with a new, flexible optimality condition that incorporates already defined notions as well.

Haris Aziz was supported by the NSF-CSIRO grant on "Fair Sequential Collective Decision-Making" (RG230833). Gergely Csáji acknowledges the financial support by the Hungarian Scientific Research Fund, OTKA Grant No. K143858, by the Momentum Grant of the Hungarian Academy of Sciences, grant number 2021-2/2021 and by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation fund, financed under the KDP-2023 funding scheme (grant number C2258525). An earlier and shorter version of this article has appeared at SAGT 2023 as Aziz et al. [5].
Author's Contact Information: Haris Aziz, UNSW Sydney, Sydney, Australia; e-mail: haris.aziz@unsw.edu.au; Gergely Csáji, HUN-REN Centre for Economic and Regional Studies, Budapest, Hungary; e-mail: csaji.gergely@gmail.com; Ágnes Cseh, Universität Bayreuth, Bayreuth, Germany; e-mail: agnes.cseh@uni-bayreuth.de.

ACM 2167-8375/2025/02-ART5
https://doi.org/10.1145/3708507

Table 1. Pointers to the Seminal Articles in Each Typical Combination of the Studied Models and Optimality Concepts

Model	Pareto optimality	Stability	Popularity
HA	Abraham et al. [1]		Abraham et al. [2]
SM		Gale and Shapley [29]	Gärdenfors [30]
SR		Irving [38]	Faenza et al. [25], Gupta et al. [32]

Input types. Our three input types differ in the structure of the underlying graph and the existence of objects as follows.

- *House allocation model (HA).* One side of a two-sided matching instance consists of agents who have strictly ordered, but possibly incomplete preferences and cast votes, while the other side is formed by objects with no preferences or votes.
- *Marriage model (SM).* Vertices on both sides of a two-sided matching instance are agents, who all have strictly ordered, but possibly incomplete preferences and cast votes.
- *Roommates model (SR).* The matching instance is not necessarily two-sided, all vertices are agents, who have strictly ordered, but possibly incomplete preferences and cast votes.

Optimality condition. For a given k, we say that a matching M is *k-stable* if there is no other matching M' that at least k agents prefer to M. Notice that this notion is highly restrictive, as the number of agents who prefer M to M' is not taken into account. Some special cases of k express very intuitive notions. The well-known notion of weak Pareto optimality is equivalent to n-stability; majority stability [57] is equivalent to $\frac{n+1}{2}$-stability, and finally, 1-stability asks whether there is a matching that assigns each agent their most preferred partner.

Structure of the article and techniques. We summarize relevant known results in Section 2 and lay the formal foundations of our investigation in Section 3. We start by describing polynomial-time algorithms for the case when k is a fixed constant in Section 4. Roughly speaking, these rely on the fact that we can guess the minimum number $k' \leq k$, such that there is a k'-stable matching and then guess k' agents who can improve at the same time in a k'-stable matching. We then turn to our complexity results for $k = cn$ values in the house allocation model in Section 5 and provide analogous proofs for the marriage and roommates models in Section 6. In the marriage model, we show that a majority stable matching always exists (a popular matching suffices), but this is the best we can do, as for $c \leq \frac{1}{2}$ the problem becomes NP-hard. For the roommates model, as stable matchings may fail to exist, we extend the NP-hardness to $c \leq \frac{2}{3}$ and using stable partitions (or stable half-matchings) we give an algorithm for $c > \frac{5}{6}$. For special preference domains, we give additional algorithms. We conclude in Section 7. Our proofs rely on tools from matching theory such as the famous Gallai–Edmonds decomposition, stable partitions, or scaling an instance with carefully designed gadgets.

2 Related Work

Matchings under preferences have been actively researched by both Economists and Computer Scientists [45, 51]. In this section, we highlight known results on the most closely related optimality concepts from the field. Table 1 summarizes the typical combinations of the studied models and optimality concepts.

2.1 Pareto Optimal Matchings

Pareto optimality is a desirable condition, most typically studied in the house allocation model. It is often combined with other criteria, such as lower and upper quotas. A matching M is *Pareto* *optimal* if there is no matching M', in which no agent is matched to an object they consider worse, while at least one agent is matched to an object they consider better than their object in M. A much less restrictive requirement implies *weak Pareto optimality*: M is weakly Pareto optimal if no matching M' exists that is preferred by all agents. This notion is equivalent to n-stability. The crucial difference is that Pareto optimality considers all agents, while k-stability (and thus weak Pareto optimality) only the ones who improve.

Weak Pareto optimality is mainly used in continuous and multi-objective optimization [23] and in economic theory [26, 59]. Pareto optimality is one of the most studied concepts in coalition formation and hedonic games [3, 7, 13, 24], and has also been defined in the context of various matching markets [4, 9, 14, 15]. As shown by Abraham et al. [1], in the house allocation model, a maximum size Pareto optimal matching can be found in polynomial time.

2.2 Stable Matchings

Possibly the most studied optimality notion for the marriage and roommates models is stability. A matching is *stable* if it is not *blocked* by any edge, that is, no pair of agents exists who are mutually inclined to abandon their partners for each other. The existence of stable matchings was shown in the seminal article of Gale and Shapley [29] for the marriage model. Later, Irving [38] gave a polynomial-time algorithm to decide whether a given roommates instance admits a stable matching. Tan [55] improved Irving's algorithm by providing an algorithm that always finds a so-called stable partition, which coincides with a stable matching if any exists. Stability was later extended to various other input settings in order to suit the growing number of applications such as employer matching markets [52], university admission decisions [6, 12], campus housing matchings [16, 49], and bandwidth matching [28]. The main difference between stability and k-stability is that, while blocking is a local property—two adjacent agents alone can block the matching—, k-stability is a global one, as it does not matter which k agents improve. For this reason, one may even argue that k-popularity would be a more suitable name. Our choice of words originates from the notion of majority stability of Thakur [57].

2.3 Popular Matchings

Popular matchings translate the simple majority voting rule into the world of matchings under preferences. Given two matchings M and M', matching M is more popular than M' if the number of vertices preferring M to M' is larger than the number of vertices preferring M' to M. A matching M is *popular* in an instance if there is no matching M' that is more popular than M.

The concept was first introduced by Gärdenfors [30] for the marriage model and then studied by Abraham et al. [2] in the house allocation model. Polynomial-time algorithms to find a popular matching were given in both models. In the marriage model, it was already noticed by Gärdenfors that all stable matchings are popular, which implies that in this model, popular matchings always exist. In fact, stable matchings are the smallest size popular matchings, as shown by Biró et al. [10], while maximum size popular matchings can be found in polynomial time as well [35, 39]. Only recently, Faenza et al. [25] and Gupta et al. [32] resolved the long-standing [10, 18, 34, 36, 45] open problem on the complexity of deciding whether a popular matching exists in a popular roommates instance and showed that the problem is NP-complete. This hardness extends to graphs with complete preference lists [20].

Besides the three matching models, popularity has also been defined for spanning trees [21], permutations [43, 58], the ordinal group activity selection problem [22], and very recently, for branchings [41]. Matchings nevertheless constitute the most actively researched area of the majority voting rule outside of the usual voting scenarios. Similar to Pareto optimality, the main difference to k-stability is that here all agents matter, so also the ones who get worse and vote against the new matching.

2.4 Relaxing Popularity

The two most commonly used notions for near-popularity are called *minimum unpopularity factor* [8, 35, 39, 41, 42, 46, 53] and *minimum unpopularity margin* [37, 40, 41, 46]. Both notions express that a near-popular matching is never beaten by too many votes in a pairwise comparison with another matching. We say that matching M' *dominates* matching M by a margin of $u - v$, where u is the number of agents who prefer M' to M, while v is the number of agents who prefer M to M'. The *unpopularity margin* of M is the maximum margin by which it is dominated by any other matching. As opposed to k-stability, the unpopularity margin takes the number of both the satisfied and dissatisfied agents into account when comparing two matchings.

Checking whether a matching M' exists that dominates a given matching M by a margin of k can be done in polynomial time by the standard popularity verification algorithms in all models [2, 10, 53]. Finding a least-unpopularity-margin matching in the house allocation model is NP-hard [46], which implies that for a given (general) k, deciding whether a matching with unpopularity margin k exists is also NP-complete. A matching of unpopularity factor 0, which is a popular matching, always exists in the marriage model, whereas deciding whether such a matching exists in the roommates model is NP-complete [25, 32].

The unpopularity margin of a matching expresses the degree of undefeatability of a matching admittedly better than our k-stability. We see a different potential in k-stability and majority stability. The fact that, compared to M, there is no alternative matching in which at least k agents improve simultaneously, is a strong reason for choosing M—especially if $k = \frac{n+1}{2}$. The decision maker might care about minimizing the number of agents who would mutually improve by switching to an alternative matching. If there is a matching, where a significant number of agents can improve simultaneously, then they may protest together against the central agency to change the outcome—even though it would make some other agents worse off—out of ignorance or lack of information about the preferences of others. The unpopularity margin and factor give no information on this aspect, as they only measure the relative number of improving and disimproving agents.

2.5 Majority Stability

The study of majority stable matchings was initiated very recently by Thakur [57]. The three well-known voting rules plurality, majority, and unanimity translate into popularity, majority stability, and Pareto optimality in the matching world. A matching M is called *majority stable* if no matching M' exists that is preferred by more than half of all voters to M. The concept is equivalent to $\frac{n+1}{2}$-stability in our terminology.

Thakur observed that majority stability, in sharp contrast to popularity, is strikingly robust to correlated preferences. Based on this, he argued that in application areas where preferences are interdependent, majority stability is a more desirable solution concept than popularity. He provided examples and simulations to illustrate that, unlike majority stable matchings, the existence of a popular matching is sensitive to even small levels of correlations across individual preferences. Via a linear programming approach, he also showed that the verification of majority stability is polynomial-time solvable in the house allocation model.

3 Preliminaries

Next, we describe our input settings in Section 3.1, formally define our optimality concepts in Section 3.2, and give a structured overview of all investigated problems in Section 3.3.

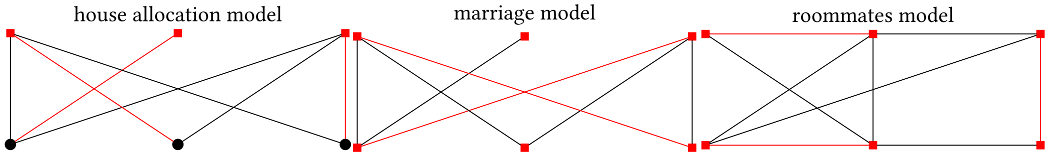

Fig. 1. A schematic picture of the three models we investigate in this article. Agents in N are denoted by red squares, while objects are denoted by black disks. The red edges constitute a matching in each figure.

3.1 Input

In the simplest of our three models, the house allocation model, we consider a set of agents $N = \{1, \ldots, n\}$ and a set of objects O. Each agent $i \in N$ has strict preferences $\succ_i$ over a subset of O, called the set of *acceptable* objects to i, while objects do not have preferences. The notation $o_1 \succ_i o_2$ means that agent i prefers object o_1 to object o_2. Being unmatched is considered worse by agents than being matched to any acceptable object or agent. To get a more complete picture, we also explore cases, where ties are allowed in the preference lists.

In the marriage model, no objects are present. Instead, the agent set $N = U \cup W$ is partitioned into two disjoint sets, and each agent seeks to be matched to an acceptable agent from the other set. In the roommates model, an agent from the agent set N can be matched to any acceptable agent in the same set.

A *matching* assigns each object to at most one agent and gives at most one acceptable object to each agent. For a matching M, we denote by $M(i)$ the object or agent assigned to agent $i \in N$, which can be the empty set as well. Each agent's preferences over objects or agents can be extended naturally to corresponding preferences over matchings. According to these extended preferences, an agent is indifferent among all matchings in which they are assigned to the same object or agent. Furthermore, agent i prefers matching M' to matching M if $M'(i) \succ_i M(i)$.

For clearer phrasing, we often work in a purely graph theoretical context. The *acceptability graph* of an instance consists of the agents and objects as vertices and the acceptability relations as edges between them. This graph is bipartite in the house allocation and marriage models. Figure 1 provides schematic pictures of acceptability graphs belonging to the three model types.

3.2 Optimality

Next, we define some standard optimality concepts from the literature. A matching M is

- *weakly Pareto optimal* [Mock] if there exists no other matching M' such that $M'(i) \succ_i M(i)$ for all $i \in N$;
- *majority stable* [Thakur] if there exists no other matching M' such that $|i \in N : M'(i) \succ M(i)| \geq \frac{n+1}{2}$;
- *popular* [Gärdenfors] if there exists no other matching M' such that $|i \in N : M'(i) \succ M(i)| > |i \in N : M(i) \succ M'(i)|$.

In words, weak Pareto optimality means that, compared to M, no matching is better for all agents, majority stability means that no matching is better for a majority of all agents, while popularity means that no matching is better for a majority of the agents who are not indifferent between the two matchings. It is easy to see that popularity implies majority stability, which in turn implies weak Pareto optimality.

We refine this scale of optimality notions by adding k-stability to it. A matching M is

- *k-stable* if there exists no other matching M' such that $|i \in N : M'(i) \succ M(i)| \geq k$.

In words, k-stability means that no matching M' is better for at least k agents than M—regardless of how many agents prefer M to M'. Weak Pareto optimality is equivalent to n-stability, while

Table 2. Each Problem Name Consists of Four Components, as Shown in the Columns of the Table

Optimality criterion	Model	Presence of ties	Completeness of preferences
k or MAJ	HA or SM or SR	T or $\emptyset$	C or I

Table 3. Our Results on the Complexity of Deciding Whether a k-stable Matching Exists Depending on the Constant c

Problem variant	HA	SM		SR	
	$0 < c < 1$	$c \leq \frac{1}{2}$	$c > \frac{1}{2}$	$c \leq \frac{2}{3}$	$c > \frac{5}{6}$
C, I, TC, TI	NP-c: Theorems 5.6, 5.7	NP-c: Theorem 6.4	P: Theorem 6.3	NP-c: Theorem 6.4	P: Theorem 6.2

NP-c abbreviates NP-complete, while P stands for polynomial-time solvable.

majority stability is equivalent to $\frac{n+1}{2}$-stability. It follows from the definition that k-stability implies $(k+1)$-stability.

We demonstrate k-stability on an example instance, which we will also use in our proofs later.

Example 3.1 (An $(n-1)$-stable Matching May Not Exist). Consider an instance in which $N = \{1, 2, \ldots, n\}$ and $O = \{o_1, o_2, \ldots, o_n\}$. Each agent has identical preferences of the form $o_1 > o_2 > \ldots > o_n$, analogously to the preferences in the famous example of Condorcet [17]. For an arbitrary matching M, each agent i of the at least $n-1$ agents, for whom $M(i) \neq o_1$ holds, could improve by switching to the matching that gives them the object directly above $M(i)$ in the preference list (or any object if i was unmatched in M). Therefore, no matching is majority stable or $(n-1)$-stable.

3.3 Our Problems and Contribution

Now, we define our central decision problems formally.

k-HAI	
Input:	Agent set N, object set O, a strict ranking $>_i$ over the acceptable objects for each $i \in N$, and an integer $1 \leq k \leq n$.
Question:	Does a k-stable matching exist?

Our problem names follow the conventions [45]. MAJ refers to majority-stability. HA stands for house allocation, SM for stable marriage, and SR for stable roommates. If the preference lists are complete, that is, if all agents find all objects acceptable, then we replace the I standing for incomplete by a C standing for complete. If ties are allowed in the preference lists, we add a T. Table 2 depicts a concise overview of the problem names.

For each of the two optimality criteria, there are $3 \cdot 2 \cdot 2 = 12$ problem variants. As shown in Section 4, all variants become tractable when k is a fixed constant. After this, our main goal was to solve all 12 variants for all $k = cn$, $0 < c < 1$ value, which by definition covers majority stability as well. Our results are summarized in Table 3. We only leave open the complexity of the four variants of k-SR for $\frac{2}{3} < c \leq \frac{5}{6}$.

In order to draw a more accurate picture in the presence of ties, we also investigate two standard input restrictions [11, 19, 50], see Table 4.

— DC: dichotomous and complete preferences, which means that agents classify all objects or other agents as either "good" or "bad", and can be matched to either one of these.
— STI: Possibly incomplete preferences consisting of a single tie, which again means that agents classify all objects or other agents as either "good" or "bad", and consider a bad match to be unacceptable.

Table 4. Results for the Two Restricted Settings in the Presence of Ties

Problem variant	HA			SM and SR		
	$c \leq \frac{1}{2}$	$\frac{1}{2} < c \leq \frac{2}{3}$	$c > \frac{2}{3}$	$c \leq \frac{1}{3}$	$\frac{1}{3} < c \leq \frac{1}{2}$	$c > \frac{1}{2}$
DC	NP-c: Theorem 5.9	NP-c: Theorem 5.9	P: Theorem 5.11	NP-c: Theorem 6.8	NP-c: Theorem 6.8	P: Theorem 6.7
STI	NP-c: Theorem 5.8	P: Theorem 5.11		NP-c: Theorem 6.6	P: Theorem 6.5	P: Theorem 6.5

Note that these two cases can be different, because if the preferences are DC, then each agent can be in one of three situations: matched to a good partner, matched to a bad partner, or remain unmatched. In STI, each agent is either matched to a good partner or unmatched.

Our hardness proofs rely on reductions from the problem named exact cover by 3-sets (X3C), which was shown to be NP-complete by Garey and Johnson [31].

> X3C
>
> **Input:** A set $\mathcal{X} = \{1, \ldots, 3\hat{n}\}$ and a family of 3-sets $\mathcal{S} \subset \mathcal{P}(\mathcal{X})$ of cardinality $3\hat{n}$ such that each element in $\mathcal{X}$ is contained in exactly three sets.
>
> **Question:** Are there $\hat{n}$ 3-sets in $\mathcal{S}$ that form an exact 3-cover of $\mathcal{X}$, that is, each element in $\mathcal{X}$ appears in exactly one of the $\hat{n}$ 3-sets?

4 Algorithms for Constant k

We start by showing that when k is a fixed constant, all problem variants are solvable in polynomial time. This also motivates the case when $k = cn, 0 < c < 1$. We first consider the house allocation model in Section 4.1 and then turn to the marriage and roommates models in Section 4.2.

4.1 Algorithm for the House Allocation Model

We describe our algorithm first, and then prove its correctness in Theorem 4.1. A pseudocode is provided in Algorithm 1. The main intuition is that if there is a k-stable matching, then we can guess k edges $(v_i, o_i), i \in [k]$ in it such that $v_1, \ldots, v_k$ cannot improve at the same time and then extend the matching while maintaining this property.

The algorithm iterates through each possible k' value from 0 to $k-1$ (line 2) and checks whether there exists a matching in which exactly k' agents can improve at the same time, but $k'+1$ agents cannot. Clearly, the instance admits a k-stable matching if and only if there is such a matching for some $k' \leq k-1$.

In one iteration, when k' is fixed (lines 3–19), the algorithm iterates through all possible matchings of size k' and starts constructing an envy graph for each. For a matching M, we say that the *envy graph* of M is the graph G_M which has vertex set $N \cup O$ and has an edge (i, o) for each pair such that $o \succ_i M(i)$. Let $M = \{(v_1, o_1), \ldots, (v_{k'}, o_{k'})\}$ be the currently checked matching (line 4). First, we only consider the vertex set $\{v_1, \ldots, v_{k'}\} \cup O$ together with the matching M and construct the envy graph of M and restrict to these vertices (line 6) (i.e., we delete the rest of the vertices and their incident edges). If this graph has a matching of size $k'+1$, then the algorithm ends this iteration and proceeds to the next one (line 8). Otherwise, let $v_{k'+1}, \ldots, v_n$ be the rest of the agents in N. For each of these agents v_j, the algorithm computes in line 15 the worst possible object $o_{j'} \in O \setminus \{o_1, \ldots, o_{k'}\}$ they could get without increasing the size of the maximum matching to at least $k'+1$ in the envy graph of the matching $M \cup \{(v_j, o_{j'})\}$ restricted to $G[\{v_1, \ldots, v_k\} \cup \{v_i\} \cup O]$. This can be done with the following simple subroutine. By going through the acceptable objects for agent v_i in the order of their preference (breaking the ties arbitrarily if needed), we add all edges to the envy graph that connect v_i to strictly better objects. If at one step, the addition of these edges creates a matching of size $k'+1$ in the current envy graph, then we can conclude that v_i must get a strictly better partner than the currently checked one, assuming the matching M is

ALGORITHM 1: k-HAI

1: **Input:** $G = (N, O, \succ)$.
2: **for** $k' = 0, \ldots, k-1$ **do**
3:   **for** every matching M of size k' **do**
4:     $M := \{(v_1, o_1), \ldots, (v_k, o_k)\}$ (by reindexing)
5:     $G' := G \setminus (\{v_1, \ldots, v_{k'}\} \cup \{o_1, \ldots, o_{k'}\})$
6:     Let H_M be the envy graph of M restricted to $G[\{v_1, \ldots, v_{k'}\} \cup O]$
7:     **if** H_M has a matching of size $k'+1$ **then**
8:       Proceed to the next matching in line 3
9:     **else**
10:       $U := \emptyset$
11:       **for** $j = k'+1, \ldots, n$ **do**
12:         **if** the envy graph of M restricted to $G[\{v_1, \ldots, v_{k'}\} \cup \{v_j\} \cup O]$ has no matching of size $k'+1$ **then**
13:           Continue
14:         **else**
15:           Let $o_{j'}$ be a worst object for v_j such that the maximum size matching in the envy graph of $M \cup \{(v_j, o_{j'})\}$ restricted to $G[\{v_1, \ldots, v_{k'}\} \cup \{v_j\} \cup O]$ has size strictly less than $k'+1$
16:           $U := U \cup \{v_j\}$
17:           Delete all edges from G' incident to v_j that go to strictly worse objects than $o_{j'}$
18:     **if** There is a matching M' in G' that covers U **then**
19:       **return** $M \cup M'$
20: **return** No k-stable matching exists.

already fixed. Hence, we add v_i to U in line 16 and delete edges from G' (defined in line 5) that would match him to at least these bad objects in line 17. If even the addition of all edges for an agent does not increase the size of the maximum matching in the envy graph, then we conclude that this particular agent can remain unmatched as it is not added to U in line 16, since the if-condition then was satisfied in line 12. Then, the algorithm checks whether the matching M can be extended in a way such that all other agents who must be matched get an as least as good object as the algorithm computed (line 18). If so, then the algorithm outputs this matching (line 19). If the algorithm does not find such a matching in any iteration, then it outputs the message "no k-stable matching exists" (line 20).

THEOREM 4.1. *Algorithm 1 solves each of k-HAC, k-HAI, k-HATC, and k-HATI in time $O(k(n \cdot |O|)^{k+2})$.*

PROOF. We show first that if Algorithm 1 outputs a matching M, then M is k-stable, by proving that the envy graph does not have a matching of size k' (the value of k' at the termination of the algorithm). As each agent from U got a partner adjacent to them in G' after the deletion of edges in line 17, the size of the maximum matching in the envy graph of M restricted to $\{v_1, \ldots, v_{k'}\} \cup \{v_i\} \cup O$ must be at most $k' \leq k-1$ for any $v_i \in N \setminus \{v_1, \ldots, v_{k'}\}$. We claim that this holds for the whole envy graph as well. Indeed, if the addition of two or more vertices together with their envy edges would increase the size of the maximum matching in the envy graph, then for any matching M' of size k' in the envy graph restricted to $\{v_1, \ldots, v_{k'}\} \cup O$, we must create an alternating path of M' and non-M' edges (meaning a path such that both the M' and the non-M' edges of it form a matching) whose first and last edges are both non-M' edges, by the Hungarian method [44] (otherwise we can conclude that M' is a maximum size matching). Such an alternating path P must start from an agent in $N \setminus \{v_1, \dots, v_{k'}\}$ and end at an unmatched object in M, as $v_1, \dots, v_{k'}$ are all matched by M. By choosing a shortest such alternating path, we get that it can only contain one vertex from $N \setminus \{v_1, \dots, v_{k'}\}$. But this means that only the addition of that vertex v_j and its envy edges increases the size of the maximum matching in the envy graph to $k \geq k' + 1$, so v_j received a strictly worse object than the computed $o_{j'}$ (line 15), which is a contradiction, as the algorithm then deleted this edge in line 17.

Secondly, we show that if there is a k-stable matching, then the algorithm must find one. Let M be a matching that is $(k'+1)$-stable for the smallest possible $k' \leq k-1$ value. Then, the envy graph of M must have a matching of size k', but none of size $k'+1$. In particular, there must be k' agents $\{v_1, \dots, v_{k'}\}$ who can improve simultaneously. Clearly, if the algorithm did not find a k-stable matching, then it must have had an iteration with exactly this k' value, these $\{v_1, \dots, v_{k'}\}$ vertices and also their partners $\{o_1, \dots, o_{k'}\}$ in M as the guessed matching in line 3. In that iteration, the envy graph H_M on $\{v_1, \dots, v_{k'}\} \cup O$ must have had a matching of size k', but then the algorithm still did not find a matching that is k-stable in lines 11–19. This could only happen because there was no matching, where all agents outside of $\{v_1, \dots, v_{k'}\}$ obtained an object such that the agent and their envy edges do not increase the size of the maximum matching in the envy graph. However, this is a contradiction, as M must be such a matching by our assumptions.

As for the running time, it is easy to see that the number of iterations for the first two loops in lines 2 and 3 are at most $O(k(n \cdot |O|)^k)$, because there are at most $\binom{n|O|}{k'} = O((n \cdot |O|)^{k'})$ matchings of size k' meaning that the loop in line 3 has at most this many iterations. In each such iteration, we do at most $n \cdot |O|$ checks of an increase in the maximum size matching in the envy graph during lines 11-19 (we have at most n iterations in the loop of line 11 and the subroutine in line 15 does at most $|O|$ such checks), which can be done with the Hungarian method [44] in $O(n \cdot |O|)$ time. Hence, the total running time is at most $O(k(n \cdot |O|)^{k+2})$. □

4.2 Algorithm for the Marriage and Roommates Models

In this subsection, we study the marriage and roommates models for constant k values. We start by describing our algorithm (Algorithm 2), and then prove its correctness in Theorem 4.2. The main idea behind the algorithm is that if we can guess the minimum k' such that there is a k'-stable matching, then we can also guess a minimum-size set of agents that can simultaneously improve in such a matching M and also their partners in M whom they can improve with, and even their partners in M. Then, in the rest of the graph, the matching must be 1-stable (for more details, see the proof). In the loop of line 2, our algorithm iterates through all possible matchings of size at most $2k-2$. Then, in line 3, it deletes the vertices covered by the guessed matching M from the graph. If this graph has a 1-stable matching M', then the algorithm checks in line 5 whether $M \cup M'$ is k-stable, and if so, it outputs $M \cup M'$ as a solution in line 6. If the algorithm fails to find a k-stable matching in the for-loop, it outputs "no k-stable matching exists" in line 8.

Theorem 4.2. *Algorithm 2 solves each of k-smc, k-smi, k-smtc, k-smti, k-src, k-sri, k-srtc, and k-srti in $O(kn^{4k-2})$ time.*

Proof. Clearly, if Algorithm 2 finds a matching that is k-stable, then the instance $G = (N, E, \succ)$ admits such a matching. Therefore, we need to prove that whenever a k-stable matching exists, the algorithm finds one.

Suppose G admits a k-stable matching and let $0 \leq k' \leq k-1$ be the smallest value k', such that the instance also admits a $(k'+1)$-stable matching and let M be such a matching. Then, compared to M, k' agents can improve at the same time, but $k'+1$ cannot. Let $\{v_1, \dots, v_{k'}\}$ be a set of k' agents who can improve at the same time in a matching M' and let $\{v_{2k'+1}, \dots, v_{3k'}\}$ be their partners in

ALGORITHM 2: k-SR

```
1: Input: G = (N, E, ≻).
2: for all matchings M of size at most 2k − 2 do
3:     G′ := G \ V(M), where V(M) is the set of vertices covered by M
4:     if G′ has a 1-stable matching M′ then
5:         if M ∪ M′ is k-stable then
6:             return M ∪ M′
7:     Proceed to the next matching
8: return No k-stable matching exists.
```

M. Furthermore, let $\{v_{k'+1}, \ldots, v_{2k'}\}$ be the partners of $\{v_1, \ldots, v_{k'}\}$ in M' and $\{v_{3k'+1}, \ldots, v_{4k'}\}$ be the partners of $\{v_{k'+1}, \ldots, v_{2k'}\}$ in M. Note that an agent may appear in more than one (even in all) of these four sets. Then, in the graph G' obtained by deleting all these vertices, it must hold that M restricted to G' is a 1-stable matching. Indeed, if there would be a way for an agent u in G' to be able to improve compared to M with an agent v, then by adding this edge (u, v) to $\{(v_1, v_{k'+1}), \ldots, (v_{k'}, v_{2k'})\}$ (this set may contain the same edge twice, in which case we only take it once) we would obtain a matching M'', where at least $k' + 1$ agents can improve simultaneously, which is a contradiction.

Therefore, when the algorithm is in the iteration corresponding to $\hat{M} = \{(v_i, v_{2k'+i}) \mid i \in [2k']\}$ (only taking each edge in the set once) in line 2, then it is indeed able to conclude that there is a matching in the remaining instance (after the deletion of $V(M)$) that is 1-stable in line 4. The fact that the algorithm checks this matching $\hat{M}$ follows from $|\hat{M}| \leq 2k' \leq 2k - 2$. Also, in this iteration, it also finds the unique 1-stable matching in the remaining instance, which contains the remaining edges of M. Therefore, the algorithm finds M in one of its iterations and also concludes that M is k-stable.

As the number of matchings of size at most $2k - 2$ is at most $O(kn^{4k-4})$, the number of iterations before termination is at most $O(kn^{4k-4})$. As in each iteration, a 1-stable matching, if any exists, can be found in linear time by looking at the first remaining entries in the preference lists. Furthermore, as checking if a matching is k-stable can be done in $O(n^2)$ time, we can conclude that the algorithm terminates in $O(kn^{4k-2})$ time. □

5 The House Allocation Model

In this section, we examine the computational aspects of k-stability and majority stability in the house allocation model for $k = cn, 0 < c < 1$. We first present positive results on verification in Section 5.1 and then turn to hardness proofs on existence and some solvable restricted cases in Section 5.2.

5.1 Verification

Thakur [57] constructed an integer linear program to check whether a given matching is majority stable. He showed that the underlying matrix of the integer linear constraints is unimodular (a square integer matrix having determinant +1 or −1) and hence the problem can be solved in polynomial time, for example, the book of [Schrijver et al.]. Here we provide a simple characterization of majority stable matchings, which also delivers a fast and simple algorithm for testing majority stability.

Our first observation characterizes 1-stable matchings.

Observation 5.1. *A matching is 1-stable if and only if each agent gets their most preferred object.*

To generalize this straightforward observation to k-stability and majority stability, we introduce the natural concept of an improvement graph. For a given matching M in an instance $(N, O, >)$, let $G_M = (N \cup O, E_{\text{imp}})$ be the corresponding *improvement graph*, where $(i, o) \in E_{\text{imp}}$ if and only if $o >_i M(i)$. In other words, the improvement graph consists of edges that agents prefer to their current matching edge. We also say that agent i *envies* object o if $(i, o) \in E_{\text{imp}}$.

Observation 5.2. *Matching M is k-stable if and only if G_M does not admit a matching of size at least k. In particular, M is majority stable if and only if G_M does not admit a matching of size at least $\frac{n+1}{2}$.*

Proof. It follows from the definition of G_M that G_M admits a matching M' of size at least k if and only if in M', at least k agents get a better object than in M. The non-existence of such a matching M' defines k-stability for M. □

Observation 5.2 delivers a polynomial verification method for checking k-stability and majority stability. Constructing G_M to a given matching M takes at most $O(m)$ time, where m is the number of acceptable agent-object pairs in total. Finding a maximum size matching in G_M takes $O(\sqrt{n}m)$ time [33].

Corollary 5.3. *For any $k \in \mathbb{N}$, it can be checked in $O(\sqrt{n}m)$ time whether a given matching is k-stable. In particular, verifying majority stability can be done in $O(\sqrt{n}m)$ time.*

5.2 Existence

By Corollary 5.3, all decision problems on the existence of a k-stable matching in the house allocation model are in NP. Our hardness proofs rely on reductions from the problem named exact cover by 3-sets (x3c), which was shown to be NP-complete by Garey and Johnson [31]. First, we present our results for the problem variants with possibly incomplete preference lists in Section 5.2.1, then extend these to complete lists in Section 5.2.2, and finally, we discuss the case of ties in the preferences in Section 5.2.3.

x3c	
Input:	A set $\mathcal{X} = \{1, \ldots, 3\hat{n}\}$ and a family of 3-sets $\mathcal{S} \subset \mathcal{P}(\mathcal{X})$ of cardinality $3\hat{n}$ such that each element in $\mathcal{X}$ is contained in exactly three sets.
Question:	Are there $\hat{n}$ 3-sets in $\mathcal{S}$ that form an exact 3-cover of $\mathcal{X}$, that is, each element in $\mathcal{X}$ appears in exactly one of the $\hat{n}$ 3-sets?

5.2.1 Incomplete Preferences.

Theorem 5.4. *k-hai is NP-complete even if each agent finds at most two objects acceptable.*

Proof. Let I be an instance of x3c, where $\mathcal{S} = \{S_1, \ldots, S_{3\hat{n}}\}$ is the family of 3-sets and $S_j = \{j_1, j_2, j_3\}$. We build an instance I' of k-hai as follows. For each set $S_j \in \mathcal{S}$ we create four agents s_j^1, s_j^2, s_j^3, t_j and an object p_j. For each element $i \in \mathcal{X}$ we create an object o_i and two dummy agents d_i^1, d_i^2. Altogether we have $12\hat{n} + 6\hat{n} = 18\hat{n}$ agents and $3\hat{n} + 3\hat{n} = 6\hat{n}$ objects. The preferences are described and illustrated in Figure 2.

We prove that there is a $(5\hat{n} + 1)$-stable matching M in I' if and only if there is an exact 3-cover in I.

Claim 1. *If I admits an exact 3-cover, then I' admits a $(5\hat{n} + 1)$-stable matching.*

Proof: Suppose that $S_{l_1}, \ldots, S_{l_{\hat{n}}}$ form an exact 3-cover. Construct a matching M as follows: for each $j \in [3\hat{n}]$ match t_j with p_j. For each $j \in \{l_1, \ldots, l_{\hat{n}}\}$ match s_j^f with o_{j_f} for $f \in [3]$. As each object is covered exactly once, M is a matching.

$$
\begin{array}{lll}
s_j^\ell : & o_{j_\ell} \succ p_j & \text{for } j \in [3\hat{n}], \ell \in [3] \\
t_j : & p_j & \text{for } j \in [3\hat{n}] \\
d_i^1, d_i^2 : & o_i & \text{for } i \in [3\hat{n}]
\end{array}
$$

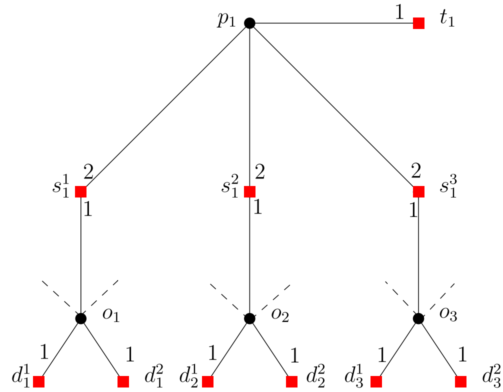

Fig. 2. The preferences and a graph illustration of a gadget of a set $S_1 = \{1, 2, 3\}$ in the proof of Theorem 5.4. Red squares are agents, black disks are objects, and the numbers on the edges indicate the preferences to the left of the graph. Dashed edges run to other gadgets. Each agent finds at most two objects acceptable.

We claim that M is $(5\hat{n} + 1)$-stable. Due to Observation 5.2, it is enough to show that at most $5\hat{n}$ objects are envied by any agent, so the improvement graph G_M has at most $5\hat{n}$ objects with non-zero degree. For each $j \in \{l_1, \dots, l_{\hat{n}}\}$, the object p_j is not envied by anyone, as all of s_j^1, s_j^2, s_j^3 got their best object and t_j got p_j. Hence, at most $2\hat{n}$ objects of type p_j and at most $3\hat{n}$ objects of type o_i are envied, proving our claim. ■

Claim 2. *If I' admits a $(5\hat{n} + 1)$-stable matching, then I admits an exact 3-cover.*

Proof: Let M be a $(5\hat{n} + 1)$-stable matching. First, we prove that in M at most $5\hat{n}$ objects are envied by any agent, because we can construct a matching M' that gives all envied objects to an agent who envies them. Each object o_i is envied in any matching by at least one of d_i^1 and d_i^2 and can be given to the envious agent in M'. Regarding envied objects of type p_j, one agent only finds at most one p_j object acceptable, which implies that each envied p_j can be assigned to an envious agent in M'. Therefore, M' is indeed a matching.

As at most $5\hat{n}$ objects can be envied and all o_i objects are envied in M, at most $2\hat{n}$ envied objects are of type p_j. This implies that at least $\hat{n}$ objects of type p_j are not envied by any agent. As these objects are the first choices of their s_j^ℓ, $\ell \in [3]$ agents, those agents must all get their first-choice object of type o_i. As M is a matching, these sets constitute an exact 3-cover. ■ □

Theorem 5.5. *MAJ-HAI is NP-complete even if each agent finds at most two objects acceptable.*

Proof. We extend our hardness reduction in the proof of Theorem 5.4. To show the hardness of MAJ-HAI, we add $8\hat{n}$ more gadgets to the instance I', each consisting of 3 agents and 3 objects, such that all 3 agents have the same preference order over their three corresponding objects. This gadget is a small version of our example instance in Example 3.1. It is easy to see that in any such gadget, there is a 3-stable matching, but there is no 2-stable matching. Hence, the new instance has $18\hat{n} + 8 \cdot 3\hat{n} = 42\hat{n}$ agents and there is a $5\hat{n} + 1 + 8 \cdot 2\hat{n} = (21\hat{n} + 1)$-stable matching if and only if there is an exact 3-cover. □

We now apply a more general scaling argument as in the proof of Theorem 5.5 to show that finding a k-stable matching is NP-complete for any non-trivial choice of k.

Theorem 5.6. *k-HAI is NP-complete for any constant $0 < c < 1$ and $k = cn$.*

Proof. Our construction in the proof of Theorem 5.4 can be extended to k-HAI by scaling the instance. This scaling happens through the addition of instances that either admit no k-stable matching even for a high k, or admit a k-stable matching even for a low k. The instance in Example 3.1 admits no $(n-1)$-stable matching. Constructing an instance with a 1-stable matching is easy: one $(1, o_1)$ edge suffices. For any constant c, we can add sufficiently many of these scaling instances to construct an instance that admits a cn-stable matching if and only if the original instance admits a majority stable matching. □

5.2.2 Complete Preferences. We now extend our hardness proof in Theorem 5.6 to cover the case of complete preference lists as well.

Theorem 5.7. *k-HAC is NP-complete for any constant $0 < c < 1$ and $k = cn$. In particular, MAJ-HAC is NP-complete.*

Proof. We start with an instance I of k-HAI and create an instance I' of k-HAC as follows. For each agent $i \in N$, we add a dedicated dummy object d_i. Then, we extend the preferences of the agents in a standard manner: we append their dedicated dummy object to the end of their original preference list, followed by all other objects in an arbitrary order.

Suppose first that there is a k-stable matching M in I. We construct a matching M' in I' by keeping all edges of M and assigning each unmatched agent in M to their dummy object in M'. As the improvement graph of M' in I' is the same as the improvement graph of M in I, M' is k-stable in I'.

Now suppose that there is a k-stable matching M' in I'. Construct a matching M by taking $M = M' \cap E$, where E denotes the edges of the acceptability graph of I. We claim that M is k-stable in I. Suppose that matching M'' is preferred to M by more than k agents in I. We claim that M'' is preferred to M' as well by more than k agents in I'. Indeed, each agent who prefers M'' to M must be assigned to an object in M'' that is acceptable to them in I, so it is better than their dummy object. As each agent either obtains the same object in M and M', or is unassigned in M and matched to their dummy object in M', we get that each improving agent prefers M'' to M'. □

5.2.3 Ties in the Preference Lists. From Theorems 5.6 and 5.7 follows that k-HATI and k-HATC are both NP-complete for any constant $0 < c < 1$ and $k = cn$. Therefore, we investigate the preference restrictions DC and STI, as defined in Section 3.3. We provide a complete complexity picture with respect to the parameter c. We first discuss the NP-complete cases, and then we complement those with polynomial algorithms for other values of c.

We start with the preference restriction STI.

Theorem 5.8. *k-HATI is NP-complete even if each agent puts their (at most two) acceptable objects into a single tie. This holds for any $k = cn$ with $0 < c < \frac{1}{2}$.*

Proof. We use the same construction as in Theorem 5.4, on $18\hat{n}$ agents, except that the agents with two objects in their preference lists are indifferent between these objects. Let I be an instance of X3C and I' be the constructed k-HATI instance. We will prove that there is a $(5\hat{n}+1)$-stable matching M in I' if and only if there is an exact 3-cover in I. As the constructions are very similar, only small modifications to the proof of Theorem 5.4 are needed.

Claim 3. *If I admits an exact 3-cover, then I' admits a $(5\hat{n}+1)$-stable matching.*

Proof: Suppose that $S_{l_1}, \ldots, S_{l_{\hat{n}}}$ form an exact 3-cover. Construct a matching M as follows: for each $j \in [3\hat{n}]$, match t_j with p_j. For each $j \in \{l_1, \ldots, l_{\hat{n}}\}$ match s_j^{ℓ} with o_{j_ℓ} for $\ell \in [3]$.

To prove that M is $(5\hat{n}+1)$-stable, it is enough to show that at most $5\hat{n}$ objects are envied, so the improvement graph G_M has at most $5\hat{n}$ objects with non-zero degree. For each $j \in \{l_1, \ldots, l_{\hat{n}}\}$, the object p_j is not envied by any agent: all of s_j^1, s_j^2, s_j^3, and t_j got an object. Hence, there are at most $2\hat{n}$ objects of type p_j and at most $3\hat{n}$ objects of type o_i envied, proving our claim. ■

Claim 4. *If I' admits a $(5\hat{n}+1)$-stable matching, then I admits an exact 3-cover.*

Proof: Let M be a $(5\hat{n}+1)$-stable matching. First, we prove that in M at most $5\hat{n}$ objects are envied by any agent, because we can construct a matching M' that gives all envied objects to an agent who envies them. Each object o_i is envied in any matching by at least one of d_i^1 and d_i^2 and can be given to the envious agent in M'. Regarding envied objects of type p_j, one agent only finds at most one p_j object acceptable, which implies that each envied p_j can be assigned to an envious agent in M'. Therefore, M' is indeed a matching.

As at most $5\hat{n}$ objects can be envied and all o_i objects are envied in M, at most $2\hat{n}$ envied objects are of type p_j. This implies that at least $\hat{n}$ objects of type p_j are not envied by any agent. As these objects are acceptable to their s_j^ℓ, $\ell \in [3]$ agents, those agents must all get their other acceptable object of type o_i. As M is a matching, these sets constitute an exact 3-cover. ■

Finally, we can add sufficiently many instances to I', each of which consists of two agents and one object, with both agents only considering their one object acceptable. In this small instance, there is no cn-stable matching with $c < \frac{1}{2}$, so hardness holds for any such $c > \frac{5}{18}$. Similarly, we can add instances with one agent and one object to prove the hardness for $0 < c < \frac{5}{18}$. □

Our next result for the house allocation model is valid for the preference restriction DC.

Theorem 5.9. *k-HATC with dichotomous preferences is NP-complete*

(1) for any $k = cn$ with $c \leq \frac{1}{2}$ even if $n = |O|$;
(2) if $|O| < n$, then even for any $k = cn$ with $c \leq \frac{2}{3}$.

Proof. We extend the reduction used in the proof of Theorem 5.8. First, we prove the hardness for $k = \frac{5}{18}$, then we pad the instance to cover $0 < c \leq \frac{1}{2}$, and finally, we extend our proof to $c \leq \frac{2}{3}$ with the assumption that $|O| < n$.

We add another $12\hat{n}$ dummy objects to the k-HATI instance and extend the preferences of the agents such that they rank their originally acceptable objects first, and all the other objects—including the $12\hat{n}$ dummy objects—second. Since the acceptability graph is complete, and $n = |O|$, it is clear that if there exists a $(5\hat{n}+1)$-stable matching, then there exists a $(5\hat{n}+1)$-stable matching in which all agents are matched.

We claim that there is a $(5\hat{n}+1)$-stable matching in this modified instance I' if an only if there is a $(5\hat{n}+1)$-stable matching in the original k-HATI instance I.

Indeed, if there is such a matching M in I, then all original objects are matched, and extending M in I' by matching the unmatched agents to the dummy objects arbitrarily produces a matching M' that must be $(5\hat{n}+1)$-stable, as only those agents can improve who got a dummy object, and only by getting a first-choice, hence originally acceptable, object.

If there is a $(5\hat{n}+1)$-stable agent-complete matching in I', then among the agents who are matched to second-choice objects, at most $5\hat{n}$ can improve simultaneously. Specifically, there must be a matching in the graph G' induced by the best object edges for each agent, such that among the unmatched agents at most $5\hat{n}$ can be matched in G'. Such a matching must be $(5\hat{n}+1)$-stable in I.

To prove hardness for all $0 < c \leq \frac{1}{2}$, we either add agents with a dedicated first-ranked object for small c or multiple instances with two agents and two objects, one as their only first-ranked object and the other one ranked second by both of them for larger c. At least half of those agents will always be able to improve even after we make the preferences complete by adding all remaining edges and setting the preferences for the originally unacceptable objects as second.

Finally, suppose that $|O| < n$. We extend the hardness for $\frac{1}{2} \leq c \leq \frac{2}{3}$. For this, we further add sufficiently many copies of an instance with two objects and three agents, such that one object is best, while the other object is second-best for all three agents. Finally, we add the remaining edges as second-ranked edges. Let x be the number of copies we added and let I'' be the new instance. We claim that there is a $\left(5\hat{n} + \frac{2}{3}x + 1\right)$-stable matching in I'' if and only if there is a $(5\hat{n} + 1)$-stable matching in I'. In one direction, if M' is $(5\hat{n} + 1)$-stable in I', then we create M'' by matching two out of the three agents in each added instance to their corresponding two objects. Then, M'' is $\left(5\hat{n} + \frac{2}{3}x + 1\right)$-stable, because at most $\frac{1}{3}x + 5\hat{n}$ agents can improve by getting first-ranked objects and at most $\frac{1}{3}x$ agents can improve by getting matched. In the other direction, if there is a $\left(5\hat{n} + \frac{2}{3}x + 1\right)$-stable matching M'' in I'', then we claim that M'' restricted to I' is $(5\hat{n} + 1)$-stable in I'. Otherwise, more than $5\hat{n} + 1$ agents could improve by getting a first-ranked object among the ones in I', and at least $\frac{x}{3}$ agents could improve by getting a first-ranked object among the newly added agents and there are at least $\frac{x}{3}$ unmatched agents, who all could improve with any object left, which is altogether more than $5\hat{n} + \frac{2}{3}x + 1$, contradiction. □

Remark 5.10. If we add $O(\hat{n}^d)$ new agents to the instances (with a 1-stable matching among them) in the scaling procedures for some constant d, we obtain that all hardness results of the section remain intact even for $k = O(n^{1/d})$.

We complement the above hardness results with positive results for the remaining cases for c.

Theorem 5.11. *If each agent's preference list is a single tie, then the following statements hold.*

(1) For MAJ-HATI*, a majority stable matching exists and can be found in $O(\sqrt{n}m)$ time.*
(2) For MAJ-HATC *with $|O| \geq n$ and dichotomous preferences, a majority stable matching exists and can be found in $O(\sqrt{n}m)$ time.*
(3) For MAJ-HATC *with $|O| < n$, a $\left(\frac{2n}{3} + 1\right)$-stable matching exists and can be found in $O(\sqrt{n}m)$ time.*

Proof. We prove each statement separately.

(1) We claim that any maximum size matching M is majority stable in this case. If $|M| \geq \frac{n}{2}$, then M is obviously majority stable, as at least half of the agents obtain an object. If $|M| < \frac{n}{2}$, then M is also majority stable. Suppose there is a matching M', where more than $\frac{n}{2}$ agents improve. Then $|M'| > \frac{n}{2}$, contradicting the fact that M with $|M| < \frac{n}{2}$ is a maximum size matching.
(2) We first find a maximum size matching in the graph containing only the first-choice edges for each agent. Then we extend this matching by assigning every so far unmatched agent a second-choice object. Note that this is possible as the preferences are dichotomous and complete. In this matching, agents can only improve by switching to a first-choice object from a second-choice object. This cannot be the case for a majority of the agents, because we started with a maximum size matching in the graph containing only the first-choice edges for each agent. So the constructed matching is majority stable.
(3) If $|O| \leq \frac{2}{3}n$, then the statement is obvious. Otherwise, let M be a maximum matching that matches as many agents to a first-choice object as possible—constructed the same way as in the previous case. If M matches at least $\frac{n}{3}$ agents to first-choice objects, then at most $\frac{2}{3}n$ agents can improve, so M is $\left(\frac{2}{3}n + 1\right)$-stable. Otherwise, let $x < \frac{n}{3}$ denote the number of agents who get a first-choice object in M and y denote the number of agents who get a second-choice object. Clearly, at most $x + (n - x - y) = n - y$ agents can improve in a different

matching. Hence, if M is not $(\frac{2}{3}n + 1)$-stable, then $y < \frac{n}{3}$ and $x < \frac{n}{3}$, contradicting that we assumed that the number of objects $x + y$ is at least $\frac{2}{3}n$. □

The first two statements and the fact that k-stable matchings are also $k + 1$-stable imply the following.

Corollary 5.12. *k-HATI is solvable in $O(\sqrt{n}m)$ time for any $k = cn$ with $c \geq \frac{1}{2}$ when each preference list is a single tie. The same holds for k-HATC with dichotomous preferences and $c \geq \frac{1}{2}$ in the case of $|O| \geq n$; and $c \geq \frac{2}{3}$ in the case of $|O| < n$.*

6 The Marriage and Roommates Models

In this section, we settle most complexity questions in the marriage and roommates models for $k = cn, 0 < c < 1$. Just as in Section 5, we first prove that verification can be done in polynomial time in Section 6.1 and then turn to the existence problems in Section 6.2.

6.1 Verification

Theorem 6.1. *Verifying whether a matching is k-stable can be done in $O(n^3)$ time, both in the marriage and roommates models, even if the preference lists contain ties.*

Proof. Let M be a matching in a k-SRI instance. We create an edge weight function ω, where $\omega(e)$ is the number of end vertices of e who prefer e to M. From the definition of k-stability follows that M is k-stable if and only if maximum weight matchings in this graph have weight less than k. Such a matching can be computed in $O(n^3)$ time [27]. □

6.2 Existence

Similar to the HA model, it turns out that the computational complexity with respect to the parameter c does not depend on whether ties or incomplete preferences are allowed or not (assuming we have no additional restrictions like STI or DC). Hence, we first settle the complexity of the general problems for the marriage and roommates models in Section 6.2.1 and then for the restricted variants STI and DC in Section 6.2.2

6.2.1 General Preference Lists. We start by providing our algorithm for the most general case, that is, the roommates model with ties and incomplete preference lists. We show that there always exists a $\left(\frac{5}{6}n + 1\right)$-stable matching and we can find one in $\mathcal{O}(m)$ time.

To show that a $\left(\frac{5}{6}n + 1\right)$-stable matching exists even in the roommates model with ties, we first introduce the concept of stable partitions, which generalizes the notion of a stable matching. A stable partition defines a set of edges and cycles. More formally, let $(N, \succ)$ be a stable roommates instance. A *stable partition* of $(N, \succ)$ is a permutation $\pi : N \longrightarrow N$ with $(i, \pi(i)) \in E\ \forall i \in N$ such that for each $i \in N$:

(1) if $\pi(i) \neq \pi^{-1}(i)$, then $\pi(i) \succ_i \pi^{-1}(i)$, i.e., the preferences in the cycles of π are cyclic;
(2) for each $(i, j) \in E$, if $\pi(i) = i$ or $j \succ_i \pi^{-1}(i)$, then $\pi^{-1}(j) \succ_j i$, i.e., there are no edges strictly preferred by both endpoints to some their partners (if there is any) in π.

One may also think of stable partitions as half-integral stable matchings, i.e., a function $\mu : E \longrightarrow \{0, \frac{1}{2}, 1\}$, such that for each edge $(i, j) \in E$ with $\mu(i, j) < 1$, we have that either $\sum_{j'} \mu(i, j') = 1$ and $j' \succeq_i j$ for all $j' \in \mu(i) = \{j' : \mu(i, j') > 0\}$ or $\sum_{i'} \mu(i', j) = 1$ and $i' \succeq j$ for all $i' \in \mu(j)$. It is easy to see that a stable partition π corresponds to a half-integral stable matching by setting $\mu(i, j) = 1$ if $\pi(i) = j, \pi(j) = i$, $\mu(i, j) = \frac{1}{2}$, if exactly one of $\pi(i) = j$ and $\pi(j) = i$ holds, and $\mu(i, j) = 0$ otherwise.

1:	2 ≻	6 ≻	4 ≻	3 ≻	5
2:	3 ≻	5 ≻	1 ≻	6 ≻	4
3:	1 ≻	6 ≻	2 ≻	5 ≻	4
4:	5 ≻	2 ≻	3 ≻	6 ≻	1
5:	6 ≻	1 ≻	3 ≻	4 ≻	2
6:	4 ≻	2 ≻	5 ≻	1 ≻	3

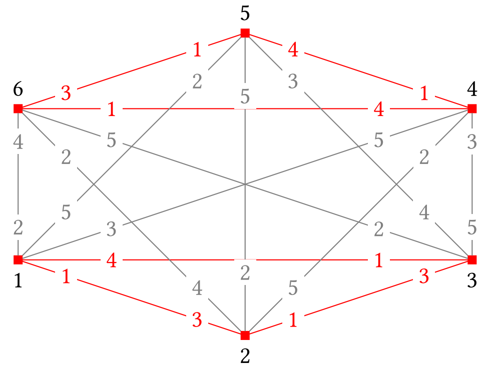

Fig. 3. A 6-agent instance given with its strict preferences on the left, and in the graph representation form on the right. The marked red entries and edges constitute the stable partition $\pi = (2, 3, 1, 5, 6, 4)$.

Reversely, it is not hard to see that in a half-integral stable matching, the fractional edges form vertex-disjoint cycles with cyclic preferences, so the matching corresponds to a stable partition π.

Tan [55] showed that a stable partition always exists and that one can be found in polynomial time. A slightly simpler approach has been given by Tan and Yuang-Cheh [56]. The main idea of the algorithm is to start from a single vertex, and then add the vertices one by one and always maintain a stable partition of the current graph with a procedure similar to the classic Gale–Shapley algorithm. Tan [55] also proved that a stable matching exists if and only if a stable partition does not contain any odd cycle. An example instance [38] with a stable partition consisting of two odd cycles is depicted in Figure 3.

Theorem 6.2. *For any $k = cn$ with $c > \frac{5}{6}$, a k-stable matching exists in k-SRI and can be found in $O(m)$ time.*

Proof. We present an algorithm to construct such a matching. We apply Tan's algorithm to obtain a stable partition. Then, for each odd cycle of length at least three, we remove an arbitrary vertex from the cycle. This leaves us with components that are either even cycles, or paths on an even number of vertices, or single vertices, as the odd cycles become paths on an even number of vertices after the removal of a vertex. In all these components except the single vertices, we choose a perfect matching.

The running time of Tan's algorithm [55] is $O(m)$. Besides this, our algorithm only has to go through each cycle to delete a vertex if it is odd and then choose a perfect matching in the cycle/-path. Therefore, the whole running time is $O(m)$ too.

Denote the matching obtained in this way by M. Clearly, M has at most $\frac{n}{2}$ edges. We claim that M is $\left(\frac{5}{6}n + 1\right)$-stable. Let M' be any matching. Observe that if an edge $e \in M'$ has the property that both of its end vertices prefer it to M, then e must be adjacent to one of the deleted vertices. This is because otherwise, both end vertices of e would prefer e to their worst partner (if there is any) in the stable partition, but this is a contradiction to it being a stable partition in the first place. Also, observe that the number of deleted vertices is at most the number of odd cycles, which is at most $\frac{n}{3}$. Combining these, we get that the number of agents who prefer M' to M is at most $\frac{n}{3} \cdot 2 + (\frac{n}{2} - \frac{n}{3}) = \frac{5}{6}n$.

Hence, M is $\left(\frac{5}{6}n + 1\right)$-stable, which proves our statement. □

As k-SRC, k-SMI, and k-SMC are subcases of k-SRI, the existence of a k-stable matching follows from Theorem 6.2.

Next, we show that in the marriage model, even majority stable matchings are guaranteed to exist.

Theorem 6.3. *In the marriage model, a majority stable matching exists and can be found in* $O(m)$ *time. Thus, for any* $k = cn$ *with* $c \geq \frac{1}{2}$*, a* k*-stable matching exists and can be found in* $O(m)$ *time.*

Proof. Even in the presence of ties, a popular matching can be found in $O(m)$ time [10, 30]. As each popular matching must be majority stable, the statement follows. □

As majority stable matchings always exist in the marriage model, it is natural to ask whether we can find a majority stable matching that is a maximum size matching as well. We denote the problem of deciding if such a matching exists by MAX-MAJ-SMI. In the case of complete preferences, a maximum size and majority stable matching exists and can be found efficiently, as popular matchings are both majority stable and maximum size. Otherwise, the situation is less preferable, as the following theorem shows, which also settles the complexity of k-SMC, k-SMI, k-SRC and k-SRI.

Theorem 6.4. *The following problems are NP-complete:*

(1) MAX-MAJ-SMI even if $|U| = |W|$*;*
(2) k*-SMC and* k*-SMI for any* $k = cn$ *with* $c < \frac{1}{2}$ *even if* $|U| = |W|$*;*
(3) k*-SRC and* k*-SRI for any* $k = cn$ *with* $c < \frac{2}{3}$*.*

Proof. By Theorem 6.1, all decision problems on the existence of a k-stable matching in the marriage and roommates models are in NP. We again reduce from X3C. Let I be an instance of X3C and let $\mathcal{S} = \{S_1, \ldots, S_{3\hat{n}}\}$. First, we create an instance I' of SM, which admits a $(16\hat{n} + 1)$-stable matching if and only if it admits a maximum size $(16\hat{n} + 1)$-stable matching if and only if I admits an exact 3-cover. This instance will be the basis of all three reductions.

Let us denote the two classes of the agents we create in I' by U and W. For each set $S_j \in S$ we create five agents $s_j^1, s_j^2, s_j^3, y_j, q_j$ in U, and five agents $c_j^1, c_j^2, c_j^3, x_j, p_j$ in W. For each element $i \in \mathcal{X}$ we create five agents $b_i, d_i, e_i, f_i', g_i'$ in U and five agents $a_i, d_i', e_i', f_i, g_i$ in W. Altogether we have $n = 60\hat{n}$ agents.

The preferences are described in Figure 4. Let $S_j = \{j_1, j_2, j_3\}$ and for an element i, let $S_{i_1}, S_{i_2}, S_{i_3}$ be the three sets that contain i, with ℓ_1, ℓ_2, ℓ_3 denoting the position of i in the sets $S_{i_1}, S_{i_2}, S_{i_3}$, respectively.

Claim 5. *If* I *admits an exact 3-cover, then* I' *admits a maximum size* $(16\hat{n} + 1)$*-stable matching.*

Proof: To the exact 3-cover $S_{l_1}, \ldots, S_{l_{\hat{n}}}$ in I we create a matching M in I' as follows—see Figure 4 as well.

- For each $i, j \in [3\hat{n}]$, we add the edges $(y_j, p_j), (d_i, d_i'), (e_i, e_i'), (q_j, x_j), (f_i', f_i), (g_i', g_i)$.
- For each $j \in \{l_1, \ldots, l_{\hat{n}}\}$ and $\ell \in [3]$, we add the edges $(s_j^\ell, a_{j_\ell}), (b_{j_\ell}, c_j^\ell)$.
- For each $j \notin \{l_1, \ldots, l_{\hat{n}}\}$ and $\ell \in [3]$, we add the edge (s_j^ℓ, c_j^ℓ).

As the sets formed an exact cover, M is a matching, and because all agents are matched, M is a maximum size matching in I'. We next show that in any matching M', at most $16\hat{n}$ agents can improve.

- Agents of type s_j^ℓ, c_j^ℓ can improve with an agent of type p_j or q_j, respectively, if $(s_j^\ell, c_j^\ell) \in M$. As $S_{l_1}, \ldots, S_{l_{\hat{n}}}$ was an exact cover, at most $2\hat{n}$ of the s_j^ℓ agents can improve with an agent of type p_j—only those with $j \notin \{l_1, \ldots, l_{\hat{n}}\}$—and similarly, at most $2\hat{n}$ of the c_j^ℓ agents can improve with an agent of type q_j.
- Agents of type a_i or b_i can improve.

$s_j^\ell:$	$a_{j_\ell} \succ p_j \succ c_j^\ell$	$c_j^\ell:$	$b_{j_\ell} \succ q_j \succ s_j^\ell$
$q_j:$	$x_j \succ c_j^1 \succ c_j^2 \succ c_j^3$	$p_j:$	$y_j \succ s_j^1 \succ s_j^2 \succ s_j^3$
$y_j:$	p_j	$x_j:$	q_j
$b_i:$	$f_i \succ c_{i_1}^{\ell_1} \succ c_{i_2}^{\ell_2} \succ c_{i_3}^{\ell_3} \succ g_i$	$a_i:$	$d_i \succ s_{i_1}^{\ell_1} \succ s_{i_2}^{\ell_2} \succ s_{i_3}^{\ell_3} \succ e_i$
$d_i:$	$a_i \succ d_i'$	$f_i:$	$b_i \succ f_i'$
$e_i:$	$a_i \succ e_i'$	$g_i:$	$b_i \succ g_i'$
$f_i':$	f_i	$d_i':$	d_i
$g_i':$	g_i	$e_i':$	e_i

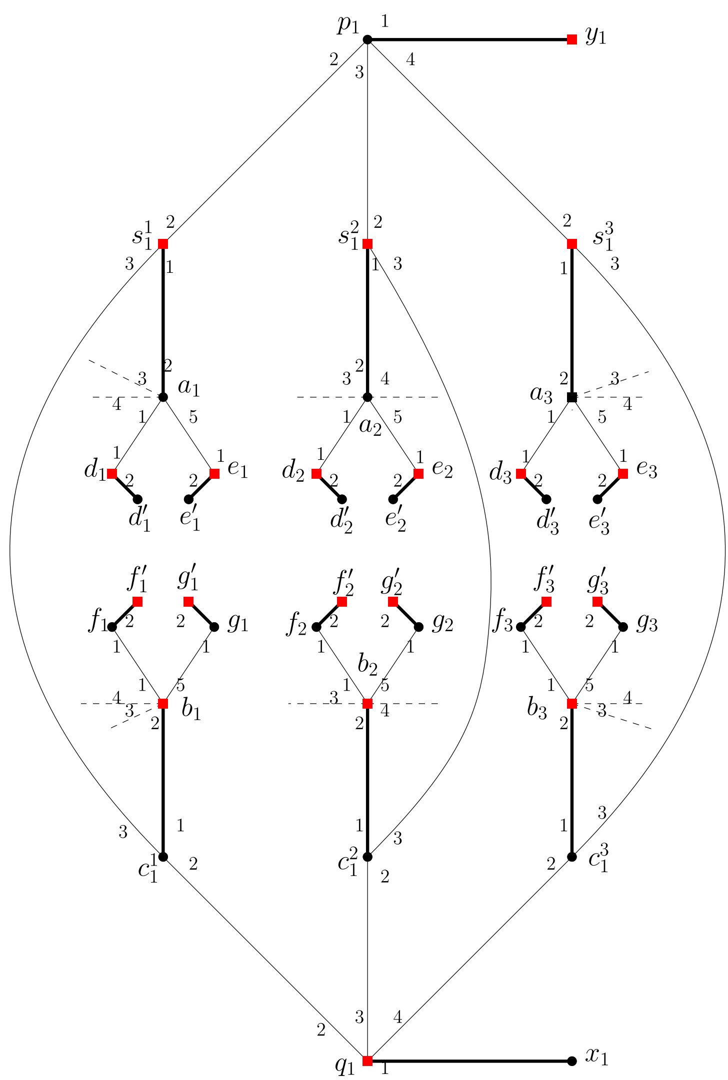

Fig. 4. The construction for Theorem 6.4. The preference list of each created agent, $i, j \in [3\hat{n}]$, $\ell \in [3]$, can be seen above the graph. The figure illustrates the gadget of a set $S_1 = \{1, 2, 3\}$. The numbers on the edges indicate the preferences. Thick edges denote the matching edges if S_1 is in the exact-3-cover, dashed edges run to other gadgets.

— All other agents can only improve by switching to an agent of type a_i or b_i. Even if all agents of type a_i, b_i improve and they get a partner who also improves with them, $2 \cdot 2 \cdot 3\hat{n} = 12\hat{n}$ agents can improve as or through agents of type a_i or b_i.

Therefore, altogether at most $4\hat{n} + 12\hat{n} = 16\hat{n}$ agents can improve, concluding the proof of our claim. ■

Claim 6. *If I' admits a maximum size $(16\hat{n}+1)$-stable matching, then I' admits a $(16\hat{n}+1)$-stable matching.*

Proof: Trivial. ■

Claim 7. *If I' admits a $(16\hat{n}+1)$-stable matching, then I admits an exact 3-cover.*

Proof: Suppose that I' admits a $(16\hat{n}+1)$-stable matching M, but there is no exact 3-cover in I. We count the number of agents who can improve.

— For any i, if a_i is not matched to d_i in M, then both a_i and d_i can improve if they get matched together, and otherwise both d_i' and e_i can improve with the edges $(d_i, d_i'), (e_i, a_i)$. Therefore, in any M, at least $6\hat{n}$ agents from the set $\{d_i', d_i, e_i, a_i \mid i \in [3\hat{n}]\}$ can improve among themselves and similarly, at least $6\hat{n}$ agents from $\{f_i', f_i, g_i, b_i \mid i \in [3\hat{n}]\}$ can improve among themselves.

— As no exact 3-cover exists in I, there are at least $2\hat{n}+1$ indices $j \in [3\hat{n}]$, such that at least one of $\{s_j^1, s_j^2, s_j^3\}$ is not matched to an agent of type a_i in M and similarly, there are at least $2\hat{n}+1$ indices $j' \in [3\hat{n}]$, such that at least one of $\{c_{j'}^1, c_{j'}^2, c_{j'}^3\}$ is not matched to an agent of type b_i in M. For each such j and j', at least one of the agents in the set $\{s_j^1, s_j^2, s_j^3, y_j, p_j\}$ and at least one of the agents in $\{c_{j'}^1, c_{j'}^2, c_{j'}^3, x_{j'}, q_{j'}\}$ can improve among themselves: if $(y_j, p_j) \notin M$ or $(q_{j'}, x_{j'}) \notin M$, then y_j and p_j or $x_{j'}$ and $q_{j'}$ can both improve with each other, otherwise there is an s_j^ℓ agent with a c_j^ℓ agent or a $c_{j'}^\ell$ agent with an $s_{j'}^\ell$ agent, who could improve with p_j or $q_{j'}$, respectively.

Altogether at least $12\hat{n}+4\hat{n}+2 = 16\hat{n}+2$ agents can improve, a contradiction to the $(16\hat{n}+1)$-stability of M. ■

Now we use this construction to prove the hardness of all three problems.

(1) For max-maj-smi, we add a path P with $28\hat{n}+2$ vertices to I', to have $88\hat{n}+2$ agents. Path P has an even number of vertices and therefore a unique maximum size matching M_P. However, we create the preferences of the agents such that they all prefer their other edge in P, except for the end vertices. Hence, by switching to the edges in P outside of M_P, $28\hat{n}$ agents improve. So in this instance, there is a maximum size majority stable matching, and, therefore, a $(44\hat{n}+2)$-stable matching if and only if I' admits a matching, where at most $16\hat{n}+1$ agents can improve, which happens if and only if I admits an exact 3-cover, as we have seen in the main part of the proof.

(2) We distinguish two cases for k-smc. First, let $\frac{4}{15} \leq c < \frac{1}{2}$. In this case, we add to I' some paths on four vertices, such that the vertices with degree two prefer the middle edge. Hence, for any matching M, in any of these paths, at least two agents can improve. By adding enough of these paths, we can get an instance that has a k-stable matching for $k = cn$ if and only if there is a $(16\hat{n}+1)$-stable matching in I'.
For $c < \frac{4}{15}$, we add a sufficiently large instance that admits a 1-stable matching—a union of edges suffices.

(3) We apply a similar case distinction for the last statement on k-src. First, let $\frac{4}{15} \leq c < \frac{2}{3}$. Now, we add to I' some (but an even number of) triangles with cyclic preferences. For any matching M, in any such triangle, at least two out of the three agents can improve. This still

holds if we add edges between the vertices of the first two, the vertices of the second two, . . ., the vertices of the last two triangles, which are worse for both sides. This leads to an instance in which there is a complete k-stable matching if there is any k-stable matching. Hence, by adding sufficiently many of these triangles, we can get an instance that has a k-stable matching for $k = cn$ if and only if there is a $(16\hat{n} + 1)$-stable matching in I'.

For $c < \frac{4}{15}$, we add a sufficiently large instance that admits a 1-stable matching.

The second and the third reductions remain intact for complete preferences when we add the remaining agents to the end of the preference lists. It is easy to see that if there is a k-stable matching M' with the extended preferences, then there is one where each agent obtains an original partner—we just project M' to the original acceptability graph to get M. If there is a matching M'', where at least k agents improve from M, then these k agents must also prefer M'' to M', which contradicts the k-stability of M'.

In the other direction, we have by our reduction that if there is a k-stable matching M with incomplete preferences, then there is one that is complete as well. Hence, agents can only improve with original edges, so if a matching M'' would be better for at least k agents than M with complete preferences, then so would it be with incomplete preferences as well, which is a contradiction. □

6.2.2 Restricted Preferences. We now turn to the preference-restricted cases of STI and DC. We start with our results for STI, and then we discuss our results for DC.

THEOREM 6.5. *If each preference list consists of a single tie, then a k-stable matching can be found $O(\sqrt{n}m)$ time for any $k = cn, c > \frac{1}{3}$.*

PROOF. Let M be a maximum size matching in the acceptability graph $G = (N, E)$, which can be found in $O(\sqrt{n}m)$ time [47]. We claim that M is $\left(\frac{n}{3} + 1\right)$-stable. Suppose there is a matching M', where at least $\frac{n}{3} + 1$ agents improve. As the preferences consist of a single tie, this is only possible if all of them were unmatched in M and they can be matched simultaneously. By the famous Gallai–Edmonds decomposition, we know that there is a set $X \subset N$ such that

— each maximum size matching matches every vertex of X;
— every vertex in X is matched to a vertex in distinct odd components in $G \setminus X$;
— has exactly $q(X) - |X|$ unmatched vertices, each of which are in distinct odd components in $G \setminus X$, where $q(X)$ denotes the number of odd components in $G \setminus X$.

As $\frac{n}{3} + 1$ agents can improve, we get that $q(X) - |X| > \frac{n}{3}$. Let s denote the number of singleton components in $G \setminus X$. If such a vertex was unmatched in M, then it can only improve by getting matched to someone in X. Let $x \leq s$ be the number of agents in singleton components who improve in M'. Then, $|X| \geq x$ by our observation. Also, at least $\frac{n}{3} - x + 1$ agents must improve from the other odd components, which all have a size of at least 3. Furthermore, there must be at least x odd components, whose vertices are all matched by M according to the Gallai–Edmonds characterization. Hence, we get that the number of vertices of G is at least $x + 3 \cdot (\frac{n}{3} - x + 1) + 2 \cdot x \geq n + 3$, which is a contradiction. □

THEOREM 6.6. *For $c \leq \frac{1}{3}$ and $k = cn$, k-SMI is NP-complete even if each preference list is a single tie.*

PROOF. We start with the reduction in the proof of Theorem 5.8. There, one side of the graph consisted of objects, which now correspond to agents. To ensure that each agent's list is a single tie, all acceptable agents are tied in the preference lists of these new agents.

In the proof of Theorem 5.8, we showed that in the base instance with $18\hat{n}$ agents and $6\hat{n}$ objects it is NP-complete to decide if there exists a $(5\hat{n} + 1)$-stable matching. As the reduction had the property that any inclusion-wise maximal matching assigned all objects, it follows that in our new instance with two-sided preferences there is a $(5\hat{n} + 1)$-stable matching if and only if there was one with one-sided preferences, as we can always suppose that the agents on the smaller side cannot improve.

To extend hardness to $c \leq \frac{5}{21}$, we just add a sufficiently large instance that admits a 1-stable matching. To extend hardness to any $\frac{5}{21} < c \leq \frac{1}{3}$, we add paths with 3 vertices to the instance. As in each such path, there is always an unmatched agent, at least one third of these agents can improve, hence by adding sufficiently many copies, the theorem follows. □

Now we move on to complete and dichotomous (DC) preferences.

THEOREM 6.7. *If the preferences are complete and dichotomous, then a k-stable matching exists and can be found in $O(n^3)$ time for any $k = cn, c > \frac{1}{2}$.*

PROOF. We define the edge weight function $w(e)$ as the number of end vertices of e who rank e best. For a matching M, $w(M)$ is equal to the number of agents who get a first choice. Let M be a maximum weight matching with respect to $w(e)$. Then we extend M to a maximum size matching, which matches all agents as the preferences are complete. Suppose there is a matching M', where more than $\frac{n}{2}$ agents improve. As agents can only improve by getting a first-choice partner, we get that $w(M') > \frac{n}{2} > w(M)$, contradiction. □

THEOREM 6.8. *For $c \leq \frac{1}{2}$ and $k = cn$, k-SMTC is NP-complete even if the preferences are dichotomous.*

PROOF. We extend the instance in Theorem 6.6—which had $3\hat{n}$ agents on one side and $18\hat{n}$ on the other side—by completing the acceptability graph via adding the remaining edges such that they are ranked second for each end vertex. It is clear that deciding if there is a $(5\hat{n} + 1)$-stable matching remains NP-complete.

To show hardness for smaller c values, we pad the instance by adding pairs of agents who rank each other first, and all other agents second. For $\frac{5}{21} < c \leq \frac{1}{2}$, we first add paths with 4 vertices, such that the middle two vertices only rank each other first, while the end vertices rank their only neighbor first. In this small instance, at least half of the agents can always improve and this fact remains true even after making the acceptability graph complete by adding the remaining edges as second best for all agents. □

Remark 6.9. If we add $O(\hat{n}^d)$ new agents to the instances in the padding procedures for some arbitrary constant d, then we obtain that all hardness results of the Section remain even for $k = n^{1/d}$.

7 Conclusion and Open Questions

We have settled the main complexity questions on the verification and existence of k-stable and majority stable matchings in all three major matching models. We derived that the existence of a k-stable solution is the easiest to guarantee in the marriage model, while it cannot be guaranteed for any non-trivial k at all in the house allocation model. Only one case remains partially open: in the roommates model, the existence of a k-stable solution is guaranteed for $k = cn, c \geq \frac{5}{6}$ (Theorem 6.2), whereas NP-completeness was only proved for $c < \frac{2}{3}$ (Theorem 6.4, point 3). We conjecture polynomial solvability for $\frac{2}{3} \leq c < \frac{5}{6}$ as we were unable to find even a single no-instance despite substantial effort.

Another interesting open problem is to answer the complexity for small, sublinear k between $O(1)$ and $O(n^{1/d})$.

A straightforward direction of further research would be to study the strategic behavior of the agents. It is easy to prove that k-stability, as stability and popularity, is fundamentally incompatible with strategyproofness. However, mechanisms that guarantee strategyproofness for a subset of agents might be developed. Another rather game-theoretic direction would be to investigate the price of k-stability.

A much more applied line of research involves computing the smallest k for which implemented solutions of real-life matching problems are k-stable. For example: given a college admission pool and its stable outcome, how many of the students could have gotten into a better college in another matching? With our terminology, how large is the maximum matching in the improvement graph G_M belonging to the calculated solution M? We conjecture that the implemented solution can only be improved for a little fraction of the applicants simultaneously. Simulations supporting this could potentially strengthen the trust in the system.

Acknowledgments

The authors thank Barton E. Lee for drawing their attention to the concept of majority stability and for producing thought-provoking example instances.

References

[1] David J. Abraham, Katarína Cechlárová, David F. Manlove, and Kurt Mehlhorn. 2004. Pareto optimality in house allocation problems. In *Proceedings of the 15th International Symposium on Algorithms and Computation, ISAAC 2004*. Springer, 3–15.

[2] David J. Abraham, Robert W. Irving, Telikepalli Kavitha, and Kurt Mehlhorn. 2007. Popular matchings. *SIAM Journal on Computing* 37, 4 (2007), 1030–1045.

[3] Haris Aziz, Felix Brandt, and Paul Harrenstein. 2013. Pareto optimality in coalition formation. *Games and Economic Behavior* 82 (2013), 562–581. DOI: https://doi.org/10.1016/j.geb.2013.08.006

[4] Haris Aziz, Jiayin Chen, Serge Gaspers, and Zhaohong Sun. 2018. Stability and Pareto optimality in refugee allocation matchings. In *Proceedings of the AAMAS'18*. 964–972.

[5] Haris Aziz, Gergely Csáji, and Ágnes Cseh. 2023. Computational complexity of k-stable matchings. In *Algorithmic Game Theory*. Argyrios Deligkas and Aris Filos-Ratsikas (Eds.), Springer Nature Switzerland, Cham, 311–328.

[6] Michel Balinski and Tayfun Sönmez. 1999. A tale of two mechanisms: Student placement. *Journal of Economic Theory* 84, 1 (1999), 73–94.

[7] Alkida Balliu, Michele Flammini, Giovanna Melideo, and Dennis Olivetti. 2022. On pareto optimality in social distance games. *Artificial Intelligence* 312 (2022), 103768. DOI: https://doi.org/10.1016/j.artint.2022.103768

[8] Sayan Bhattacharya, Martin Hoefer, Chien-Chung Huang, Telikepalli Kavitha, and Lisa Wagner. 2015. Maintaining near-popular matchings. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*. Springer, 504–515.

[9] Péter Biró and Jens Gudmundsson. 2021. Complexity of finding Pareto-efficient allocations of highest welfare. *European Journal of Operational Research* 291, 2 (2021), 614–628.

[10] Péter Biró, Robert W. Irving, and David F. Manlove. 2010. Popular matchings in the marriage and roommates problems. In *CIAC '10: Proceedings of the 7th International Conference on Algorithms and Complexity (Lecture Notes in Computer Science, Vol. 6078)*. Springer, 97–108.

[11] Anna Bogomolnaia and Hervé Moulin. 2004. Random matching under dichotomous preferences. *Econometrica* 72, 1 (2004), 257–279.

[12] Sebastian Braun, Nadja Dwenger, and Dorothea Kübler. 2010. Telling the truth may not pay off: An empirical study of centralized university admissions in Germany. *The B.E. Journal of Economic Analysis and Policy* 10, Article 22 (2010). https://www.degruyter.com/document/doi/10.2202/1935-1682.2294/html#MLA

[13] Martin Bullinger. 2020. Pareto-optimality in cardinal hedonic games. In *Proceedings of the AAMAS'20*. 213–221.

[14] Katarína Cechlárová, Pavlos Eirinakis, Tamás Fleiner, Dimitrios Magos, David Manlove, Ioannis Mourtos, Eva Oceláková, and Baharak Rastegari. 2016. Pareto optimal matchings in many-to-many markets with ties. *Theory of Computing Systems* 59, 4 (2016), 700–721.

[15] Katarína Cechlárová, Pavlos Eirinakis, Tamás Fleiner, Dimitrios Magos, Ioannis Mourtos, and Eva Potpinková. 2014. Pareto optimality in many-to-many matching problems. *Discrete Optimization* 14 (2014), 160–169. DOI: https://doi.org/10.1016/j.disopt.2014.09.002

[16] Yan Chen and Tayfun Sönmez. 2002. Improving efficiency of on-campus housing: An experimental study. *American Economic Review* 92, 5 (2002), 1669–1686.

[17] Marie Jean Antoine Nicolas de Caritat, Marquis de Condorcet. 1785. *Essai Sur L'application de L'analyse à La Probabilité des Décisions Rendues à la Pluralité Des Voix*. L'Imprimerie Royale.

[18] Ágnes Cseh. 2017. Popular matchings. *Trends in Computational Social Choice* 105, 3 (2017).

[19] Ágnes Cseh, Chien-Chung Huang, and Telikepalli Kavitha. 2017. Popular matchings with two-sided preferences and one-sided ties. *SIAM Journal on Discrete Mathematics* 31, 4 (2017), 2348–2377.

[20] Ágnes Cseh and Telikepalli Kavitha. 2021. Popular matchings in complete graphs. *Algorithmica* 83, 5 (2021), 1493–1523.

[21] Andreas Darmann. 2013. Popular spanning trees. *International Journal of Foundations of Computer Science* 24, 05 (2013), 655–677.

[22] Andreas Darmann. 2018. A social choice approach to ordinal group activity selection. *Mathematical Social Sciences* 93 (2018), 57–66. DOI: https://doi.org/10.1016/j.mathsocsci.2018.01.005

[23] Matthias Ehrgott and Stefan Nickel. 2002. On the number of criteria needed to decide Pareto optimality. *Mathematical Methods of Operations Research* 55, 3 (2002), 329–345.

[24] Edith Elkind, Angelo Fanelli, and Michele Flammini. 2020. Price of Pareto optimality in hedonic games. *Artificial Intelligence* 288 (2020), 103357. DOI: https://doi.org/10.1016/j.artint.2020.103357

[25] Yuri Faenza, Telikepalli Kavitha, Vladlena Powers, and Xingyu Zhang. 2019. Popular matchings and limits to tractability. In *Proceedings of SODA '19: the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM-SIAM, 2790–2809.

[26] Monique Florenzano, Pascal Gourdel, and Alejandro Jofré. 2006. Supporting weakly pareto optimal allocations in infinite dimensional nonconvex economies. *Economic Theory* 29, 3 (2006), 549–564. Retrieved December 21, 2024 from http://www.jstor.org/stable/25056127

[27] Harold N. Gabow. 1976. An efficient implementation of Edmonds algorithm for maximum matching on graphs. *Journal of the ACM* 23, 2 (1976), 221–234.

[28] Anh-Tuan Gai, Dmitry Lebedev, Fabien Mathieu, Fabien De Montgolfier, Julien Reynier, and Laurent Viennot. 2007. Acyclic preference systems in P2P networks. In *Euro-Par '07: Proceedings of the 13th International Euro-Par Conference (European Conference on Parallel and Distributed Computing).* Anne-Marie Kermarrec, Luc Bougé, and Thierry Priol (Eds.), Lecture Notes in Computer Science, Vol. 4641, Springer, 825–834.

[29] David Gale and Lloyd S. Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15.

[30] Peter Gärdenfors. 1975. Match making: Assignments based on bilateral preferences. *Behavioral Science* 20, 3 (1975), 166–173.

[31] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability*. Freeman, San Francisco, CA.

[32] Sushmita Gupta, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. 2021. Popular matching in roommates setting is NP-hard. *ACM Transactions on Computation Theory* 13, 2, Article 9 (2021), 20 pages.

[33] John E. Hopcroft and Richard M. Karp. 1973. A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing* 2, 4 (1973), 225–231.

[34] Chien-Chung Huang and Telikepalli Kavitha. 2013. Near-popular matchings in the roommates problem. *SIAM Journal on Discrete Mathematics* 27, 1 (2013), 43–62.

[35] Chien-Chung Huang and Telikepalli Kavitha. 2013. Popular matchings in the stable marriage problem. *Information and Computation* 222 (2013), 180–194. DOI: https://doi.org/10.1016/j.ic.2012.10.012

[36] Chien-Chung Huang and Telikepalli Kavitha. 2021. Popularity, mixed matchings, and self-duality. *Mathematics of Operations Research* 46, 2 (2021), 405–427.

[37] Chien-Chung Huang, Telikepalli Kavitha, Dimitrios Michail, and Meghana Nasre. 2011. Bounded unpopularity matchings. *Algorithmica* 61, 3 (2011), 738–757.

[38] Robert W. Irving. 1985. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms* 6, 4 (1985), 577–595.

[39] Telikepalli Kavitha. 2014. A size-popularity tradeoff in the stable marriage problem. *SIAM Journal on Computing* 43, 1 (2014), 52–71.

[40] Telikepalli Kavitha, Tamás Király, Jannik Matuschke, Ildikó Schlotter, and Ulrike Schmidt-Kraepelin. 2022. The popular assignment problem: when cardinality is more important than popularity. In *SODA '22: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 103–123.

[41] Telikepalli Kavitha, Tamás Király, Jannik Matuschke, Ildikó Schlotter, and Ulrike Schmidt-Kraepelin. 2022. Popular branchings and their dual certificates. *Mathematical Programming* 192, 1 (2022), 567–595.

[42] Telikepalli Kavitha, Julián Mestre, and Meghana Nasre. 2011. Popular mixed matchings. *Theoretical Computer Science* 412, 24 (2011), 2679–2690.

[43] Sonja Kraiczy, Ágnes Cseh, and David Manlove. 2023. On weakly and strongly popular rankings. *Discrete Applied Mathematics* 340 (2023), 134–152.

[44] Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97.

[45] David F. Manlove. 2013. *Algorithmics of Matching Under Preferences.* World Scientific.

[46] Richard Matthew McCutchen. 2008. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In *Proceedings of LATIN '08: the 8th Latin-American Theoretical Informatics Symposium.* EduardoSany Laber, Claudson Bornstein, LoanaTito Nogueira, and Luerbio Faria (Eds.), Lecture Notes in Computer Science, Vol. 4957, Springer, Berlin, 593–604.

[47] Silvio Micali and Vijay V. Vazirani. 1980. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In *Proceedings of FOCS '80: the 21st Annual IEEE Symposium on Foundations of Computer Science.* IEEE Computer Society, 17–27.

[48] William B. T. Mock. 2011. *Pareto Optimality.* Springer Netherlands, Dordrecht. DOI: https://doi.org/10.1007/978-1-4020-9160-5_341

[49] Nitsan Perach, Julia Polak, and Uriel G. Rothblum. 2008. A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the Technion. *International Journal of Game Theory* 36 (2008), 519–535. https://link.springer.com/article/10.1007/s00182-007-0083-4#citeas

[50] Dominik Peters. 2016. Complexity of hedonic games with dichotomous preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[51] Alvin E. Roth. 1982. Incentive compatibility in a market with indivisible goods. *Economics Letters* 9, 2 (1982), 127–132.

[52] Alvin E. Roth and Marilda A. Oliveira Sotomayor. 1990. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis.* Econometric Society Monographs. Cambridge University Press.

[53] Suthee Ruangwises and Toshiya Itoh. 2021. Unpopularity factor in the marriage and roommates problems. *Theory of Computing Systems* 65, 3 (2021), 579–592.

[54] Alexander Schrijver. 2003. Combinatorial optimization: polyhedra and efficiency. Springer.

[55] Jimmy J. M. Tan. 1991. A necessary and sufficient condition for the existence of a complete stable matching. *Journal of Algorithms* 12, 1 (1991), 154–178.

[56] Jimmy J. M. Tan and Hsueh Yuang-Cheh. 1995. A generalization of the stable matching problem. *Discrete Applied Mathematics* 59, 1 (1995), 87–102.

[57] Ashutosh Thakur. 2021. Combining social choice and matching theory to understand institutional stability. In *Proceedings of the 25th Annual ISNIE/SIOE Conference.* Society for Institutional and Organizational Economics.

[58] Anke van Zuylen, Frans Schalekamp, and David P. Williamson. 2014. Popular ranking. *Discrete Applied Mathematics* 165 (2014), 312–316. DOI: https://doi.org/10.1016/j.dam.2012.07.006

[59] A. R. Warburton. 1983. Quasiconcave vector maximization: Connectedness of the sets of Pareto-optimal and weak Pareto-optimal alternatives. *J. Optim. Theory Appl.* 40, 4 (August 1983), 537–557. DOI: https://doi.org/10.1007/BF00933970

Received 25 January 2024; revised 12 June 2024; accepted 28 October 2024