How Hard Is It to Satisfy (Almost) All Roommates?

Jiehua Chen
Ben-Gurion University of the Negev, Beer Sheva, Israel
jiehua.chen2@gmail.com

Danny Hermelin
Ben-Gurion University of the Negev, Beer Sheva, Israel
hermelin@bgu.ac.il

Manuel Sorge
Ben-Gurion University of the Negev, Beer Sheva, Israel
sorge@post.bgu.ac.il

Harel Yedidsion
Ben-Gurion University of the Negev, Beer Sheva, Israel
yedidsio@post.bgu.ac.il

Abstract

The classic Stable Roommates problem (the non-bipartite generalization of the well-known Stable Marriage problem) asks whether there is a *stable matching* for a given set of agents, *i.e.* a partitioning of the agents into disjoint pairs such that *no* two agents induce a *blocking pair*. Herein, each agent has a *preference list* denoting who it prefers to have as a partner, and two agents are blocking if they prefer to be with each other rather than with their assigned partners.

Since stable matchings may not be unique, we study an NP-hard optimization variant of Stable Roommates, called Egal Stable Roommates, which seeks to find a stable matching with a minimum egalitarian cost γ, *i.e.* the sum of the dissatisfaction of the agents is minimum. The *dissatisfaction* of an agent is the number of agents that this agent prefers over its partner if it is matched; otherwise it is the length of its preference list. We also study almost stable matchings, called Min-Block-Pair Stable Roommates, which seeks to find a matching with a minimum number β of blocking pairs. Our main result is that Egal Stable Roommates parameterized by γ is fixed-parameter tractable, while Min-Block-Pair Stable Roommates parameterized by β is W[1]-hard, even if the length of each preference list is at most five.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms, Theory of computation → Algorithmic game theory, Mathematics of computing → Combinatorial optimization

Keywords and phrases NP-hard problems Data reduction rules Kernelizations Parameterized complexity analysis and algorithmics

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.35

Related Version A full version of this work is available at [14], `https://arxiv.org/abs/1707.04316`.

Acknowledgements This work is supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement number 631163.11 and Israel Science Foundation (grant number 551145/14).

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 35; pp. 35:1–35:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

This paper presents algorithms and hardness results for two variants of the Stable Roommates problem, a well-studied generalization of the classic Stable Marriage problem. Before describing our results, we give a brief background that will help motivate our work.

Stable Marriage and Stable Roommates. An instance of the Stable Marriage problem consists of two disjoint sets of n men and n women (collectively called *agents*), who are each equipped with his or her own personal *strict* preference list that ranks *every* member of the opposite sex. The goal is to find a bijection, or *matching*, between the men and the women that does not contain any blocking pairs. A *blocking pair* is a pair of man and woman who are not matched together but both prefer each other over their own matched partner. A matching with no blocking pairs is called a *stable matching*, and *perfect* if it is a bijection between *all* men and women.

Stable Marriage is a classic and fundamental problem in computer science and applied mathematics, and as such, entire books were devoted to it [24, 32, 50, 37]. The problem emerged from the economic field of matching theory, and it can be thought of as a generalization of the Maximum Matching problem when restricted to complete bipartite graphs. The most important result in this context is the celebrated Gale-Shapley algorithm [22]: This algorithm computes in polynomial time a perfect stable matching in *any* given instance, showing that regardless of their preference lists, there always exists a perfect stable matching between any equal number of men and women.

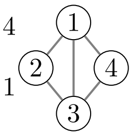
The Stable Marriage problem has several interesting variants. First, the preference lists of the agents may be *incomplete*, meaning that not every agent is an acceptable partner to every agent of the opposite sex. In graph theoretic terms, this corresponds to the bipartite incomplete case. The preference lists could also have *ties*, meaning that two or more agents may be considered equally good as partners. Finally, the agents may not be partitioned into two disjoint sets, but rather each agent may be allowed to be matched to any other agent. This corresponds to the non-bipartite case in graph theoretic terms, and is referred to in the literature as the Stable Roommates problem.

While Stable Marriage and Stable Roommates seem very similar, there is quite a big difference between them in terms of their structure and complexity. For one, any instance of Stable Marriage always contains a stable matching (albeit perhaps not perfect), even if the preference lists are incomplete and with ties. Moreover, computing some stable matching in any Stable Marriage instance with $2n$ agents can be done in $O(n^2)$ time [22]. However, an instance of Stable Roommates may have no stable matchings at all, even in the case of complete preference lists without ties (see the third example in Figure 1). Furthermore, when ties are present, deciding whether an instance of Stable Roommates contains a stable matching is NP-complete [47], even in the case of complete preference lists.

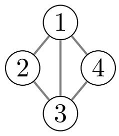
All variants of Stable Marriage and Stable Roommates mentioned here have several applications in a wide range of application domains. These include partnership issues in the real-world [22], resource allocation [5, 16, 27], centralized automated mechanisms that assign children to schools [3, 4], assigning school graduates to universities [7, 8], assigning medical students to hospitals [1, 2], and several others [6, 21, 29, 30, 33, 34, 35, 37, 38, 48, 49].

Optimization variants. As noted above, some Stable Roommates instances do not admit any stable matching at all, and in fact, empirical study suggests that a constant fraction of all sufficiently large instances will have no solution [46]. Moreover, even if a given Stable

$1: 2 \succ 3 \succ 4$
$2: 3 \succ 1$
$3: 4 \succ 2 \succ 1$
$4: 1 \succ 3$



$1: 2 \sim 3 \succ 4$
$2: 1 \succ 3$
$3: 1 \sim 2 \succ 4$
$4: 3 \succ 1$



$1: 2 \succ 3 \succ 4$
$2: 3 \succ 1 \succ 4$
$3: 1 \succ 2 \succ 4$
$4: 1 \succ 2 \succ 3$

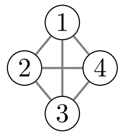


Figure 1 An example of three Stable Roommates instances, where $x \succ y$ means that x is strictly preferred to y, and $x \sim y$ means that they are equally good and tied as a partner. The instance on the left is incomplete without ties and has exactly two stable matchings $\{\{1,2\},\{3,4\}\}$ and $\{\{1,4\},\{2,3\}\}$, both of which are perfect. The instance in the middle is incomplete with ties and has two stable matchings $\{\{1,3\}\}$ and $\{\{1,2\},\{3,4\}\}$, the latter being perfect while the former not. The right instance is complete without ties and has no stable matchings at all.

Roommates instance admits a solution, this solution may not be unique, and there might be other stable matchings with which the agents are more satisfied overall. Given these two facts, it makes sense to consider two types of optimization variants for Stable Roommates: In one type, one would want to compute a stable matching that optimizes a certain social criterion in order to maximize the overall satisfaction of the agents. In the other, one would want to compute matchings which are as close as possible to being stable, where closeness can be measured by various metrics. In this paper, we focus on one prominent example of each of these two types – minimizing the egalitarian cost of a stable matching, and minimizing the number of blocking pairs in a matching which is close to being stable.

Egalitarian optimal stable matchings. Over the years, several social optimality criteria have been considered, yet arguably one of the most popular of these is the egalitarian cost metric [41, 32, 31, 36, 39]. The *egalitarian cost* of a given matching is the sum of the *rank*s of the partners of all agents, where the *rank* of the partner y of an agent x is the number of agents that are strictly preferred over y by x. The corresponding Egal Stable Marriage and Egal Stable Roommates problems ask whether there is a stable matching with egalitarian cost at most γ, for some given bound $\gamma \in \mathbb{N}$ (Section 2 contains the formal definition).

When the input preferences do not have ties (but could be incomplete), Egal Stable Marriage is solvable in $O(n^4)$ time [31]. For preferences with ties, Egal Stable Marriage becomes NP-hard [36]. Thus, already in the bipartite case, it becomes apparent that allowing ties in preference lists makes the task of computing an optimal egalitarian matching much more challenging. Marx and Schlotter [39] showed that Egal Stable Marriage is fixed-parameter tractable when parameterized by the parameter "sum of the lengths of all ties".

For Egal Stable Roommates, Feder [20] showed that the problem is NP-hard even if the preferences are complete and have no ties, and gave a 2-approximation algorithm for this case. Halldórsson et al. [25] showed inapproximability results for Egal Stable Roommates, and Teo and Sethuraman [51] proposed a specific LP formulation for Egal Stable Roommates and other variants. Cseh et al. [17] studied Egal Stable Roommates for preferences with bounded length ℓ and without ties. They showed that the problem is polynomial-time solvable if $\ell = 2$, and is NP-hard for $\ell \geq 3$.

Matchings with minimum number of blocking pairs. For the case where no stable matchings exist, the agents may still be satisfied with a matching that is close to being stable. One very natural way to measure how close a matching is to being stable is to count the number of blocking pairs [45, 19]. Accordingly, the Min-Block-Pair Stable Roommates problem asks to find a matching with a minimum number of blocking pairs.

Abraham et al. [6] showed that Min-Block-Pair Stable Roommates is NP-hard, and cannot be approximated within a factor of $n^{0.5-\varepsilon}$ unless P = NP, even if the given preferences are complete. They also showed that the problem can be solved in $n^{O(\beta)}$ time, where n and β denote the number of agents and the number of blocking pairs, respectively. This implies that the problem is in the XP class (for parameter β) of parameterized complexity. Biró et al. [9] showed that the problem is NP-hard and APX-hard even if each agent has a preference list of length at most 3, and presented a $(2\ell - 3)$-approximation algorithm for bounded list length ℓ. Biró et al. [10] and Hamada et al. [26] showed that the related variant of Stable Marriage, where the goal is to find a matching with minimum blocking pairs among all maximum-cardinality matchings, cannot be approximated within $n^{1-\varepsilon}$ unless P = NP.

Our contributions. We analyze both Egal Stable Roommates and Min-Block-Pair Stable Roommates from the perspective of parameterized complexity, under the natural parameterization of each problem (*i.e.* the egalitarian cost and number of blocking pairs, respectively). We show that while the former is fixed-parameter tractable, the latter is W[1]-hard even when each preference list has length at most five and has no ties. This shows a sharp contrast between the two problems: Computing an optimal egalitarian stable matching is a much easier task than computing a matching with minimum blocking pairs.

When no ties are present, an instance of the Egal Stable Roommates problem has a lot of structure, and so we can apply a simple branching strategy for finding a stable matching with egalitarian cost of at most γ in $2^{O(\gamma)}n^2$ time. Moreover, we derive a kernelization algorithm, obtaining a polynomial problem kernel (Theorems 3 and 4). Note that the original reduction of Feder [20] already shows that Egal Stable Roommates cannot be solved in $2^{o(\gamma)}n^{O(1)}$ time unless the Exponential Time Hypothesis [18] fails.

When ties are present, the problem becomes much more challenging because several agents may be tied as a first ranked partner and it is not clear how to match them to obtain an optimal egalitarian stable matching. Moreover, we have to handle unmatched agents. When preferences are complete or without ties, all stable matchings match the same (sub)set of agents and this subset can be found in polynomial time [24, Chapter 4.5.2]. Thus, unmatched agents do not cause any real difficulties. However, in the case of ties and with incomplete preferences, stable matchings may involve different sets of unmatched agents. Aiming at a socially optimal egalitarian stable matching, we consider the cost of an unmatched agent to be the length of its preference list [39]. (For the sake of completeness, we also consider two other variants where the cost of an unmatched agent is either zero or a constant value, and show that both these variants are unlikely to be fixed-parameter tractable.) Our first main result is given in the following theorem:

▶ **Theorem 1.** Egal Stable Roommates *can be solved in* $\gamma^{O(\gamma)} \cdot (n \log n)^3$ *time, even for incomplete preferences with ties, where* n *is the number of agents and* γ *the egalitarian cost.*

The general idea behind our algorithm is to apply random separation [13] to "separate" irrelevant pairs from the pairs that belong to the solution matching, and from some other pairs that would not block our solution. This is done in two phases, each involving some technicalities, but in total the whole separation can be computed in $\gamma^{O(\gamma)} \cdot n^{O(1)}$ time. After the separation step, the problem reduces to Minimum-Weight Perfect Matching, and we can apply known techniques. Recall that for the case where the preferences have no ties, a simple depth-bounded search tree algorithm suffices (Theorem 4).

In Section 4, we show that Min-Block-Pair Stable Roommates is W[1]-hard with respect to the parameter β (the number of the blocking pairs) even if each input preference list has length at most five and does not have ties. This implies that assuming bounded length of the preferences does not help in designing an $f(\beta) \cdot n^{O(1)}$-time algorithm for MIN-BLOCK-PAIR STABLE ROOMMATES, unless FPT = W[1]. Our W[1]-hardness result also implies as a corollary a lower-bound on the running time of any algorithm. By adapting our reduction, we also answer in the negative an open question regarding the number of blocking agents proposed by Manlove [37, Chapter 4.6.5] (Corollary 14).

▶ **Theorem 2.** *Let n denote the number of agents and β denote the number of blocking pairs. Even when each input preference list has length at most five and has no ties,* MIN-BLOCK-PAIR STABLE ROOMMATES *is W[1]-hard with respect to β and admits* no *$f(\beta) \cdot n^{o(\beta)}$-time algorithms unless the Exponential Time Hypothesis is false.*

Besides the relevant work mentioned above there is a growing body of research regarding the parameterized complexity of preference-based stable matching problems [39, 40, 43, 42, 23, 15]. Due to space constraints we deferred the proofs for results marked by $\star$ to a full version [14].

2 Definitions and notation

Let $V = \{1, 2, \ldots, n\}$ be a set of even number n agents. Each agent $i \in V$ has a subset of agents $V_i \subseteq V$ which it finds *acceptable* as a partner and has a *preference list* $\succeq_i$ on V_i (*i.e.* a transitive and complete binary relation on V_i). Here, $x \succeq_i y$ means that i weakly prefers x over y (*i.e.* x is better or as good as y). We use $\succ_i$ to denote the asymmetric part (*i.e.* $x \succeq_i y$ and $\neg(y \succeq_i x)$) and $\sim_i$ to denote the symmetric part of $\succeq_i$ (*i.e.* $x \succeq_i y$ and $y \succeq_i x$). For two agents x and y, we call x *most acceptable* to y if x is a maximal element in the preference list of y. Note that an agent can have more than one most acceptable agent. We extend $\succeq$ to $X \succeq Y$ for pairs of disjoint subsets $X, Y \subseteq V$ in the natural way.

A preference profile $\mathcal{P}$ for V is a collection $(\succeq_i)_{i \in V}$ of preference lists for each agent $i \in V$. A profile $\mathcal{P}$ may have the following properties: It is *complete* if for each agent $i \in V$ it holds that $V_i \cup \{i\} = V$; otherwise it is *incomplete*. If there are three agents $i \in V$, $x, y \in V_i$ such that $x \sim_i y$, then we say that x and y are *tied* by i and that the profile $\mathcal{P}$ has *ties*. To an instance $(V, \mathcal{P})$ we assign an *acceptability graph*, which has V as its vertex set and two agents are connected by an edge if each finds the other acceptable. Without loss of generality, G does not contain isolated vertices. The *rank* of an agent i in the preference list of some agent j is the number of agents x that j strictly prefers over i: $\mathsf{rank}_j(i) := |\{x \mid x \succ_j i\}|$.

For a preference profile with acceptability graph G and edge set $E(G)$, a *matching* $M \subseteq E(G)$ is a subset of disjoint pairs $\{x, y\}$ of agents with $x \neq y$. If $\{x, y\} \in M$, then we denote the partner y of x by $M(x)$; otherwise we call the pair $\{x, y\}$ *unmatched*. We write $M(x) = \bot$ if agent x has *no partner*; *i.e.* if agent x is not involved in any pair in M. If *no* agent x has $M(x) = \bot$ then M is *perfect*. Given a matching M of $\mathcal{P}$, an unmatched pair $\{x, y\} \in E(G) \setminus M$ *is blocking* M if both x and y prefer each other to being unmatched or to their assigned partners, *i.e.* it holds that $\big(M(x) = \bot \vee y \succ_x M(x)\big) \wedge \big(M(y) = \bot \vee x \succ_y M(y)\big)$. We call a matching M *stable* if no unmatched pair is blocking M. The STABLE ROOMMATES problem has as input a preference profile $\mathcal{P}$ for a set V of (even number) n agents and asks whether $\mathcal{P}$ admits a stable matching. When preferences are complete, each stable matching is perfect.

The two problems we consider in the paper are EGAL STABLE ROOMMATES and MIN-BLOCK-PAIR STABLE ROOMMATES. The latter asks to determine whether a given preference profile $\mathcal{P}$ for a set of agents V has a stable matching with at most β blocking pairs. The former problem asks to find a stable matching with minimum egalitarian cost; the egalitarian cost of a given matching M is as follows: $\gamma(M) := \sum_{i \in V} \mathsf{rank}_i(M(i))$, where we augment the definition rank with $\mathsf{rank}_i(\bot) := |V_i|$. For example, the second profile in Figure 1 has two stable matchings $M_1 = \{\{1, 3\}\}$ and $M_2 = \{\{1, 2\}, \{3, 4\}\}$ with $\gamma(M_1) = 4$ and $\gamma(M_2) = 2$.

Algorithm 1: A modified version of the phase-1 algorithm of Irving [28].

repeat
  foreach agent $u \in U$ whose preference list contains at least one unmarked agent **do**
    $w \leftarrow$ the first agent in the preference list of u such that $\{u, w\}$ is not yet marked
    foreach u' with $u \succ_w u'$ **do** **mark** $\{u', w\}$
until no new pair was marked in the last iteration

The egalitarian cost, as originally introduced for the Stable Marriage problem, does not include the cost of an unmatched agent because the preference lists are complete. For complete preferences, a stable matching must assign a partner to each agent, meaning that our notion of egalitarian cost equals the one used in the literature. For preferences without ties, all stable matchings match the same subset of agents [24, Chapter 4.5.2]. Thus, the two concepts differ only by a fixed value which can be pre-determined in polynomial time [24, Chapter 4.5.2]. For incomplete preferences with ties, there seems to be no consensus on whether to "penalize" stable matchings by the cost of unmatched agents [17]. Our concept of egalitarian cost complies with Marx and Schlotter [39], but we tackle other concepts as well (Section 3.3).

3 Minimizing the egalitarian cost

In this section we give our algorithmic and hardness results for Egal Stable Roommates. Section 3.1 treats the case when no ties are present, where we can use a straightforward branching strategy. In Section 3.2 we solve the case where ties are present. Herein, we need a more sophisticated approach based on random separation. Finally, in Section 3.3, we study variants of the egalitarian cost, differing in the cost assigned to unmatched agents.

3.1 Warm-up: Preferences without ties

By the stability concept, if the preferences have no ties and two agents x and y that are each other's most acceptable agents, then any stable matching must contain $\{x, y\}$, which has cost zero. Hence, we can safely add such pairs to a solution matching. After we have matched all pairs of agents with zero cost, all remaining, unmatched agents induce cost at least one when they are matched. This leads to a simple depth-bounded branching algorithm. In terms of kernelization, we can delete any two agents that induce zero cost and delete agents from some preference list that are ranked higher than γ. This gives us a polynomial kernel.

First, we recall a part of the polynomial-time algorithm by Irving [28] which finds an *arbitrary* stable matching for preferences without ties. The whole algorithm works in two phases. We present here a modified version of the first phase to determine "relevant" agents by sorting out *fixed* pairs – pairs of agents that occur in every stable matching [24, Chapter 4.4.2] – and *marked pairs* – pairs of agents that cannot occur in any stable matching. The modified phase-1 algorithm is given in Algorithm 1. Herein, by *marking a pair* $\{u, w\}$ we mean marking the agents u and w in the preference lists of w and u, respectively.

Let $\mathcal{P}_0$ be the preference profile produced by Algorithm 1. We introduce some more notions. For each agent x, let $\mathsf{first}(\mathcal{P}_0, x)$ and $\mathsf{last}(\mathcal{P}_0, x)$ denote the first and the last agent in the preference list of x that are not marked, respectively. We call a pair $\{x, y\}$ a *fixed pair* if $\mathsf{first}(\mathcal{P}_0, x) = y$ and $\mathsf{first}(\mathcal{P}_0, y) = x$. Let $\mathsf{marked}(\mathcal{P}_0)$ denote the set of all agents whose preference lists consist of only marked agents, and let $\mathsf{unmarked}(\mathcal{P}_0)$ denote the set of all agents whose preference lists have at least one unmarked agent. By [24, Chapters 4.4.2 and

4.5.2], we can neglect all agents that are in the fixed pairs and ignore all “irrelevant” agents from $\mathsf{marked}(\mathcal{P}_0)$. We can now shrink our instance to obtain a polynomial size problem kernel.

▶ **Theorem 3.** Egal Stable Roommates *for preferences without ties but with possibly incomplete preferences admits a size-$O(\gamma^2)$ problem kernel with at most $3\gamma + 1$ agents and at most $\gamma + 1$ agents in each of the preference lists.*

Proof sketch. Let $I = (\mathcal{P}, V, \gamma)$ be an instance of Egal Stable Roommates and let $\mathcal{P}_0$ be the profile that Algorithm 1 produces for $\mathcal{P}$. We use F to denote the set of agents of all fixed pairs, and we use O to denote the set of ordered pairs (x, y) of agents such that x ranks y higher than γ. Briefly put, our kernelization algorithm will delete all agents in $F \cup \mathsf{marked}(\mathcal{P}_0)$, and introduce $O(\gamma)$ dummy agents to replace the deleted agents and some more that are identified by O. Initially, F and O are set to empty sets.

1. If $|\mathsf{marked}(\mathcal{P}_0)| > \gamma$ or if there is an agent x in $\mathsf{unmarked}(\mathcal{P}_0)$ with $\mathsf{rank}_x(\mathsf{first}(\mathcal{P}_0, x)) > \gamma$, then return a trivial no-instance.
2. For each two agents $x, y \in \mathsf{unmarked}(\mathcal{P}_0)$ with $\mathsf{first}(\mathcal{P}_0, x) = y$ and $\mathsf{first}(\mathcal{P}_0, y) = x$, add to F the agents x and y. Let $\hat{\gamma} = \gamma - \sum_{x \in F} \mathsf{rank}_x(\mathsf{first}(\mathcal{P}_0, x)) - \sum_{x \in \mathsf{marked}(\mathcal{P}_0)} |V_x|$.
3. If $2\hat{\gamma} < |\mathsf{unmarked}(\mathcal{P}_0) \setminus F|$, then return a trivial no-instance.
4. Add to the original agent set a set D of $2k$ *dummy agents* $d_1, d_2, \ldots, d_{2k}$, where $k = 2\lceil \hat{\gamma}/2 \rceil$, such that for each $i \in \{1, 2, \ldots, k\}$, the preference list of d_i consists of only d_{k+i}, and the preference list of d_{k+i} consists of only d_i.
5. For each two $x, y \in \mathsf{unmarked}(\mathcal{P}_0)$ with $\mathsf{rank}_x(y) > \hat{\gamma}$, add to O the ordered pair (x, y).
6. For each agent $a \in \mathsf{unmarked}(\mathcal{P}_0) \setminus F$ do the following.
   (1) For each $i \in \{0, 1, 2, \ldots, \hat{\gamma}\}$, let x be the agent with $\mathsf{rank}_a(x) = i$. If $x \in F \cup \mathsf{marked}(\mathcal{P}_0)$ or if $(x, a) \in O$, then replace in a's preference list agent x with a dummy agent d, using a different dummy for each i, and append a to the preference list of d.
   (2) Delete all agents y in the preference list of a with $\mathsf{rank}_a(y) > \hat{\gamma}$.
7. Delete $F \cup \mathsf{marked}(\mathcal{P}_0)$ from $\mathcal{P}_0$.

The proof that the above algorithm produces a problem kernel with the desired size in the desired running time is deferred to a full version [14]. ◀

Using a simple branching algorithm, we obtain the following.

▶ **Theorem 4** ($\star$). *Let n denote the number of agents and γ denote the egalitarian cost.* Egal Stable Roommates *without ties can be solved in $O(2^\gamma \cdot n^2)$ time.*

3.2 Preferences with ties

When the preferences may contain ties, we can no longer assume that if two agents are each other's most acceptable agents, denoted as a *good pair*, then a minimum egalitarian cost stable matching would match them together; note that good pairs do not induce any egalitarian cost. This is because their match could force other pairs to be matched together that have large cost. Nevertheless, a good pair will never block any other pair, *i.e.* no agent in a good pair will form with an agent in some other pair a blocking pair. However, a stable matching may still contain some other pairs which have non-zero cost. We call such pairs *costly pairs*. Aiming to find a stable matching M with egalitarian cost at most γ, it turns out that we can also identify in $\gamma^{O(\gamma)} \cdot n^{O(1)}$ time a subset S of pairs of agents, which contains all costly pairs of M and contains no two pairs that may *induce a blocking pair*. It hence suffices to find a minimum-cost maximal matching in the graph induced by S and the good pairs.

The crucial idea is to use the random separation technique [13] to highlight the difference between the matched costly pairs in M and the unmatched costly pairs. This enables us to ignore the costly pairs which pairwisely block each other or are blocked by some pair in M so as to obtain the desired subset S. Before describing the algorithm, we show that we can focus on the case of perfect matchings, even for incomplete preferences. (Note that the case with complete preferences is covered because stable matchings for such case are always perfect.) We show this by introducing dummy agents to extend each non-perfect stable matching to a perfect one, without altering the egalitarian cost.

▶ **Lemma 5** ($\star$). EGAL STABLE ROOMMATES *for n agents and egalitarian cost γ is $O(\gamma \cdot n^2)$-time reducible to* EGAL STABLE ROOMMATES *for at most $n+\gamma$ agents and egalitarian cost γ with an additional requirement that the stable matching should be perfect.*

Lemma 5 allows, in a subprocedure of our main algorithm, to compute a min-cost *perfect* matching in polynomial time instead of a min-cost *maximal* matching (which is NP-hard).

The algorithm. As mentioned, we use random separation [13]. We apply it already in derandomized form using Bshouty's construction of cover-free families [11], a notion related to universal sets [44]. Let $\hat{n}, p, q \in \mathbb{N}$ such that $p + q \leq \hat{n}$. A family $\mathcal{F}$ of subsets of some $\hat{n}$-element universe U is called $(\hat{n}, p, q)$-*cover-free family* if for each subset $S \subseteq U$ of cardinality $p+q$ and each subset $S' \subseteq S$ of cardinality p, there is a member $A \in \mathcal{F}$ with $S \cap A = S'$.[1] The result by Bshouty [11, Theorem 4] implies that if $p \in o(q)$, then there is an $(\hat{n}, p, q)$-cover-free family of cardinality $q^{O(p)} \cdot \log \hat{n}$ which can be computed in time linear of this cardinality.

In the remainder of this section, we prove Theorem 1. Let $\mathcal{P}$ be a preference profile for a set V of agents, possibly incomplete and with ties. For brevity we denote by a *solution* (of $\mathcal{P}$) a stable matching M with egalitarian cost at most γ. By Lemma 5, we assume that each solution is perfect. Our goal is to construct a graph with vertex set V which contains all matched "edges", representing the pairs, of some solution and some other edges for which no two edges in this graph are blocking each other. Herein, we say that two edges $e, e' \in \binom{V}{2}$ are *blocking each other* if, assuming both edges (which are two disjoint pairs of agents) are in the matching, they would *induce a blocking pair*, *i.e.* $u' \succ_u v$ and $u \succ_{u'} v'$, where $e \coloneqq \{u, v\}$ and $e' \coloneqq \{u', v'\}$. Pricing the edges with their corresponding cost, by Lemma 5, it is then enough to find a minimum-cost perfect matching. The graph is constructed in three phases (see Algorithm 2). In the first phase, we start with the acceptability graph of our profile $\mathcal{P}$ and remove all edges whose "costs" each exceed γ. In the second and the third phases, we remove all edges that block each other while keeping a stable matching with minimum egalitarian cost intact.

We introduce some more necessary concepts. Let G be the acceptability graph corresponding to $\mathcal{P}$ with vertex set V, which also denotes the agent set, and with edge set E. The *cost* of an edge $\{x, y\}$ is the sum of the ranks of each endpoint in the preference list of the other: $\mathsf{rank}_x(y) + \mathsf{rank}_y(x)$. We call an edge $e \coloneqq \{x, y\}$ a *zero edge* if it has *cost* zero, *i.e.* $\mathsf{rank}_x(y) + \mathsf{rank}_y(x) = 0$, otherwise it is a *costly edge* if the cost does not exceed γ. We ignore all edges with cost exceeding γ. Note that no such edge belongs to or is blocking any stable matching with egalitarian cost at most γ. To distinguish between zero edges and costly edges, we construct two subsets E^{zero} and E^{exp} such that E^{zero} consists of all zero edges, *i.e.* $E^{\mathsf{zero}} \coloneqq \{\{x, y\} \in E \mid \mathsf{rank}_x(y) + \mathsf{rank}_y(x) = 0\}$, and E^{exp} consists of all costly edges, *i.e.* $E^{\mathsf{exp}} \coloneqq \{\{x, y\} \in E \mid 0 < \mathsf{rank}_x(y) + \mathsf{rank}_y(x) \leq \gamma\}$.

[1] The standard definition of cover-free families [11] is stated differently from but equivalent [12] to ours.

Algorithm 2: Constructing a perfect stable matching of egalitarian cost at most γ.

Input: A set V of agents, a preference profile $\mathcal{P}$ over V, and a budget $\gamma \in \mathbb{N}$.
Output: A stable matching of egalitarian cost at most γ if it exists.

/* Phase 1 */
1 $(V, E) \leftarrow$ The acceptability graph of $\mathcal{P}$
2 $E^{\text{zero}} \leftarrow \{\{x, y\} \in E \mid \text{rank}_x(y) + \text{rank}_y(x) = 0\}$ // The set of zero edges in E
3 $E^{\text{exp}} \leftarrow \{\{x, y\} \in E \mid 1 \leq \text{rank}_x(y) + \text{rank}_y(x) \leq \gamma\}$ // The set of costly edges in E
4 $E_1 \leftarrow E^{\text{zero}} \cup E^{\text{exp}}$
/* Phase 2 */
5 $\mathcal{F}^{\text{exp}} \leftarrow (|E^{\text{exp}}|, \gamma, \gamma^3)$-cover-free family over the universe E^{exp}
6 **foreach** $E' \in \mathcal{F}^{\text{exp}}$ **do**
7   Apply Rule 1 to E_1 to obtain E_2
  /* Phase 3 */
8   $\mathcal{C} \leftarrow (|V|, \gamma^2 + 2 \cdot \gamma, 2 \cdot \gamma)$-cover-free family over the universe V
9   **foreach** $V' \in \mathcal{C}$ **do**
10   Apply Rules 2 and 3 to E_2 to obtain E_3
11     $M \leftarrow$ Minimum-cost perfect matching in the graph (V, E_3) or $\perp$ if none exists
12   **if** $M \neq \perp$ and M has cost at most γ **then return** M

Phase 1. We construct a graph $G_1 = (V, E_1)$ from G with vertex set V and with edge set $E_1 := E^{\text{zero}} \cup E^{\text{exp}}$. The following is easy to see.

▶ **Lemma 6.** *If $\mathcal{P}$ has a stable matching M with egalitarian cost at most γ, then $M \subseteq E_1$.*

Observe also that a zero edge cannot block any other edge because the agents in a zero edge already obtain their most acceptable agents. Thus, we have the following.

▶ **Lemma 7.** *If two edges in E_1 block each other, then they are both costly edges.*

Phase 2. In this phase, comprising Lines 5–7 in Algorithm 2, we remove from G_1 some of the costly edges that block each other (by Lemma 7, no zero edges are blocking any other edge). For technical reasons, we distinguish two types of costly edges: We say that a costly edge e with $e := \{u, v\}$ is *critical for its endpoint* u if the largest possible rank of v over all linearizations of the preference list of u exceeds γ, *i.e.* $|\{x \in V_u \setminus \{v\} \mid x \succeq_u v\}| > \gamma$. Otherwise, e is *harmless for* u. If an edge is critical for at least one endpoint, then we call it *critical* and otherwise *harmless*. Observe that a critical edge could still belong to a solution. If two edges e and e' block each other due to the blocking pair $\{u, u'\}$ with $u \in e, u' \in e'$ such that e' is harmless for u', then we say that e is *harmlessly blocking* e' (at the endpoint u'). Note that blocking is symmetric while harmlessly blocking is not.

Intuitively, we want to distinguish the solution edges from all edges blocked by the solution. There is a "small" number of harmless edges blocked by the solution, so we can easily distinguish between them. For the critical edges, we do not have such a bound; we deal with the critical edges blocked by the solution in Phase 3 in some other way.

▶ **Lemma 8** ($\star$). *Let M be a stable matching with egalitarian cost at most γ. In G_1, at most γ^2 edges are harmlessly blocked by some edge in M.*

Let $M' := M \cap E^{\text{exp}}$ be the set of all costly edges in some solution M and let B_M be the set of all edges harmlessly blocked by some edge in M. By the definition of costly edges and by Lemma 8, it follows that $|M'| \leq \gamma$ and $|B_M| \leq \gamma^2$. In order to identify and delete all edges in B_M we apply random separation. Compute a $(|E^{\text{exp}}|, \gamma, \gamma^2)$-cover-free family $\mathcal{F}^{\text{exp}}$ over the universe E^{exp}. For each member of $\mathcal{F}^{\text{exp}}$, perform all the computations below (in this phase and in Phase 3). By the properties of cover-free families, $\mathcal{F}^{\text{exp}}$ contains a *good* member E' that "separates" M' from B_M, *i.e.* $M' \subseteq E'$ and $B_M \subseteq E^{\mathsf{exp}} \setminus E'$. Formally, we call a member $E' \in \mathcal{F}^{\mathsf{exp}}$ *good* if there is a solution M such that each costly edge in M belongs to E', and each edge that is harmlessly blocked by M belongs to $E^{\mathsf{exp}} \setminus E'$. We also call E' *good for* M. By the property of cover-free families, if there is a solution M, then $\mathcal{F}^{\mathsf{exp}}$ contains a member E' which is good for M. In the following we present two data reduction rules that delete edges and show their correctness. By *correctness* we mean that, if some member $E' \in \mathcal{F}^{\mathsf{exp}}$ is good, then the corresponding solution is still present after the edge deletion.

Recall that the goal was to compute a graph that contains all edges from a solution and some other edges such that no two edges in the graph block each other. Observe that we can ignore the edges in $E^{\mathsf{exp}} \setminus E'$, because, if E' is good, then it contains all costly edges in the corresponding solution. This implies the correctness of the first part of the following reduction rule. The correctness for the second part follows from the definition of being good.

▶ **Rule 1.** Remove all edges in $E^{\mathsf{exp}} \setminus E'$ from E_1. If there are two edges $e, e' \in E'$ that are harmlessly blocking each other, then remove both e and e' from E_1.

Let $G_2 = (V, E_2)$ be the graph obtained from G_1 by exhaustively applying Rule 1. By the goodness of E' and by the correctness of Rule 1, we have the following.

▶ **Lemma 9.** *If there is a stable matching M with egalitarian cost at most γ, then $\mathcal{F}^{\mathsf{exp}}$ contains a member E' such that the edge set E_2 of G_2 defined for E' contains all edges of M.*

By Lemma 7 and since all pairs of edges that are harmlessly blocking each other are deleted by Rule 1, we have the following.

▶ **Lemma 10.** *If two edges in G_2 block each other due to a blocking pair $\{u, u'\}$, then one of the edges is critical for u or u'.*

Phase 3. In Line 10 of Algorithm 2 we remove from G_2 the remaining (critical) edges that do not belong to M but are blocked by some other edges. This includes the edges that are blocked by M. While the number of edges blocked by M could still be unbounded, we show that there are only $O(\gamma^2)$ agents due to which an edge could be blocked by M. The idea here is to identify such agents, helping to find and delete edges blocked by M or blocking some other edges. We introduce one more notion. Consider an arbitrary matching N (*i.e.* a set of disjoint pairs of agents) of G_2. Let $e \in N$ and $e' \in E_2 \setminus N$ be two edges. If they induce a blocking pair $\{u, u'\}$ with $u \in e$ and $u' \in e'$, then we say that u' is a *culprit* of N. We obtain the following upper bound on the number of culprits with respect to a solution.

▶ **Lemma 11** ($\star$). *Let M be a stable matching. Then, each culprit of M is incident with some edge in M. If M has egalitarian cost at most γ, then it admits at most γ culprits.*

Consider a solution M and let $\mathsf{Cl}(M) = \{v \in V \mid v \text{ is a culprit of or incident with some costly edge of } M\}$. By Lemma 11 and since M has at most γ costly edges, it follows that $|\mathsf{Cl}(M)| \leq 3\gamma$. We aim to identify in $\mathsf{Cl}(M)$ a subset $\mathsf{R}(M)$ of agents incident with a critical edge in M, *i.e.* $\mathsf{R}(M) = \{v \in \mathsf{Cl}(M) \mid \{v, w\} \in M \text{ with } \{v, w\} \text{ being critical for } v\}$. Since M has at most γ costly edges, it follows that $|\mathsf{R}(M)| \leq 2\gamma$. To "separate" $\mathsf{R}(M)$ from $\mathsf{Cl}(M)$, we compute a $(|V|, 2\gamma, 3\gamma)$-cover-free family $\mathcal{C}$ on the set V. We call a member $V' \in \mathcal{C}$ *good* if there is a solution $M \subseteq E_2$ such that $\mathsf{R}(M) \subseteq V'$ and $(\mathsf{Cl}(M) \setminus \mathsf{R}(M)) \subseteq V \setminus V'$. By a similar reasoning as given for Phase 2 and by the properties of cover-free families, if there is a solution $M \subseteq E_2$, then $\mathcal{C}$ contains a good member V'. For this member, the following two reduction rules will not destroy the solution.

▶ **Rule 2** ($\star$). For each agent $y \in V \setminus V'$, delete all incident edges that are critical for y.

After having exhaustively applied Rule 2, we use the following reduction rule.

▶ **Rule 3** ($\star$). If E_2 contains two edges e and e' that induce a blocking pair $\{u, u'\}$ with $u \in e$ and $u' \in e'$ such that e is critical for u, then remove e' from E_2.

Let $G_3 = (V, E_3)$ be the graph obtained after having exhaustively applied Rules 2 and 3 to G_2. By the correctness of Rules 2 and 3 we have the following.

▶ **Lemma 12.** *If there is a stable matching $M \subseteq E_2$ with egalitarian cost at most γ, then the constructed cover-free family $\mathcal{C}$ contains a good member $V' \in \mathcal{C}$ such that the edge set E_3 of G_3 resulting from the application of Rules 2 and 3 contains all edges of M.*

Since for each member $V' \in \mathcal{C}$, we delete all edges that pairwisely block each other, each perfect matching in G_3 induces a stable matching. We thus have the following.

▶ **Lemma 13** ($\star$). *If G_3 admits a perfect matching M with edge cost at most γ, then M corresponds to a stable matching with egalitarian cost at most γ.*

Thus, to complete Algorithm 2, in Line 11 we compute a minimum-cost perfect matching for G_3 and output yes, if it has egalitarian cost at most γ. Summarizing, by Lemma 5 if there is a stable matching of egalitarian cost at most γ, then it is perfect and thus, by Lemmas 6, 9 and 12, there is a perfect matching in G_3 of cost at most γ. Hence, if our input is a yes-instance, then Algorithm 2 accepts by returning a desired solution. Ifi it accepts, then by Lemma 13 the input is a yes-instance. The running time is proved in a full version [14].

3.3 Variants of the egalitarian cost for unmatched agents

As discussed in Sections 1 and 2, when the input preferences are incomplete, a stable matching may leave some agents unmatched. In the absence of ties, all stable matchings leave the same set of agents unmatched [24, Chapter 4.5.2]. Hence, whether an unmatched agent should infer any cost is not relevant in terms of complexity. However, when preferences are incomplete and with ties, stable matchings may involve different sets of matched agents. The cost of unmatched agents changes the parameterized complexity dramatically. In particular, as soon as the cost of an unmatched agent is bounded by a fixed constant, seeking for an optimal egalitarian stable matching is parameterized intractable. See the full version [14].

4 Minimizing the number of blocking pairs

In this section, we strengthen the known result [6] by showing that MIN-BLOCK-PAIR STABLE ROOMMATES is W[1]-hard with respect to "the number β blocking pairs", even when each preference list has length at most five. The main building block of our reduction, which is from the W[1]-hard MULTI-COLORED INDEPENDENT SET problem (see our full version [14] for the definition), is a selector gadget (Construction 1) that always induces at least one blocking pair and allows for many different configurations. To keep the lengths of the preference lists short we use "duplicating" agents (Construction 2).

First, we discuss a vertex-selection gadget which we later use to select a vertex of the input graph into the independent set. The selected vertex is indicated by an agent which is matched to someone *outside* of the vertex-selection gadget. The gadget always induces at least one blocking pair. An illustration is shown in a full version [14]. In the following, let n' be a positive integer, and all additions and subtractions in the superscript are taken modulo $2n' + 1$:

▶ **Construction 1.** Consider the following four disjoint sets U, A, C, D of $2n' + 1$ agents each, where $A \coloneqq \{a^i \mid 0 \le i \le 2n'\}$, $U \coloneqq \{u^i \mid 0 \le i \le 2n'\}$, $C \coloneqq \{c^i \mid 0 \le i \le 2n'\}$, and $D \coloneqq \{d^i \mid 0 \le i \le 2n'\}$. The preferences of the agents in $A \cup C \cup D$ are: $\forall i \in \{0, 1, \ldots, 2n'\}$: agent a^i: $a^{i+1} \succ a^{i-1} \succ u^i \succ c^i \succ d^i$, agent c^i: $d^i \succ a^i$, agent d^i: $a^i \succ c^i$.

The preferences of the agents in U are intentionally left unspecified and we define them later when we use the gadget. Regardless of the preferences of the agents in U, we can verify that if no a^i obtains an agent u^i as a partner, then it induces at least two blocking pairs.

Next, we construct verification gadgets that ensure that no two adjacent vertices are chosen into the independent set solution. See the full version for an illustration [14]. Herein, let δ be a positive integer, and all additions and subtractions in the superscript are taken modulo $2\delta + 2$.

▶ **Construction 2.** Consider two disjoint sets $X \uplus Y$ where $X = \{x^i \mid 0 \le i \le 2\delta + 1\}$ is a set of $2\delta + 2$ agents and $Y = \{y^i \mid 1 \le i \le \delta\}$ is a set of δ agents. Let a, b be two agents distinct from the agents in $X \cup Y$. The preference lists of the agents from X are as follows.

$$\begin{array}{lllll} & \text{Agent } x^0\colon & x^1 \succ a \succ x^{2\delta+1}, & \text{Agent } x^{2\delta+1}\colon & x^0 \succ b \succ x^{2\delta}. \\ \forall i \in \{1, \ldots, \delta\}\colon & \text{Agent } x^{2i-1}\colon & x^{2i} \succ x^{2i-2}, & \text{Agent } x^{2i}\colon & x^{2i+1} \succ y^i \succ x^{2i-1}. \end{array}$$

The preferences of the agents a, b and those in Y are intentionally left unspecified and will be defined when we use the gadget later. Regardless of the concrete preferences of agents in $Y \cup \{a, b\}$, we claim that the above gadget has two possible matchings such that no blocking pair involves any agent from X. The first one is straightforward from the definition of the preference lists: $\{\{x^{2i}, x^{2i+1}\} \mid i \in \{0, 1, \ldots, \delta\}\}$. The second one matches x^0 to a, $x^{2\delta+1}$ to b, while keeping the remaining agents matched in some stable way.

Proof sketch of Theorem 2. Let $(G = (V_1, V_2, \ldots, V_k, E))$ be a Multi-Colored Independent Set instance (see our full version [14] for the definition). Without loss of generality, assume that each vertex subset V_j has exactly $2n' + 1$ vertices with the form $V_j = \{v_j^0, v_j^1, \ldots, v_j^{2n'}\}$. Construct a Min-Block-Pair Stable Roommates instance with the following groups of agents: $U_j, A_j, B_j, C_j, D_j, F_j, W_j$, $j \in \{1, 2, \ldots, k\}$, where U_j corresponds to the vertex subset V_j. Let δ_j^i be the degree of vertex v_j^i. For each vertex $v_j^i \in V_j$, construct $2\delta_j^i + 2$ agents $u_j^{i,0}, u_j^{i,1}, \ldots, u_j^{i,2\delta_j^i+1}$ and let $U_j^i = \{u_j^{i,z} \mid 0 \le z \le 2\delta_j^i + 1\}$. Define $U_j = \cup_{0 \le i \le 2n'} U_j^i$. For each $(Q, q) \in \{(A, a), (B, b), (C, c), (D, d), (F, f), (W, w)\}$ and for each $i \in \{1, 2, \ldots, k\}$, the set $Q_j \coloneqq \{q_j^i \mid 0 \le i \le 2n'\}$ consists of $2n' + 1$ agents. The preference lists of the agents in U_j^i obey the verification gadget constructed in Construction 2. Formally, for each $j \in \{1, \ldots, k\}$ and each $i \in \{0, 1, \ldots, 2n'\}$ we introduce a *verification gadget for* v_j^i as in Construction 2 where we set $\delta = \delta_j^i$, $x^z = u_j^{i,z}$, $0 \le z \le 2\delta_j^i + 1$, $a = a_j^i$, and $b = b_j^i$. The agents from Y correspond to the neighbors of v_j^i: For each neighbor $v_{j'}^{i'}$ of v_j^i we pick a not-yet-set agent y^z in the verification gadget for v_j^i and a not-yet-set agent $y^{z'}$ in the verification gadget for $v_{j'}^{i'}$, and define $y^z = u_{j'}^{i',2z'}$ and $y^{z'} = u_j^{i,2z}$.

For each $j \in \{1, \ldots, k\}$, the preference lists of $A_j \cup C_j \cup D_j \cup \{u_j^{i,0} \mid 0 \le i \le 2n'\}$ obey Construction 1. Formally, for each $j \in \{1, \ldots, k\}$ we introduce a vertex-selection gadget as in Construction 1 and for each $i \in \{0, 1, \ldots, 2n'\}$ we set $a^i = a_j^i$, $c^i = c_j^i$, $d^i = d_j^i$, and $u^i = u_j^{i,0}$. Analogously, for each $j \in \{1, \ldots, k\}$ we introduce a vertex-selection gadget for $B_j \cup F_j \cup W_j \cup \{u_j^{i,2\delta_j^i+1} \mid 0 \le i \le 2n'\}$: For each $i \in \{0, 1, \ldots, 2n'\}$ we set $a^i = b_j^i$, $c^i = f_j^i$, $d^i = w_j^i$, and $u^i = u_j^{i,2\delta_i^j+1}$. To complete the construction, we set the upper bound on the number of blocking pairs as $\beta = 2k$. The correctness proof is deferred to a full version [14]. ◀

The reduction given in the proof of Theorem 2 shows that the lower-bound on β in [6, Lemma 4] is tight. The reduction also answers an open question by Manlove [37, Chapter 4.6.5] about the complexity of the following problem. Given a preference profile and an integer η, Min-Block-Agents Stable Roommates asks whether there is a matching with at most η *blocking agents*. Herein, an agent is a *blocking agent* if it is involved in a blocking pair.

▶ **Corollary 14** ($\star$). *Let n be the number of agents and η be the number of blocking agents. Even when each input preference list has length at most five and has no ties,* Min-Block-Agents Stable Roommates *is NP-hard and W[1]-hard with respect to η.* Min-Block-Agents Stable Roommates *for preferences without ties is solvable in $O(2^{\eta^2} \cdot n^{\eta+2})$ time.*

References

1. National resident matching program website. URL: http://www.nrmp.org.
2. Scottish foundation allocation scheme website. URL: http://www.nes.scot.nhs.uk/sfas.
3. Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. The Boston public school match. *American Economic Review*, 95(2):368—371, 2005.
4. Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. The New York City high school match. *American Economic Review*, 95(2):364–367, 2005.
5. David Abraham, Ning Chen, Vijay Kumar, and Vahab S. Mirrokni. Assignment problems in rental markets. In *Proceedings of the Second International Workshop on Internet and Network Economics (WINE '06)*, pages 198–213, 2006.
6. David J. Abraham, Péter Biró, and David Manlove. "Almost stable" matchings in the roommates problem. In *Proceedings of the Third International Workshop on Approximation and Online Algorithms (WAOA '05)*, pages 1–14, 2005. doi:10.1007/11671411_1.
7. Mourad Baïou and Michel Balinski. Student admissions and faculty recruitment. *Theoretical Computer Science*, 322(2):245–265, 2004.
8. Péter Biró and Sofya Kiselgof. College admissions with stable score-limits. *Central European Journal of Operations Research*, 23(4):727–741, 2015. doi:10.1007/s10100-013-0320-9.
9. Péter Biró, David Manlove, and Eric McDermid. "Almost stable" matchings in the Roommates problem with bounded preference lists. *Theoretical Computer Science*, 432:10–20, 2012.
10. Péter Biró, David Manlove, and Shubham Mittal. Size versus stability in the marriage problem. *Theoretical Computer Science*, 411(16-18):1828–1841, 2010.
11. Nader H. Bshouty. Linear time Constructions of some d-Restriction Problems. In *Proceedings of the 9th International Conference on Algorithms and Complexity (CIAC '15)*, volume 9079 of *Lecture Notes in Computer Science*, pages 74–88. Springer, 2015.
12. Nader H. Bshouty and Ariel Gabizon. Almost Optimal Cover-Free Families. In *Proceedings of the 11th International Conference on Algorithms and Complexity (CIAC '17)*, volume 10236 of *Lecture Notes in Computer Science*, pages 140–151. Springer, 2017.
13. Leizhen Cai, Siu Man Chan, and Siu On Chan. Random Separation: A New Method for Solving Fixed-Cardinality Optimization Problems. In *Proceedings of the Second International Workshop on Parameterized and Exact Computation (IWPEC '06)*, pages 239–250. Springer, 2006.
14. Jiehua Chen, Danny Hermelin, Manuel Sorge, and Harel Yedidsion. How hard is it to satisfy (almost) all roommates? Technical report, arXiv:1707.04316 [cs.CC], 2018.
15. Jiehua Chen, Rolf Niedermeier, and Piotr Skowron. Stable marriage with multi-modal preferences. Technical report, arXiv:1801.02693 [cs.MA,cs.DS], 2018.
16. Yan. Chen and Tayfun Sönmez. Improving efficiency of on-campus housing: An experimental study. *American Economic Review*, 92(5):1669—-1686, 2002.

17 Ágnes Cseh, Robert W. Irving, and David F. Manlove. The stable roommates problem with short lists. In *Proceedings of the 9th International Symposium on Algorithmic Game Theory (SAGT '16)*, pages 207–219, 2016.

18 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

19 Kimmo Eriksson and Olle Häggström. Instability of matchings in decentralized markets with various preference structures. *International Journal of Game Theory*, 36(3):409–420, 2008.

20 Tomás Feder. A new fixed point approach for stable networks and stable marriages. *Journal of Computer and System Sciences*, 45(2):233–284, 1992.

21 Anh-Tuan Gai, Dmitry Lebedev, Fabien Mathieu, Fabien de Montgolfier, Julien Reynier, and Laurent Viennot. Acyclic preference systems in P2P networks. In *Proceedings of the 13th International Euro-Par Conference*, pages 825–834, 2007.

22 D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 120(5):386–391, 2013. `doi:10.4169/amer.math.monthly.120.05.386`.

23 Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. Balanced stable marriage: How close is close enough? Technical report, arXiv:1707.09545 [cs.DS], 2017.

24 Dan Gusfield and Robert W. Irving. *The Stable marriage problem–Structure and algorithms*. Foundations of computing series. MIT Press, 1989.

25 Magnús M. Halldórsson, Robert W. Irving, Kazuo Iwama, David Manlove, Shuichi Miyazaki, Yasufumi Morita, and Sandy Scott. Approximability results for stable marriage problems with ties. *Theoretical Computer Science*, 306(1-3):431–447, 2003. `doi:10.1016/S0304-3975(03)00321-9`.

26 Koki Hamada, Kazuo Iwama, and Shuichi Miyazaki. An improved approximation lower bound for finding almost stable maximum matchings. *Information Processing Letters*, 109(18):1036–1040, 2009.

27 Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979.

28 Robert W. Irving. An efficient algorithm for the 'stable roommates' problem. *Journal of Algorithms*, 6(4):577–595, 1985.

29 Robert W. Irving. Optimal stable marriage. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 1470–1473. Springer, 2016. `doi:10.1007/978-1-4939-2864-4_271`.

30 Robert W. Irving. Stable marriage. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 2060–2064. Springer, 2016. `doi:10.1007/978-1-4939-2864-4_393`.

31 Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the 'optimal' stable marriage. *Journal of the ACM*, 34(3):532–543, 1987. `doi:10.1145/28869.28871`.

32 Donald Knuth. *Mariages Stables*. Les Presses de L'Université de Montréal, 1976.

33 Eija Kujansuu, Tuukka Lindberg, and Erkki Mäkinen. The Stable Roommates problem and chess tournament pairings. *Divulgaciones Matemáticas*, 7(1):19–28, 1999.

34 Dmitry Lebedev, Fabien Mathieu, Laurent Viennot, Anh-Tuan Gai, Julien Reynier, and Fabien de Montgolfier. On using matching theory to understand P2P network design. In *Proceedings of the International Network Optimization Conference (INOC 2007)*, pages 1–7, 2007.

35 David Manlove. Hospitals/residents problem. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*. Springer, 2008.

36 David Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002. `doi:10.1016/S0304-3975(01)00206-7`.

37 David F. Manlove. *Algorithmics of Matching Under Preferences*, volume 2 of *Series on Theoretical Computer Science*. WorldScientific, 2013. `doi:10.1142/8591`.

38 David F. Manlove and Gregg O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM Journal of Experimental Algorithmics*, 19(1), 2014. `doi:10.1145/2670129`.

39 Dániel Marx and Ildikó Schlotter. Parameterized complexity and local search approaches for the stable marriage problem with ties. *Algorithmica*, 58(1):170–187, 2010. `doi:10.1007/s00453-009-9326-z`.

40 Dániel Marx and Ildikó Schlotter. Stable assignment with couples: Parameterized complexity and local search. *Discrete Optimization*, 8(1):25–40, 2011. `doi:10.1016/j.disopt.2010.07.004`.

41 D. G. McVitie and L. B. Wilson. The Stable Marriage problem. *Communications of the ACM*, 14(7):486–490, 1971.

42 Kitty Meeks and Baharak Rastegari. Solving hard stable matching problems involving groups of similar agents. Technical report, arXiv:1708.04109 [cs.GT], 2017.

43 Matthias Mnich and Ildikó Schlotter. Stable marriage with covering constraints-A complete computational trichotomy. In *Proceedings of the 10th International Symposium on Algorithmic Game Theory (SAGT '17)*, pages 320–332, 2017. `doi:10.1007/978-3-319-66700-3_25`.

44 M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 182–191, 1995.

45 Muriel Niederle and Alvin E. Roth. Market culture: How norms governing exploding offers affect market performance. Working Paper 10256, National Bureau of Economic Research, February 2004.

46 Boris Pittel and Robert W. Irving. An upper bound for the solvability of a random stable roommates instance. *Random Structures and Algorithms*, 5(3):465–487, 1994.

47 Eytan Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11(2):285–304, 1990. `doi:10.1016/0196-6774(90)90007-2`.

48 Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188, 2005. `doi:10.1016/j.jet.2005.04.004`.

49 Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97(3):828–851, 2007.

50 Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, 1992. Part of Econometric Society Monographs.

51 Chung-Piaw Teo and Jay Sethuraman. On a cutting plane heuristic for the stable roommates problem and its applications. *European Journal of Operational Research*, 123(1):195–205, 2000.