

McKay, M. and Manlove, D. (2021) The Three-Dimensional Stable Roommates Problem with Additively Separable Preferences. In: 14th International Symposium on Algorithmic Game Theory (SAGT 2021), Aarhus, Denmark, 21-24 Sep 2021, pp. 266-280. ISBN 9783030859466 (doi:[10.1007/978-3-030-85947-3_18](https://doi.org/10.1007/978-3-030-85947-3_18)).

This is the author's final accepted version.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

<http://eprints.gla.ac.uk/245422/>

Deposited on: 06 July 2021

The Three-Dimensional Stable Roommates Problem with Additively Separable Preferences^{*}

Michael McKay^[0000–0003–1496–7434] and David Manlove^[0000–0001–6754–7308]

School of Computing Science, University of Glasgow, UK
`m.mckay.1@research.gla.ac.uk`, `david.manlove@glasgow.ac.uk`

Abstract. The Stable Roommates problem involves matching a set of agents into pairs based on the agents’ strict ordinal preference lists. The matching must be stable, meaning that no two agents strictly prefer each other to their assigned partners. A number of three-dimensional variants exist, in which agents are instead matched into triples. Both the original problem and these variants can also be viewed as hedonic games. We formalise a three-dimensional variant using general additively separable preferences, in which each agent provides an integer valuation of every other agent. In this variant, we show that a stable matching may not exist and that the related decision problem is NP-complete, even when the valuations are binary. In contrast, we show that if the valuations are binary and symmetric then a stable matching must exist and can be found in polynomial time. We also consider the related problem of finding a stable matching with maximum utilitarian welfare when valuations are binary and symmetric. We show that this optimisation problem is NP-hard and present a novel 2-approximation algorithm.

Keywords: Stable roommates · Stable matching · Three dimensional roommates · Hedonic games · Coalition formation · Complexity

1 Introduction

The Stable Roommates problem (SR) is a classical problem in the domain of matching under preferences. It involves a set of agents that must be matched into pairs. Each agent provides a *preference list*, ranking all other agents in strict order. We call a set of pairs in which each agent appears in exactly one pair a *matching*. The goal is to produce a matching M that admits no *blocking pair*, which comprises two agents, each of whom prefers the other to their assigned partner in M . Such a matching is called *stable*. This problem originates from a seminal paper of Gale and Shapley, published in 1962, as a generalisation of the Stable Marriage problem [15]. They showed that an SR instance need not contain a stable matching. In 1985, Irving presented a polynomial-time algorithm to either find a stable matching or report that none exist, given an arbitrary SR

^{*} This work was supported by the Engineering and Physical Sciences Research Council (Doctoral Training Partnership grant number EP/R513222/1 and grant number EP/P028306/1)

instance [20]. Since then, many papers have explored extensions and variants of the fundamental SR problem model.

In this paper we consider the extension of SR to three dimensions (i.e., agents must be matched into triples rather than pairs). A number of different formalisms have already been proposed in the literature. The first, presented in 1991 by Ng and Hirschberg, was the *3-Person Stable Assignment Problem* (3PSA) [24]. In 3PSA, agents’ preference lists are formed by ranking every pair of other agents in strict order. A matching M is a partition of the agents into unordered triples. A *blocking triple* t of M involves three agents that each prefer their two partners in t to their two assigned partners in M . Accordingly, a *stable matching* is one that admits no blocking triple. The authors showed that an instance of this model may not contain a stable matching and the associated decision problem is NP-complete [24]. In the instances constructed by their reduction, agents’ preferences may be *inconsistent* [19], meaning that it is impossible to derive a logical order of individual agents from a preference list ranking pairs of agents.

In 2007, Huang considered the restriction of 3PSA to *consistent* preferences. He showed that a stable matching may still not exist and the decision problem remains NP-complete [19,18]. In his technical report, he also described another variant of 3PSA using *Precedence by Ordinal Number* (PON). PON involves each agent providing a preference list ranking all other agents individually. An agent’s preference over pairs is then based on the sum of the ranks of the agents in each pair. Huang left open the problem of finding a stable matching, as defined here, in the PON variant. He also proposed another problem variant involving a more general system than PON, in which agents provide arbitrary numerical “ratings”. It is this variant that we consider in this paper. He concluded his report by asking if there exist special cases of 3PSA in which stable matchings can be found using polynomial time algorithms. This question is another motivation for our paper.

The same year, Iwama, Miyazaki and Okamoto presented another variant of 3PSA [21]. In this model, agents rank individual agents in strict order of preference, and an ordering over pairs is inferred using a specific *set extension rule* [5,7]. The authors showed that a stable matching may not exist and that the decision problem remains NP-complete.

In 2009, Arkin et al. presented another variant of 3PSA called *Geometric 3D-SR* [1]. In this model, preference lists ranking pairs are derived from agents’ relative positions in a metric space. Among other results, they showed that in this model a stable matching, as defined here, need not exist. In 2013, Deineko and Woeginger showed that the corresponding decision problem is NP-complete [14].

All of the problem models described thus far, including SR, can be viewed as *hedonic games* [6]. A hedonic game is a type of *coalition formation game*. In general, coalition formation games involve partitioning a set of agents into disjoint sets, or coalitions, based on agents’ preferences. The term ‘hedonic’ refers to the fact that agents are only concerned with the coalition that they belong to. The study of hedonic games and coalition formation games is broad and many different problem models have been considered in the literature [17].

In particular, SR and its three-dimensional variants can be viewed as hedonic games with a constraint on permissible coalition sizes [26]. In the context of a hedonic game, the direct analogy of stability as described here is *core stability*. In a given hedonic game, a partition is *core stable* if there exists no set of agents S , of any size, where each agent in S prefers S to their assigned coalition [6].

Recently, Boehmer and Elkind considered a number of hedonic game variants, including 3PSA, which they described as *multidimensional roommate games* [8]. In their paper they supposed that the agents have *types*, and an agent’s preference between two coalitions depends only on the proportion of agents of each type in each coalition. They showed that, for a number of different ‘solution concepts’, the related problems are NP-hard, although many problems are solvable in linear time when the room size is a fixed parameter. For stability in particular, they presented an integer linear programming formulation to find a stable matching in a given instance, if one exists, in linear time.

In 2020, Bredereck et al. considered another variation of multidimensional roommate games involving either a *master list* or *master poset*, a central list or poset from which all agents’ preference lists are derived [10]. They presented two positive results relating to restrictions of the problem involving a master poset although they showed for either a master list or master poset that finding a stable matching in general remains NP-hard or W[1]-hard, for three very natural parameters.

Other research involving hedonic games with similar constraints has considered Pareto optimality rather than stability [13]; ‘flatmate games’, in which any coalition contains three or fewer agents [9]; and strategic aspects [27].

The template of a hedonic game helps us formalise the extension of SR to three dimensions. In this paper we apply the well-known system of *additively separable preferences* [2]. In a general hedonic game, additive separable preferences are derived from each agent α_i assigning a numerical valuation $val_{\alpha_i}(\alpha_j)$ to every other agent α_j . A preference between two sets is then obtained by comparing the sum of valuations of the agents in each set. This system formalises the system of “ratings” proposed by Huang [19]. In a general hedonic game with additively separable preferences, a core stable partition need not exist, and the associated decision problem is strongly NP-hard [25]. This result holds even when preferences are *symmetric*, meaning that $val_{\alpha_i}(\alpha_j) = val_{\alpha_j}(\alpha_i)$ for any two agents α_i, α_j [3].

The three-dimensional variant of SR that we consider in this paper can also be described as an additively separable hedonic game in which each coalition in a feasible partition has size three. To be consistent with previous research relating to three-dimensional variants of SR [19,21], in this paper we refer to a partition into triples as a *matching* rather than a partition and write *stable matching* rather than *core stable partition*. We finally remark that the usage of the terminology “three-dimensional” to refer to the coalition size rather than, say, the number of agent sets [24], is consistent with previous work in the literature [1,10,21,26].

Our contribution. In this paper we use additively separable preferences to formalise the three-dimensional variant of SR first proposed by Huang in 2007 [19]. The problem model can be equally viewed as a modified hedonic game with additively separable preferences [3,25]. We show that deciding if a stable matching exists is NP-complete, even when valuations are binary (Section 3). In contrast, when valuations are binary and symmetric we show that a stable matching always exists and give an $O(|N|^3)$ algorithm for finding one, where N is the set of agents (Sections 4.1 – 4.4). We believe that this restriction to binary and symmetric preferences has practical as well as theoretical significance. For example, this model could be applied to a social network graph involving a symmetric “friendship” relation between users. Alternatively, in a setting involving real people it might be reasonable for an administrator to remove all asymmetric valuations from the original preferences.

We also consider the notion of *utility* based on agents’ valuations of their partners in a given matching. This leads us to the notion of *utilitarian welfare* [4,11] which is the sum of the utilities of all agents in a given matching. We consider the problem of finding a stable matching with maximum utilitarian welfare given an instance in which valuations are binary and symmetric. We prove that this optimisation problem is NP-hard and provide a novel 2-approximation algorithm (Section 4.5).

We continue in the next section (Section 2) with some preliminary definitions and results.

2 Preliminary definitions and results

Let $N = \{\alpha_1, \dots, \alpha_{|N|}\}$ be a set of *agents*. A *triple* is an unordered set of three agents. A *matching* M comprises a set of pairwise disjoint triples. For any agent α_i , if some triple in M contains α_i then we say that α_i is *matched* and use $M(\alpha_i)$ to refer to that triple. If no triple in M contains α_i then we say that α_i is *unmatched* and write $M(\alpha_i) = \emptyset$. Given a matching M and two distinct agents α_i, α_j , if $M(\alpha_i) = M(\alpha_j)$ then we say that α_j is a *partner* of α_i .

We define *additively separable preferences* as follows. Each agent α_i supplies a *valuation function* $val_{\alpha_i} : N \setminus \{\alpha_i\} \rightarrow \mathbb{Z}$. Given agent α_i , let the *utility* of any set $S \subseteq N$ be $u_{\alpha_i}(S) = \sum_{\alpha_j \in S \setminus \{\alpha_i\}} val_{\alpha_i}(\alpha_j)$. We say that $\alpha_i \in N$ prefers

some triple t_1 to another triple t_2 if $u_{\alpha_i}(t_1) > u_{\alpha_i}(t_2)$. An agent’s preference between two distinct matchings depends only on that agent’s partners in each matching, so given a matching M we write $u_{\alpha_i}(M)$ as shorthand for $u_{\alpha_i}(M(\alpha_i))$. Let $V = \bigcup_{\alpha_i \in N} val_{\alpha_i}$ be the collection of all valuation functions.

Suppose we have some pair (N, V) and a matching M involving the agents in N . We say that a triple $\{\alpha_{k_1}, \alpha_{k_2}, \alpha_{k_3}\}$ *blocks* M in (N, V) if $u_{\alpha_{k_1}}(\{\alpha_{k_2}, \alpha_{k_3}\}) > u_{\alpha_{k_1}}(M)$, $u_{\alpha_{k_2}}(\{\alpha_{k_1}, \alpha_{k_3}\}) > u_{\alpha_{k_2}}(M)$, and $u_{\alpha_{k_3}}(\{\alpha_{k_1}, \alpha_{k_2}\}) > u_{\alpha_{k_3}}(M)$. If no triple in N blocks M in (N, V) then we say that M is *stable* in (N, V) . We say that (N, V) *contains* a stable matching if at least one matching exists in (N, V) that is stable.

We now define the *Three-Dimensional Stable Roommates problem with Additively Separable preferences* (3D-SR-AS). An instance of 3D-SR-AS is given by the pair (N, V) . The problem is to either find a stable matching in (N, V) or report that no stable matching exists. In this paper we consider two different restrictions of this model. The first is when preferences are *binary*, meaning $val_{\alpha_i}(\alpha_j) \in \{0, 1\}$ for any $\alpha_i, \alpha_j \in N$. The second is when preferences are also *symmetric*, meaning $val_{\alpha_i}(\alpha_j) = val_{\alpha_j}(\alpha_i)$ for any $\alpha_i, \alpha_j \in N$.

Lemma 1 illustrates a fundamental property of matchings in instances of 3D-SR-AS. We shall use it extensively in the proofs. Throughout this paper the omitted proofs can be found in the full version [23].

Lemma 1. *Given an instance (N, V) of 3D-SR-AS, suppose that M and M' are matchings in (N, V) . Any triple that blocks M' but does not block M contains at least one agent $\alpha_i \in N$ where $u_{\alpha_i}(M') < u_{\alpha_i}(M)$.*

We also make an observation that unmatched agents may be arbitrarily matched if required. The proof follows from Lemma 1.

Proposition 1. *Suppose we are given an instance (N, V) of 3D-SR-AS. Suppose $|N| = 3k + l$ where $k \geq 0$ and $0 \leq l < 3$. If a stable matching M exists in (N, V) then without loss of generality we may assume that $|M| = k$.*

Finally, some notes on notation: in this paper, we use $L = \langle \dots \rangle$ to construct an ordered list of elements L . If L and L' are lists then we write $L \cdot L'$ meaning the concatenation of L' to the end of L . We also write L_i to mean the i^{th} element of list L , starting from $i = 1$, and $e \in L$ to describe membership of an element e in L . When working with sets of sets, we write $\bigcup S$ to mean $\bigcup_{T \in S} T$.

3 General binary preferences

Let 3D-SR-AS-BIN be the restriction of 3D-SR-AS in which preferences are binary but need not be symmetric. In this section we establish the NP-completeness of deciding whether a stable matching exists, given an instance (N, V) of 3D-SR-AS-BIN.

Theorem 1. *Given an instance of 3D-SR-AS-BIN, the problem of deciding whether a stable matching exists is NP-complete. The result holds even if each agent must be matched.*

Proof sketch. Given an instance (N, V) of 3D-SR-AS-BIN and a matching M , it is straightforward to test in $O(|N|^3)$ time if M is stable in (N, V) . This shows that the decision version of 3D-SR-AS-BIN belongs to the class NP.

We present a polynomial-time reduction from Partition Into Triangles (PIT), which is the following decision problem: “Given a simple undirected graph $G = (W, E)$ where $W = \{w_1, w_2, \dots, w_{3q}\}$ for some integer q , can the vertices of G be partitioned into q disjoint sets $X = \{X_1, X_2, \dots, X_q\}$, each set containing

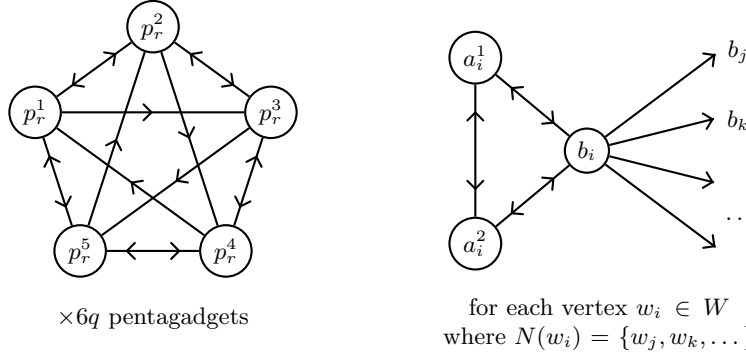


Fig. 1. The reduction from PIT to 3D-SR-AS-BIN. Each vertex represents an agent. An arc is present from agent α_i to agent α_j if $val_{\alpha_i}(\alpha_j) = 1$.

exactly three vertices, such that for each $X_p = \{w_i, w_j, w_k\} \in X$ all three of the edges $\{w_i, w_j\}$, $\{w_i, w_k\}$, and $\{w_j, w_k\}$ belong to E ?" PIT is NP-complete [16].

The reduction from PIT to 3D-SR-AS-BIN is as follows (see Figure 1). Unless otherwise specified assume that $val_{\alpha_i}(\alpha_j) = 0$ for any $\alpha_i, \alpha_j \in N$. For each vertex $w_i \in W$ create agents a_i^1, a_i^2, b_i in N . Then set $val_{a_i^1}(a_i^2) = val_{a_i^1}(b_i) = 1$, $val_{a_i^2}(a_i^1) = val_{a_i^2}(b_i) = 1$, $val_{b_i}(a_i^1) = val_{b_i}(a_i^2) = 1$, and $val_{b_i}(b_j) = 1$ if $\{w_i, w_j\} \in E$ for any $w_j \in N \setminus \{w_i\}$. Next, for each r where $1 \leq r \leq 6q$ create $p_r^1, p_r^2, p_r^3, p_r^4, p_r^5$ in N . Then set $val_{p_r^1}(p_r^2) = val_{p_r^1}(p_r^3) = val_{p_r^1}(p_r^5) = 1$, $val_{p_r^2}(p_r^3) = val_{p_r^2}(p_r^4) = val_{p_r^2}(p_r^1) = 1$, $val_{p_r^3}(p_r^4) = val_{p_r^3}(p_r^5) = val_{p_r^3}(p_r^2) = 1$, $val_{p_r^4}(p_r^5) = val_{p_r^4}(p_r^1) = val_{p_r^4}(p_r^3) = 1$, and $val_{p_r^5}(p_r^1) = val_{p_r^5}(p_r^2) = val_{p_r^5}(p_r^4) = 1$. We shall refer to $\{p_r^1, \dots, p_r^5\}$ as the r^{th} *pentagadget*. Note that $|N| = 39q$. In the full proof of this theorem, contained in [23], we show that a partition into triangles X exists in $G = (W, E)$ if and only if a stable matching M exists in (N, V) where $|M| = |N|/3$. \square

4 Symmetric binary preferences

Consider the restriction of 3D-SR-AS in which preferences are binary and symmetric, which we call *3D-SR-SAS-BIN*. In this section we show that every instance of 3D-SR-SAS-BIN admits a stable matching. We give a step-by-step constructive proof of this result between Sections 4.1 – 4.4, leading to an $O(|N|^3)$ algorithm for finding a stable matching. In Section 4.5 we consider an optimisation problem related to 3D-SR-SAS-BIN.

4.1 Preliminaries

An instance (N, V) of 3D-SR-SAS-BIN corresponds to a simple undirected graph $G = (N, E)$ where $\{\alpha_i, \alpha_j\} \in E$ if $val_{\alpha_i}(\alpha_j) = 1$, which we refer to as the *underlying graph*.

We introduce a restricted type of matching called a *P-matching*. Recall that by definition, $M(\alpha_p) = \emptyset$ implies that $u_{\alpha_p}(M) = 0$ for any $\alpha_p \in N$ in an arbitrary matching M . We say that a matching M in (N, V) is a *P-matching* if $M(\alpha_p) \neq \emptyset$ implies $u_{\alpha_p}(M) > 0$.

It follows that a *P-matching* corresponds to a $\{K_3, P_3\}$ -packing in the underlying graph [22]. Note that any triple in a *P-matching* M must contain some agent with utility two. A *stable P-matching* is a *P-matching* that is also stable. We will eventually show that any instance of 3D-SR-SAS-BIN contains a stable *P-matching*.

In an instance (N, V) of 3D-SR-SAS-BIN, a *triangle* comprises three agents $\alpha_{m_1}, \alpha_{m_2}, \alpha_{m_3}$ such that $val_{\alpha_{m_1}}(\alpha_{m_2}) = val_{\alpha_{m_2}}(\alpha_{m_3}) = val_{\alpha_{m_3}}(\alpha_{m_1}) = 1$. If (N, V) contains no triangle then we say it is *triangle-free*. If (N, V) is not triangle-free then it can be reduced by successively removing three agents that belong to a triangle until it is triangle-free. This operation corresponds to removing a *maximal triangle packing* (see [12, 22]) in the underlying graph and can be performed in $O(|N|^3)$ time. The resulting instance is triangle-free. We summarise this observation in the following lemma.

Lemma 2. *Given an instance (N, V) of 3D-SR-SAS-BIN, we can identify an instance (N', V') of 3D-SR-SAS-BIN and a set of triples M_Δ in $O(|N|^3)$ time such that (N', V') is triangle-free, $|N'| \leq |N|$, and if M is a stable *P-matching* in (N', V') then $M' = M \cup M_\Delta$ is a stable *P-matching* in (N, V) .*

4.2 Repairing a *P-matching* in a triangle-free instance

In this section we consider an arbitrary triangle-free instance (N, V) of 3D-SR-SAS-BIN. Since the only instance referred to in this section is (N, V) so here we shorten “is stable in (N, V) ” to “is stable”, or similar.

We first define a special type of *P-matching* which is ‘repairable’. We then present Algorithm **repair** (Algorithm 1), which, given (N, V) and a ‘repairable’ *P-matching* M , constructs a new *P-matching* M' that is stable. We shall see in the next section how this relates to a more general algorithm that, given a triangle-free instance, constructs a *P-matching* that is stable in that instance.

Given a triangle-free instance (N, V) , we say a *P-matching* M is *repairable* if it is not stable and there exists exactly one $\alpha_i \in N$ where $u_{\alpha_i}(M) = 0$ and any triple that blocks M comprises $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$ for some $\alpha_{j_1}, \alpha_{j_2} \in N$ where $u_{\alpha_{j_1}}(M) = 1$, $u_{\alpha_{j_2}}(M) = 0$, and $val_{\alpha_i}(\alpha_{j_1}) = val_{\alpha_{j_1}}(\alpha_{j_2}) = 1$.

We now provide some intuition behind Algorithm **repair** and refer the reader to Figure 2. Recall that the overall goal of the algorithm is to construct a stable *P-matching* M' . Since the given *P-matching* M is repairable, our aim will be to modify M such that $u_{\alpha_i}(M') \geq 1$ while ensuring that no three agents that are ordered to different triples in M' block M' . The stability of the constructed *P-matching* M' then follows.

The algorithm begins by selecting some triple $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$ that blocks M . The two agents in $M(\alpha_{j_1}) \setminus \{\alpha_{j_1}\}$ are labelled α_{j_3} and α_{j_4} . We present two example scenarios in which it is possible to construct a stable *P-matching*. First,

suppose there exists some α_{z_1} where $val_{\alpha_{j_3}}(\alpha_{z_1}) = 1$ and $u_{\alpha_{z_1}}(M) = 0$. Construct M' from M by removing $\{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$ and adding $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$ and $\{\alpha_{j_3}, \alpha_{j_4}, \alpha_{z_1}\}$. Now, $u_{\alpha_i}(M') = 1$ and $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N \setminus \{\alpha_i\}$. It follows by Lemma 1 that M' is stable. Second, suppose there exists no such α_{z_1} but there exists some α_{z_2} where $val_{\alpha_{j_4}}(\alpha_{z_2}) = 1$ and $u_{\alpha_{z_2}}(M) = 0$. Now construct M' from M by removing $\{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$ and adding $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$ and $\{\alpha_{j_3}, \alpha_{j_4}, \alpha_{z_2}\}$. Note that $u_{\alpha_i}(M') = 1$ and $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N \setminus \{\alpha_i, \alpha_{j_3}\}$. It can be shown that α_{j_3} does not belong to a triple that blocks M' since no α_{z_1} exists as described. It follows again by Lemma 1 that M' is stable. Generalising the approach in the two example scenarios, the algorithm constructs a list S of agents, which initially comprises $\langle \alpha_{j_1}, \alpha_{j_3}, \alpha_{j_4} \rangle$. The list S has length $3c$ for some $c \geq 1$, where $\{S_{3c-2}, S_{3c-1}, S_{3c}\} \in M$ and $val_{S_p}(S_{p+1}) = 1$ for each p ($1 \leq p < 3c$). The list S therefore corresponds to a path in the underlying graph. In each iteration of the main loop, three agents

Algorithm 1 Algorithm repair

Input: a triangle-free instance (N, V) of 3D-SR-SAS-BIN, repairable P -matching M in (N, V) (Section 4.2) with some such $\alpha_i \in N$.

Output: stable P -matching M' in (N, V)

```

 $\{\alpha_{j_1}, \alpha_{j_2}\} \leftarrow$  some  $\alpha_{j_1}, \alpha_{j_2} \in N$  where  $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$  blocks  $M$  and  $u_{\alpha_{j_1}}(M) = 1$ 
 $\{\alpha_{j_3}, \alpha_{j_4}\} \leftarrow M(\alpha_{j_1}) \setminus \{\alpha_{j_1}\}$  where  $u_{\alpha_{j_3}}(M) = 2$ 
 $S \leftarrow \langle \alpha_{j_1}, \alpha_{j_3}, \alpha_{j_4} \rangle$ 
 $c \leftarrow 1$ 
 $b \leftarrow 0$ 
 $\alpha_{z_1}, \alpha_{z_2}, \alpha_{y_1}, \alpha_{y_2}, \alpha_{w_1} \leftarrow \perp$ 
while true
   $\alpha_{z_1} \leftarrow$  some  $\alpha_{z_1} \in N \setminus \{\alpha_i\}$  where  $val_{\alpha_{z_1}}(S_{3c-1}) = 1$  and  $u_{\alpha_{z_1}}(M) = 0$ , else  $\perp$ 
   $\alpha_{z_2} \leftarrow$  some  $\alpha_{z_2} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$  where  $val_{\alpha_{z_2}}(S_{3c}) = 1$  and  $u_{\alpha_{z_2}}(M) = 0$ , else  $\perp$ 
   $\alpha_{y_1} \leftarrow$  some  $\alpha_{y_1} \in N$  where  $val_{S_{3c}}(\alpha_i) = val_{\alpha_{y_1}}(\alpha_i) = 1$  and  $u_{\alpha_{y_1}}(M) = 0$ , else  $\perp$ 
   $\alpha_{y_2} \leftarrow$  some  $\alpha_{y_2} \in N$  where  $val_{S_{3c}}(\alpha_{j_2}) = val_{\alpha_{y_2}}(\alpha_{j_2}) = 1$  and  $u_{\alpha_{y_2}}(M) = 0$ , else  $\perp$ 
   $b \leftarrow$  some  $1 \leq b < c$  where  $val_{S_{3b}}(\alpha_{j_2}) = val_{S_{3c}}(S_{3b}) = 1$ , else 0
   $\alpha_{w_1} \leftarrow$  some  $\alpha_{w_1} \in N$  where  $val_{S_{3c}}(\alpha_{w_1}) = 1$ ,  $u_{\alpha_{w_1}}(M) = 1$  and  $\alpha_{w_1} \notin S$ 
    and there exists some  $\alpha_{z_3} \in N \setminus \{\alpha_i\}$  where  $val_{\alpha_{w_1}}(\alpha_{z_3}) = 1$  and  $u_{\alpha_{z_3}}(M) = 0$ ,
    else  $\perp$ 
  if  $\alpha_{z_1} \neq \perp$  or  $\alpha_{z_2} \neq \perp$  or  $\alpha_{y_1} \neq \perp$  or  $\alpha_{y_2} \neq \perp$  or  $b > 0$  or  $\alpha_{w_1} = \perp$  then
    break
  else
     $\{\alpha_{w_2}, \alpha_{w_3}\} \leftarrow M(\alpha_{w_1}) \setminus \{\alpha_{w_1}\}$  where  $u_{\alpha_{w_2}}(M) = 2$ 
     $S \leftarrow S \cdot \langle \alpha_{w_1}, \alpha_{w_2}, \alpha_{w_3} \rangle$ 
     $c \leftarrow c + 1$ 
  end if
end while

```

continued overleaf

Algorithm 1 Algorithm repair*continued from previous page***if** $\alpha_{z_1} \neq \perp$ and $\alpha_{z_1} \neq \alpha_{j_2}$ **then**

▷ Case 1

$$M_S \leftarrow \{\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}\} \cup \bigcup_{1 \leq d < c} \{\{S_{3d-1}, S_{3d}, S_{3d+1}\}\} \cup \{\{\alpha_{z_1}, S_{3c-1}, S_{3c}\}\}$$

else if $\alpha_{z_2} \neq \perp$ **then**

▷ Case 2

$$M_S \leftarrow \{\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}\} \cup \bigcup_{1 \leq d < c} \{\{S_{3d-1}, S_{3d}, S_{3d+1}\}\} \cup \{\{S_{3c-1}, S_{3c}, \alpha_{z_2}\}\}$$

else if $\alpha_{z_1} \neq \perp$ and $\alpha_{z_1} = \alpha_{j_2}$ **then**

▷ Case 3

$$\alpha_{z_4} \leftarrow \text{some } \alpha_{z_4} \in N \setminus \{\alpha_i, \alpha_{j_2}\} \text{ where } \text{val}_{S_{3c-2}}(\alpha_{z_4}) = 1 \text{ and } u_{\alpha_{z_4}}(M) = 0$$

$$M_S \leftarrow \{\{\alpha_i, \alpha_{j_1}, \alpha_{j_3}\}\} \cup \bigcup_{1 \leq d < c-1} \{\{S_{3d}, S_{3d+1}, S_{3d+2}\}\} \cup \{\{S_{3c-3}, S_{3c-2}, \alpha_{z_4}\}\} \\ \cup \{\{S_{3c-1}, S_{3c}, \alpha_{j_2}\}\}$$

else if $\alpha_{y_1} \neq \perp$ **then**

▷ Case 4

$$M_S \leftarrow \{\{\alpha_{j_2}, \alpha_{j_1}, \alpha_{j_3}\}\} \cup \bigcup_{1 \leq d < c} \{\{S_{3d}, S_{3d+1}, S_{3d+2}\}\} \cup \{\{S_{3c}, \alpha_i, \alpha_{y_1}\}\}$$

else if $\alpha_{y_2} \neq \perp$ **then**

▷ Case 5

$$M_S \leftarrow \{\{\alpha_i, \alpha_{j_1}, \alpha_{j_3}\}\} \cup \bigcup_{1 \leq d < c} \{\{S_{3d}, S_{3d+1}, S_{3d+2}\}\} \cup \{\{S_{3c}, \alpha_{j_2}, \alpha_{y_2}\}\}$$

else if $b > 0$ **then**

▷ Case 6

$$\alpha_{z_5} \leftarrow \text{some } \alpha_{z_5} \in N \setminus \{\alpha_i, \alpha_{j_2}\} \text{ where } \text{val}_{S_{3b+1}}(\alpha_{z_5}) = 1 \text{ and } u_{\alpha_{z_5}}(M) = 0$$

$$M_S \leftarrow \{\{\alpha_i, \alpha_{j_1}, \alpha_{j_3}\}\} \cup \bigcup_{1 \leq d < b} \{\{S_{3d}, S_{3d+1}, S_{3d+2}\}\} \cup \{\{\alpha_{z_4}, S_{3b+1}, S_{3b+2}\}\} \\ \cup \bigcup_{b+1 \leq d < c} \{\{S_{3d}, S_{3d+1}, S_{3d+2}\}\} \cup \{\{S_{3c}, S_{3b}, \alpha_{j_2}\}\}$$

else▷ Case 7. Note that $\alpha_{w_1} = \perp$.

$$M_S \leftarrow \{\{\alpha_i, \alpha_{j_1}, \alpha_{j_3}\}\} \cup \bigcup_{1 \leq d < c} \{\{S_{3d}, S_{3d+1}, S_{3d+2}\}\}$$

end if**return** $M' = M_S \cup \{r \in M \mid r \cap S = \emptyset\}$

belonging to some triple in M are appended to the end of S . The loop continues until S satisfies at least one of six specific conditions. We show that eventually at least one of these conditions must hold.

These six stopping conditions correspond to seven different cases, labelled Case 1 – Case 7, in which a stable P -matching M' may be constructed. The exact construction of M' depends on which condition(s) caused the main loop to terminate. Cases 1 and 3 generalise the first example scenario, in which some α_{z_1} exists as described. Case 2 generalises the second example scenario, in which no such α_{z_1} exists but some α_{z_2} exists as described. Cases 4 – 7 correspond to similar scenarios. The six stopping conditions and seven corresponding constructions of

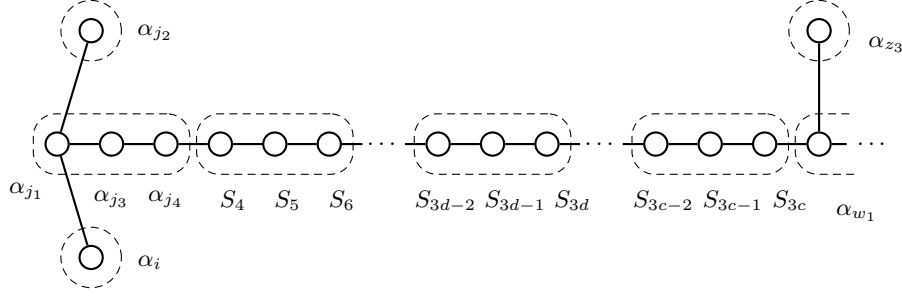


Fig. 2. Players and triples in M before a new iteration of the **while** loop

M' are somewhat hierarchical. For example, the proof that M' is stable in Case 4 relies on the fact that in no iteration did the condition for Cases 1 and 3 hold. A similar reliance exists in the proofs of each of the other cases. The proof that M' is stable in Case 7 is the most complex. It relies on the fact that no condition relating to any of the previous six cases held in the final or some previous iteration of the main loop. Further intuition for the different cases is given in the full version of this paper [23].

Algorithm **repair** is presented in Algorithm 1 in two parts. The first part involves the construction of S and exploration of the instance. The second part involves the construction of M' . The following lemma establishes the correctness and complexity of this algorithm.

Lemma 3. *Algorithm **repair** returns a stable P -matching in $O(|N|^2)$ time.*

4.3 Finding a stable P -matching in a triangle-free instance

In the previous section we supposed that (N, V) was a triangle-free instance of 3D-SR-SAS-BIN and considered a P -matching M that was repairable (Section 4.2). We presented Algorithm **repair**, which can be used to construct a stable P -matching M' in $O(|N|^2)$ time (Lemma 3). In this section we present Algorithm **findStableInTriangleFree** (Algorithm 2), which, given a triangle-free instance (N, V) , constructs a P -matching M' that is stable in (N, V) . Algorithm **findStableInTriangleFree** is recursive. The algorithm first removes an arbitrary agent α_i to construct a smaller instance (N', V') . It then uses a recursive call to construct a P -matching M that is stable in (N', V') . By Lemma 1, any triple that blocks M in the larger instance (N, V) must contain α_i or block M in (N', V') . There are then three cases involving types of triple that block M in (N', V') . In two out of three cases, M' can be constructed by adding to M a new triple containing α_i and two players unmatched in M . In the third case, M is not stable in (N, V) but, by design, is repairable (see Section 4.2). It follows that Algorithm **repair** can be used to construct a P -matching that is stable in (N, V) (Lemma 3). It is relatively straightforward to show that the running time of Algorithm **findStableInTriangleFree** is $O(|N|^3)$.

Algorithm 2 Algorithm `findStableInTriangleFree`**Input:** an instance (N, V) of 3D-SR-SAS-BIN**Output:** stable P -matching M' in (N, V)

```

if  $|N| = 2$  then return  $\emptyset$ 
 $\alpha_i \leftarrow$  an arbitrary agent in  $N$ 
 $(N', V') \leftarrow (N \setminus \{\alpha_i\}, V \setminus \{val_{\alpha_i}\})$ 
 $M \leftarrow \text{findStableInTriangleFree}((N', V'))$ 
if some  $\alpha_{l_1}, \alpha_{l_2} \in N$  exist where  $u_{\alpha_{l_1}}(M) = u_{\alpha_{l_2}}(M) = 0$ 
  and  $val_{\alpha_i}(\alpha_{l_1}) = val_{\alpha_i}(\alpha_{l_2}) = 1$  then
    return  $M \cup \{\{\alpha_i, \alpha_{l_1}, \alpha_{l_2}\}\}$ 
else if some  $\alpha_{l_3}, \alpha_{l_4} \in N$  exist where  $u_{\alpha_{l_3}}(M) = u_{\alpha_{l_4}}(M) = 0$ 
  and  $val_{\alpha_i}(\alpha_{l_3}) = val_{\alpha_i}(\alpha_{l_4}) = 1$  then
    return  $M \cup \{\{\alpha_i, \alpha_{l_3}, \alpha_{l_4}\}\}$ 
else if some  $\alpha_{l_5}, \alpha_{l_6} \in N$  exist where  $u_{\alpha_{l_5}}(M) = 1, u_{\alpha_{l_6}}(M) = 0$ 
  and  $val_{\alpha_i}(\alpha_{l_5}) = val_{\alpha_i}(\alpha_{l_6}) = 1$  then
     $\triangleright M$  is repairable in  $(N, V)$  (see Section 4.2). Note that  $\alpha_{j_1} = \alpha_{l_5}$  and  $\alpha_{j_2} = \alpha_{l_6}$ .
    return repair $((N, V), M, \alpha_i)$ 
else
  return  $M$ 
end if

```

Lemma 4. *Algorithm `findStableInTriangleFree` returns a stable P -matching in (N, V) in $O(|N|^3)$ time.*

4.4 Finding a stable P -matching in an arbitrary instance

In the previous section we considered instances of 3D-SR-SAS-BIN that are triangle-free. We showed that, given such an instance, Algorithm `findStableInTriangleFree` can be used to find a stable P -matching in $O(|N|^3)$ time (Lemma 4). In Section 4.1, we showed that an arbitrary instance can be reduced in $O(|N|^3)$ time to construct a corresponding triangle-free instance (Lemma 2). Algorithm `findStable` therefore comprises two steps. First, the instance is reduced by removing a maximal set of triangles. Call this set M_Δ . Then, Algorithm `findStableInTriangleFree` is called to construct a P -matching M' that is stable in the reduced, triangle-free instance. It is straightforward to show that $M_\Delta \cup M'$ is a stable P -matching. The running time of Algorithm `findStable` is thus $O(|N|^3)$. A pseudocode description of Algorithm `findStable` can be found in the full version of this paper [23]. We arrive at the following result.

Theorem 2. *Given an instance (N, V) of 3D-SR-SAS-BIN, a stable P -matching, and hence a stable matching, must exist and can be found in $O(|N|^3)$ time. Moreover, if $|N|$ is a multiple of three then, if required, every agent can be matched in the returned stable matching.*

4.5 Stability and utilitarian welfare

Given an instance (N, V) of 3D-SR-SAS-BIN and matching M , let the *utilitarian welfare* [4,11] of a set $S \subseteq N$, denoted $u_S(M)$, be $\sum_{\alpha_i \in S} u_{\alpha_i}(M)$. Let $u(M)$ be short for $u_N(M)$. Given a matching M in an arbitrary instance (N, V) of 3D-SR-SAS-BIN, it follows that $0 \leq u(M) \leq 2|N|$. It is natural to then consider the optimisation problem of finding a stable matching with maximum utilitarian welfare, which we refer to as *3D-SR-SAS-BIN-MAXUW*. This problem is closely related to Partition Into Triangles (PIT, defined in Section 3), which we reduce from in the proof that 3D-SR-SAS-BIN-MAXUW is NP-hard.

Theorem 3. *3D-SR-SAS-BIN-MAXUW is NP-hard.*

We note that the reduction from PIT to 3D-SR-SAS-BIN-MAXUW also shows that the problem of finding a (not-necessarily stable) matching with maximum utilitarian welfare, given an instance of 3D-SR-SAS-BIN, is also NP-hard.

In Section 4.4 we showed that, given an arbitrary instance (N, V) of 3D-SR-SAS-BIN, a stable P -matching exists and can be found in $O(|N|^3)$ time. We now present Algorithm `findStableUW` (Algorithm 3) as an approximation algorithm for 3D-SR-SAS-BIN-MAXUW.

This algorithm first calls Algorithm `findStable` to construct a stable P -matching. It then orders the unmatched agents into triples such that the produced matching is still stable in (N, V) (by Lemma 1) but is not necessarily a P -matching.

Algorithm 3 Algorithm `findStableUW`

Input: an instance (N, V) of 3D-SR-SAS-BIN

Output: stable matching M_A in (N, V)

$M_1 \leftarrow \text{findStable}((N, V))$

$U \leftarrow$ agents in N unmatched in M_1

$Y \leftarrow \text{maximum2DMatching}((N, V), U)$

if $|Y| \geq \lfloor |U|/3 \rfloor$ **then**

$X \leftarrow$ any $\lfloor |U|/3 \rfloor$ elements of Y

else

▷ Note that since Y is a set of disjoint pairs, it follows that

$$|U \setminus \bigcup Y| = |U| - 2|Y| \geq \lfloor |U|/3 \rfloor - |Y|.$$

$W \leftarrow$ an arbitrary set of $\lfloor |U|/3 \rfloor - |Y|$ pairs of elements in $U \setminus \bigcup Y$

$X \leftarrow Y \cup W$

end if

$Z \leftarrow U \setminus \bigcup X$

▷ Suppose $X = \{x_1, x_2, \dots, x_{\lfloor |U|/3 \rfloor}\}$ and $Z = \{z_1, z_2, \dots, z_{\lfloor |U|/3 \rfloor}\}$.

▷ Note that x_i is a pair of agents and z_i is a single agent for each $1 \leq i \leq \lfloor |U|/3 \rfloor$.

$M_2 \leftarrow \{x_i \cup \{z_i\} \text{ for each } 1 \leq i \leq \lfloor |U|/3 \rfloor\}$

return $M_1 \cup M_2$

The pseudocode description of Algorithm **findStableUW** includes a call to **maximum2DMatching**. Given an instance (N, V) and some set $U \subseteq N$, this subroutine returns a (two-dimensional) *maximum cardinality matching* Y in the subgraph of G , the underlying graph of (N, V) , induced by U . From Y , Algorithm **findStableUW** constructs a set X of pairs with cardinality $\lfloor |U|/3 \rfloor$. It also constructs a set Z from the remaining agents, also with cardinality $\lfloor |U|/3 \rfloor$. Finally, it constructs the matching M_2 such that each triple in M_2 is union of a pair of agents in X and a single agent in Z . Let M_A be an arbitrary matching returned by Algorithm **findStableUW** given (N, V) . Suppose M_{opt} is a stable matching in (N, V) with maximum utilitarian welfare. To prove the performance guarantee of Algorithm **findStableUW** we show that $2u(M_A) \geq u(M_{\text{opt}})$. The proof involves apportioning the welfare of agents in M_A by the triples of those agents in M_{opt} .

Theorem 4. *Algorithm **findStableUW** is a 2-approximation algorithm for 3D-SR-SAS-BIN-MAXUW.*

In the instance of 3D-SR-SAS-BIN shown in Figure 3, Algorithm **findStableUW** always returns $M_A = \{\{\alpha_3, \alpha_5, \alpha_6\}\}$ while $M_{\text{opt}} = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_8\}, \{\alpha_6, \alpha_7, \alpha_9\}\}$. Since $u(M_A) = 6$ and $u(M_{\text{opt}}) = 12$ it follows that $u(M_{\text{opt}}) = 2u(M_A)$. This shows that the analysis of Algorithm **findStableUW** is tight. Moreover, this particular instance shows that any approximation algorithm with a better performance ratio than 2 should not always begin, like Algorithm **findStableUW** does, by selecting a maximal set of triangles.

5 Open questions

In this paper we have considered the three-dimensional stable roommates problem with additively separable preferences. We considered the special cases in which preferences are binary but not necessarily symmetric, and both binary and symmetric. There are several interesting directions for future research.

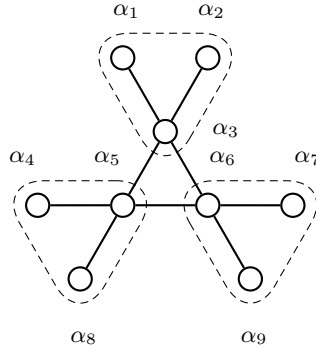


Fig. 3. An instance in which $u(M_{\text{opt}}) = 2u(M_A)$.

- Does there exist an approximation algorithm for 3D-SR-SAS-BIN-MAXUW (Section 4.5) with a better performance guarantee than 2?
- In 3D-SR-AS, there are numerous possible restrictions besides symmetric and binary preferences. Do any other restrictions ensure that a stable matching exists? For example, we could consider the restriction in which preferences are symmetric and $val_{\alpha_i} \in \{0, 1, 2\}$ for each $\alpha_i \in N$.
- Additively separable preferences are one possible structure of agents' preferences that can be applied in a model of three-dimensional SR. Are there other systems of preferences that result in new models in which a stable matching can be found in polynomial time?
- The 3D-SR-AS problem model can be generalised to higher dimensions. It would be natural to ask if the algorithm for 3D-SR-SAS-BIN can be generalised to the same problem in $k \geq 3$ dimensions, in which a k -set of agents S is blocking if, for each of the k agents in S , the utility of S is strictly greater than that agent's utility in the matching. We conjecture that when $k \geq 4$, a stable matching need not exist, and that the associated decision problem is NP-complete, even when preferences are both binary and symmetric.

References

1. Arkin, E., Bae, S., Efrat, A., Mitchell, J., Okamoto, K., Polishchuk, V.: Geometric Stable Roommates. *Information Processing Letters* **109**, 219–224 (2009)
2. Aziz, H., Brandt, F., Seedig, H.G.: Optimal partitions in additively separable hedonic games. In: *Proceedings of IJCAI '11: the 22nd International Joint Conference on Artificial Intelligence - Volume One*. pp. 43–48. AAAI Press (2011)
3. Aziz, H., Brandt, F., Seedig, H.G.: Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence* **195**, 316–334 (2013)
4. Aziz, H., Gaspers, S., Gudmundsson, J., Mestre, J., Täubig, H.: Welfare maximization in fractional hedonic games. In: *Proceedings of IJCAI '15: the 24th International Joint Conference on Artificial Intelligence*. pp. 461–467. AAAI Press (2015)
5. Aziz, H., Lang, J., Monnot, J.: Computing Pareto optimal committees. In: *Proceedings of IJCAI '16: the 25th International Joint Conference on Artificial Intelligence*. pp. 60–66. AAAI Press (2016)
6. Aziz, H., Savani, R., Moulin, H.: Hedonic games. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D. (eds.) *Handbook of Computational Social Choice*, p. 356–376. Cambridge University Press (2016)
7. Barberà, S., Bossert, W., Pattanaik, P.: Ranking sets of objects. In: Barberà, S., Hammond, P., Seidl, C. (eds.) *Handbook of Utility Theory*, vol. 2, chap. 17, pp. 893–977. Kluwer Academic Publishers (2004)
8. Boehmer, N., Elkind, E.: Stable roommate problem with diversity preferences. In: *Proceedings of IJCAI '20: the 29th International Joint Conference on Artificial Intelligence*. pp. 96–102. IJCAI Organization (2020)
9. Brandt, F., Bullinger, M.: Finding and recognizing popular coalition structures. In: *Proceedings of AAMAS '20: the 19th International Conference on Autonomous Agents and Multiagent Systems*. pp. 195–203. IFAAMAS (2020)
10. Brederbeck, R., Heeger, K., Knop, D., Niedermeier, R.: Multidimensional stable roommates with master list. In: *Proceedings of WINE '20: The 16th Conference*

- on Web and Internet Economics. Lecture Notes in Computer Science, vol. 12495, pp. 59–73. Springer (2020)
11. Bullinger, M.: Pareto-optimality in cardinal hedonic games. In: Proceedings of AAMAS '20: the 19th International Conference on Autonomous Agents and Multiagent Systems. pp. 213–221. IFAAMAS (2020)
 12. Chataigner, F., Manić, G., Wakabayashi, Y., Yuster, R.: Approximation algorithms and hardness results for the clique packing problem. *Discrete Applied Mathematics* **157**(7), 1396–1406 (2009)
 13. Cseh, Á., Fleiner, T., Harján, P.: Pareto optimal coalitions of fixed size. *Journal of Mechanism and Institution Design* **4**(1), 87–108 (2019)
 14. Deineko, V.G., Woeginger, G.J.: Two hardness results for core stability in hedonic coalition formation games. *Discrete Applied Mathematics* **161**(13), 1837–1842 (2013)
 15. Gale, D., Shapley, L.: College admissions and the stability of marriage. *American Mathematical Monthly* **69**, 9–15 (1962)
 16. Garey, M., Johnson, D.: *Computers and Intractability*. Freeman, San Francisco, CA. (1979)
 17. Hajduková, J.: Coalition formation games: a survey. *International Game Theory Review* **8**(4), 613–641 (2006)
 18. Huang, C.C.: Two’s company, three’s a crowd: Stable family and threesome roommates problems. In: Proceedings of ESA’07: the 15th European Symposium on Algorithms. Lecture Notes in Computer Science, vol. 4698, pp. 558–569. Springer (2007)
 19. Huang, C.C.: Two’s company, three’s a crowd: Stable family and threesome roommates problems. Computer Science Technical Report TR2007-598, Dartmouth College (2007)
 20. Irving, R.: An Efficient Algorithm for the “Stable Roommates” Problem. *Journal of Algorithms* **6**, 577–595 (1985)
 21. Iwama, K., Miyazaki, S., Okamoto, K.: Stable roommates problem with triple rooms. In: Proceedings of WAAC ’07: the 10th Korea-Japan Workshop on Algorithms and Computation. pp. 105–112 (2007)
 22. Kirkpatrick, D.G., Hell, P.: On the complexity of general graph factor problems. *siam* **12**(3), 601–609 (1983)
 23. McKay, M., Manlove, D.: The Three-Dimensional Stable Roommates Problem with Additively Separable Preferences, preprint arXiv:2107.04368 [cs.GT], available from <https://arxiv.org/abs/2107.04368> (2021)
 24. Ng, C., Hirschberg, D.: Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics* **4**(2), 245–252 (1991)
 25. Sung, S.C., Dimitrov, D.: Computational complexity in additive hedonic games. *European Journal of Operational Research* **203**(3), 635–639 (2010)
 26. Woeginger, G.J.: Core stability in hedonic coalition formation. In: Proceedings of SOFSEM ’13: the 39th International Conference on Current Trends in Theory and Practice of Computer Science. Lecture Notes in Computer Science, vol. 7741, pp. 33–50. Springer (2013)
 27. Wright, M., Vorobeychik, Y.: Mechanism design for team formation. In: Proceedings of the 29th Conference on Artificial Intelligence. pp. 1050–1056. AAAI ’15, AAAI Press (2015)