

ALGORITHMIC ASPECTS OF FIXED-SIZE COALITION FORMATION

MICHAEL MCKAY

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING SCIENCE
COLLEGE OF SCIENCE AND ENGINEERING



University
of Glasgow

16TH JANUARY 2023

© MICHAEL MCKAY 2023

Abstract

We study algorithmic aspects of models in which a set of agents is to be organised into coalitions of a fixed size. Such models can be viewed as a type of hedonic game, coalition formation game, or multidimensional matching problem. We mostly consider models in which coalitions have size three and are formalisms of *Three-Dimensional Roommates* (3DR). Models of 3DR are characterised by a combination of the system by which agents have preferences over coalitions, and the solution concept (e.g. stability). Since the computational problems associated with 3DR are typically hard, we consider approximate solutions and restricted cases, with the aim of characterising the boundary between tractable and intractable variants.

Part of our contribution relates to two new models of 3DR, which involve two existing systems of preferences called \mathcal{B} - and \mathcal{W} -preferences. In each model, we consider the existence of matchings that are stable. We show that the related decision problems are NP-complete and devise approximation algorithms for corresponding optimisation problems.

In a model of 3DR with additively separable preferences, we investigate stable matchings and envy-free matchings, for three successively weaker definitions of envy-freeness. We consider restrictions on the agents' preferences including symmetric, binary, and ternary valuations. We identify dichotomies based on these restrictions and provide a comprehensive complexity classification. Interestingly, we identify a general trend that, for successively weaker solution concepts, existence and polynomial-time solvability hold under successively weaker preference restrictions.

We also consider a variant of 3DR known as *Three-Dimensional Stable Matching with Cyclic preferences* (3-DSM-CYC), which has been of independent interest. It was recently shown that finding a stable matching is NP-hard, so we consider a related optimisation problem and present an approximation algorithm based on serial dictatorship. We also consider a situation in which the preferences of some agents are sufficiently similar to some master list, and show that the approximation ratio of this algorithm can be improved in relation to a specific similarity measure.

Finally, we consider a problem in graph theory that generalises the notion of assigning agents to coalitions of a fixed size. Rather than organising a set of agents, the problem is to find a

maximum-cardinality set of r -cliques in an undirected graph subject to that set being either vertex disjoint or edge disjoint, for a fixed integer $r \geq 3$. This general problem is known as the *K_r -packing problem*. Here we study the restriction of this problem in which the graph has a fixed maximum degree Δ . It is known for $r = 3$ that the vertex-disjoint (edge-disjoint) variant is solvable in linear time if $\Delta = 3$ ($\Delta = 4$) but APX-hard if $\Delta \geq 4$ ($\Delta \geq 5$). We generalise these results to an arbitrary but fixed $r \geq 3$, and provide a full complexity classification for both the vertex- and edge-disjoint variants in graphs of maximum degree Δ , for all $r \geq 3$.

Table of Contents

1	Introduction	1
1.1	About	1
1.2	Thesis statement	2
1.3	Contribution	2
1.4	Structure	5
2	Literature review	6
2.1	Introduction	6
2.2	Hedonic games and stable partitions	7
2.3	Coalitions of restricted size	13
2.3.1	Two-dimensional matching and roommates	13
2.3.2	Multidimensional matching of a multipartite agent set	15
2.3.3	Multidimensional roommates	19
2.3.4	Other models	21
3	Approximability of Three-Dimensional Stable Matching with Cyclic Preferences	24
3.1	Introduction	24
3.2	Unrestricted preferences	25
3.3	Preferences derived from a master list	31
3.4	Summary and open problems	36
4	Three-Dimensional Stable Roommates with \mathcal{B}-preferences	38
4.1	Introduction	38
4.2	Deciding existence	39

4.3	Approximation	44
4.4	Towards a minimal “no” instance	50
4.5	Summary and open problems	52
5	Three-Dimensional Stable Roommates with \mathcal{W}-preferences	55
5.1	Introduction	55
5.2	Deciding existence	56
5.3	Approximation	60
5.4	Summary and open problems	61
6	Three-Dimensional Stable Roommates with Additively Separable Preferences	63
6.1	Introduction	63
6.2	Preliminaries	64
6.3	Symmetric binary preferences	65
6.3.1	Finding a stable matching	65
6.3.2	Maximising utilitarian welfare	84
6.4	Binary preferences	89
6.5	Symmetric ternary preferences	93
6.6	Summary and open problems	97
7	Three-Dimensional Envy-Free Roommates with Additively Separable Preferences	100
7.1	Introduction	100
7.2	Envy-freeness	102
7.2.1	Symmetric binary preferences with maximum degree two	102
7.2.2	Symmetric binary preferences with maximum degree three	104
7.3	Weakly justified envy-freeness	109
7.3.1	Symmetric binary preferences with maximum degree two	109
7.3.2	Symmetric binary preferences with maximum degree three	122
7.4	Justified envy-freeness	128
7.4.1	Binary preferences	128
7.4.2	Ternary preferences	130

7.4.3	Symmetric non-binary preferences	135
7.5	Summary and open problems	154
8	The K_r-packing Problem in bounded degree graphs	157
8.1	Introduction	157
8.1.1	Background	157
8.1.2	Our contribution	159
8.1.3	Preliminaries	160
8.2	Linear-time solvability	160
8.3	Polynomial-time solvability	163
8.3.1	Vertex-disjoint packing	163
8.3.2	Edge-disjoint packing	164
8.4	APX-hardness	166
8.4.1	Vertex-disjoint packing	166
8.4.2	Edge-disjoint packing	168
8.5	Summary and future work	176
9	Conclusion	178
	Glossary of abbreviations	181
	Bibliography	192

List of Figures

2.1	Part of the known hierarchy of solution concepts in hedonic games	9
4.1	A representation of the instance (N', P') of 3DR-B that contains no stable matching	52
6.1	Agents and triples in M before a new iteration of the while loop	67
6.2	The structure of M' in Construction 3	71
6.3	The structure of M' in Construction 6	71
6.4	The structure of M' in Construction 1	72
6.5	The structure of M' in Construction 2	73
6.6	The structure of M' in Construction 4	73
6.7	The structure of M' in Construction 5	73
6.8	The structure of M' in Construction 7	74
6.9	A hypothetical blocking triple in M' in Construction 7	76
6.10	An instance of 3DR-AS-SMUW in which $u(M^*) = 2u(M)$	89
6.11	The reduction from PIT to the problem of deciding if an instance of 3DR-AS with binary preferences contains a stable matching	90
6.12	The reduction from PIT to the problem of deciding if an instance of 3DR-AS with ternary preferences contains a stable matching	94
7.1	The reduction from $\text{X3SAT}_+^{\leq 3}$ to the problem of deciding if a given instance of 3DR-AS contains an envy-free matching	105
7.2	The reduction from $\text{X3SAT}_+^{\leq 3}$ to the problem of deciding if a given instance of 3DR-AS contains an wj -envy-free matching	123
7.3	The reduction from DTP to the problem of deciding if a given instance of 3DR-AS with ternary preferences contains a j -envy-free matching	131

7.4	The reduction from PIT to the problem of deciding if an instance of 3DR-AS with symmetric preferences contains a j-envy-free matching	137
8.1	The reduction from $\text{Max } 2\text{SAT}^{\leq 3}$ to EDK_4	169
8.2	The reduction from $\text{Max } 2\text{SAT}^{\leq 3}$ to EDK_5	174

List of Pseudocode

3.1	Algorithm cyclicSerialDictatorship	27
4.1	Algorithm serialDictatorship	46
6.1	Subroutine repair (Phase 1)	69
6.2	Subroutine repair (Phase 2)	70
6.3	Subroutine findStableInTriangleFree	81
6.4	Algorithm findStableUW	85
7.1	Subroutine nonC4Components	112
7.2	Subroutine oneC4TwoSingles	113
7.3	Subroutine multipleOfThreeC4s	114
7.4	Subroutine configureSurplusAgents	115
7.5	Algorithm wjPathsCycles	117
8.1	Algorithm greedyCliques	162

Acknowledgements

In the course of my studies I have been supported by more people than can be listed.

David Manlove has been an outstanding supervisor. I have benefited enormously from his superlative knowledge of algorithmics and his meticulous (latterly virtual) blue pen. Most importantly however, he has been a constant source of enthusiasm and encouragement.

Many thanks to my two examiners, Martin Gairing and Ciaran McCreesh, for taking the time to read my thesis in full and for their helpful and constructive feedback.

I am very grateful to the EPSRC for funding my degree, and to the Graduate School and the School of Computing Science for providing other funding, training, and administrative support.

I am thankful to my collaborator Ágnes Cseh for working with me. Our meetings and correspondence have been both enjoyable and hugely valuable. Thanks also to my office-mate Peace Ayegba, who kindly read through my thesis draft and made suggestions to improve its style.

Thanks to each and every one of my other office-mates, fellow PhD students, colleagues, and other friends in the School of Computing. I hope your time at Glasgow will be (or has been!) as fun as mine.

Thank you also to my family and friends outside work, for all their encouragement and moral support.

Finally, a special thank you to my partner Ailsa for her unwavering support — *QAPLA'!*

Declaration

I certify that the thesis presented here for examination for a Ph.D. degree of the University of Glasgow is my own work, with the following exceptions:

- Chapter 3 includes joint work with Klaus Heeger. In particular, it was Klaus' idea that led to a significant improvement of the approximation ratio (in Theorem 3.2) and a collaborative brainstorm that produced the approximation algorithm in the setting of master lists (shown in Theorem 3.3).
- The instance described in Theorem 4.5, presented in Chapter 4, is the product of a joint effort with Sofia Simola.
- Chapter 7 includes joint work with Ágnes Cseh, conducted part-time from December 2020 to May 2022. Ágnes provided expert guidance and advice throughout the project. In terms of our new results, it was her ideas that led to the polynomial-time algorithm and NP-completeness reduction related to envy (now shown in Theorem 7.2). We worked jointly on the algorithm and reduction for weakly justified envy, which I then formalised and wrote up (in Theorem 7.4). I developed the NP-completeness reduction for weakly justified envy (shown in Theorem 7.3) and proved the results relating to justified envy (shown in Theorems 7.5–7.7). Everything else was a combined effort.

My supervisor, David Manlove, has supported this research throughout, providing general guidance and critically reviewing technical material.

Research output

The following papers are a result of the research presented in this thesis:

- M. McKay and D. Manlove. The Three-Dimensional Stable Roommates Problem with Additively Separable Preferences. In *Proceedings of SAGT '21: The 14th International Symposium on Algorithmic Game Theory*. LNCS, vol. 12885, pp. 266–280. Springer (2021). This paper contains some of the results presented in Chapter 6.
- Á. Cseh, M. McKay, and D. Manlove. Envy-freeness in 3D Hedonic Games. Submitted to *Autonomous Agents and Multi-Agent Systems*, 2023. A preprint is available as Technical Report number 2209.07440 (2022) on arXiv, from <https://arxiv.org/abs/2209.07440>. This paper contains the results presented in Chapter 7.
- M. McKay and D. Manlove. Packing K_r s in bounded degree graphs. Submitted to *Discrete Applied Mathematics*, 2022. A preprint is available as Technical Report number 2209.03684 (2022) on arXiv, from <https://arxiv.org/abs/2209.03684>. This paper contains the results presented in Chapter 8.

Chapter 1

Introduction

1.1 About

In this thesis we study the algorithmic aspects of models in which agents form coalitions.

Sometimes, these models can be applied in practice. For example, consider a set of cooperative robots that may form work teams in order to maximise their total efficiency (in terms of some specific criteria). In such an application, research in this area can provide practical tools and techniques to organise the robots in an optimal way [1]. Other models are less directly applicable, but by studying them we can make insights into how autonomous agents, such as humans, behave in the real world [2, 3]. Even purely abstract research involving agents that form coalitions has led to interesting and sometimes unexpected theoretical results.

The models that we focus on involve a set of agents which is to be partitioned into disjoint *coalitions* of a fixed size. We call such a partition a *matching*. For example, this might represent a set of robot workers who will organise themselves into work teams, where each work team contains three robots. We assume that each agent has *preferences* between the possible coalitions that they might belong to. For example, it may be that for practical reasons each robot can only evaluate its own individual efficiency in a specific work team. Each robot therefore assigns a numerical score to each of the work teams it may belong to, where a higher score indicates that a given robot is more effective in that work team. Alternatively, such a model might represent a set of students in a class who must be assigned to pairs by their teacher. Each student might list their classmates in order from the most preferred to least preferred.

In these models, a central idea is the existence of a matching that satisfies some specific criteria, or is somehow optimal. For example, for a specific set of robots, we might seek a matching in which the sum of the individual robots' efficiency scores is maximised. Alternatively, for a specific set of students we might ask whether it is possible to pair the students

such that no two students both prefer each other to their respectively assigned partners. We refer to these criteria as *solution concepts*. The solution concept in the latter example, which involves a set of agents that may deviate away from their assigned coalitions, is a type of *stability*. Stability is a central concept in the theory of coalition formation in general and this thesis in particular.

The specific contribution of this thesis relates to the algorithmic aspects of these models. For example, one of our new results is an algorithm that could be used to construct a matching of robots into work teams that satisfies a type of stability. For a slightly different solution concept we show that, for a given set of robots, a matching that satisfies that solution concept does not necessarily exist. Furthermore, deciding whether a given set of robots can be matched in such a way that satisfies that solution concept is NP-complete. In contrast, we show that if the robots' scores are all restricted in a certain way then such a matching must always exist, and can be found by an efficient algorithm.

1.2 Thesis statement

Computational problems involving agents that form coalitions are generally NP-hard, and in particular when those coalitions must have a fixed size. Nevertheless, there exist natural models of fixed-size coalition formation in which optimal or near-optimal matchings can be found using efficient algorithms.

1.3 Contribution

Our main contribution relates to a family of models of fixed-size coalition formation known as *Three-Dimensional Roommates* (3DR). The defining characteristic of a model of 3DR is that there are a set of $3n$ agents that must be partitioned into n triples, which we call a *matching*, and each agent has some kind of preference between the triples to which they may belong. Working within the framework of 3DR, we systematically study a set of fixed-size coalition formation problems.

Many real-world scenarios involve coalitions of restricted size, which provides a strong rationale for our study [4]. We motivate our study of 3DR in particular with the fact that results relating to coalitions of size three often generalise directly to problems involving other restrictions on coalition size. Even when this is not the case, it may be that they at least give an indication of what to expect in more general models. For this reason, much of the existing research also focused specifically on models in which coalitions must have size three.

We begin by reviewing the literature in and around coalition formation, including both models involving restricted coalition sizes and models in which coalitions need not have a fixed size. We tie related concepts and theory together from across computing science, economics, and game theory.

In our first technical contribution, we consider a variant of 3DR known as *Three-Dimensional Stable Matching with Cyclic preferences* (3-DSM-CYC), which has been of independent interest. In 3-DSM-CYC, each agent has one of three types, sometimes termed *man*, *woman*, and *dog*, which have a cyclic order. Any feasible triple must contain exactly one agent of each type. There are n agents of each type and each agent has a preference list only over the agents of the next type. A matching is *stable* if there exists no triple t in which each agent in t prefers the triple t to their assigned triple in the matching. It was recently shown, contrary to previous conjectures, that a given instance of 3-DSM-CYC need not contain a stable matching and that the associated decision problem is NP-complete [5]. We thus consider the approximability of a closely related optimisation problem, in which the objective is to construct a matching that is, in terms of a specific measure, as stable as possible. To our knowledge, this work is the first investigation into the approximability of 3-DSM-CYC. We first show that an existing algorithm for another three-dimensional matching problem, which is closely related to 3-DSM-CYC, can be used to construct a $9/4$ -approximation algorithm. Improving this approximation, we then present a $6/5$ -approximation algorithm based on serial dictatorship. We then consider a restriction of 3-DSM-CYC in which the preferences of all agents of at least one type are in some way similar, using a specific similarity measure. Specifically, we suppose the preference lists of all agents of at least one type are derived from a master preference list and consider the maximum *Kendall tau distance* [6] between the master list and the list of any agent of that type. We show that if this distance is sufficiently small then as it is further reduced the approximation ratio decreases from $6/5$ to 1.

Next, we define two new models of 3DR that involve so-called \mathcal{B} - and \mathcal{W} -preferences, which we name 3DR-B and 3DR-W. Using \mathcal{B} - (\mathcal{W} -) preferences, each agent has a strict preference list over the other agents and compares two triples based only on the most-preferred (least-preferred) member of each triple. We consider in both 3DR-B and 3DR-W the existence of matchings that are stable. We first show that it is NP-complete to decide if a given instance (of agents and preferences) of either model contains a stable matching. Interestingly, this contrasts with the existence of polynomial-time algorithms in two analogous models in which coalitions may have any size [7, 8]. For both 3DR-B and 3DR-W, we also consider a closely related optimisation problem in which the objective is to construct a matching that is, in terms of a specific measure, as stable as possible. We show that an existing result leads to a $9/4$ -approximation algorithm in both models and a simple algorithm based on serial dictatorship gives a $3/2$ -approximation for 3DR-B.

We then formalise a model of 3DR with additively separable preferences, which we call

3DR-AS. In this model, each agent provides a numerical valuation of every other agent and compares triples based on the sum of the valuations of the other two agents in each triple. We investigate in 3DR-AS the existence of stable matchings as well as matchings that are envy-free, meaning there exists no pair of agents where the one would prefer to swap places with the other. In fact, we consider three successively weaker formalisms of this notion, namely envy-freeness, weakly justified envy-freeness, and justified envy-freeness. We consider the computational problems of deciding if such a matching exists, and constructing one if so. In particular, we study these problems in a setting where the agents' valuations are restricted. We consider various restrictions involving binary, ternary, and symmetric valuations. We provide a full complexity classification and identify dichotomies in terms of these restrictions. Interestingly, we identify a general trend that shows, for successively weaker solution concepts, existence and polynomial-time solvability hold under three successively weaker restrictions on the agents' preferences.

Building on our new result that any instance of 3DR-AS with binary and symmetric preferences must contain a stable matching, we also consider a related optimisation problem in which the objective is to construct a stable matching in such an instance with maximum utilitarian welfare, i.e. the total sum of agents' valuations of their assigned partners is maximised. We devise a 2-approximation algorithm for this optimisation problem.

Finally, we consider a problem in graph theory that generalises the notion of assigning agents to coalitions of a fixed size. Rather than partitioning a set of agents, the problem is to find a maximum-cardinality set of r -cliques in an undirected graph, subject to that set being either vertex disjoint or edge disjoint, for a fixed integer $r \geq 3$. This general problem is known as the K_r -packing problem. Here we study the restriction of this problem in which the graph has a fixed maximum degree Δ . It is known for $r = 3$ that the vertex-disjoint (edge-disjoint) variant is solvable in linear time if $\Delta = 3$ ($\Delta = 4$) but APX-hard if $\Delta \geq 4$ ($\Delta \geq 5$) [9]. In other words, there exists some fixed constants $\varepsilon > 1$ and $\varepsilon' > 1$ such that no polynomial-time ε -approximation algorithm exists for the vertex-disjoint variant if $\Delta \geq 4$; and no polynomial-time ε' -approximation algorithm exists for the edge-disjoint variant if $\Delta \geq 5$. We generalise these results to an arbitrary but fixed $r \geq 3$, and provide a full complexity classification for both the vertex- and edge-disjoint variants in graphs of maximum degree Δ , for every $r \geq 3$. Specifically, we show that the vertex-disjoint problem is solvable in linear time if $\Delta < 3r/2 - 1$, solvable in polynomial time if $\Delta < 5r/3 - 1$, and APX-hard if $\Delta \geq \lceil 5r/3 \rceil - 1$. We also show that if $r \geq 6$ then these implications also hold for the edge-disjoint problem. If $r < 6$, then the edge-disjoint problem is solvable in linear time if $\Delta < 3r/2 - 1$, solvable in polynomial time if $\Delta \leq 2r - 2$, and APX-hard if $\Delta > 2r - 2$.

1.4 Structure

The remainder of this thesis is structured as follows:

- In Chapter 2, we give an overview of literature involving coalition formation, hedonic games, and fixed-size coalitions, setting the scene for our subsequent results.
- In Chapter 3, we introduce 3-DSM-CYC and present new results on the approximability of a related optimisation problem, both in the general case and in a restriction of the problem involving master lists.
- In Chapter 4, we define 3DR-B and present new results concerning the existence of stable matchings in instances of 3DR-B and on the approximability of a related optimisation problem.
- In Chapter 5, we define 3DR-W and present new results concerning the existence of stable matchings in instances of 3DR-W and on the approximability of a related optimisation problem.
- In Chapter 6, we define 3DR-AS and present new results concerning the existence of stable matchings in 3DR-AS, under various restrictions on the agents' valuations.
- In Chapter 7, we present new results concerning the existence of envy-free, weakly justified envy-free, and justified envy-free matchings in instances of 3DR-AS, under various restrictions on the agents' valuations.
- In Chapter 8, we present new results on the complexity of the K_r -packing problem in graphs of fixed maximum degree.
- In Chapter 9, we summarise the contribution of this thesis and discuss some directions for future work.

Chapter 2

Literature review

2.1 Introduction

Models involving agents that form coalitions appear in a variety of settings, and terminology varies depending on the context and application. In this chapter we give an overview of the existing models in which a set of agents is to be partitioned into disjoint coalitions based somehow on the preferences of those agents, with a focus on models related to fixed-size coalitions and in particular Three-Dimensional Roommates (3DR). We shall present models that appear across the literature of computing science, economics, and game theory. For many concepts, the notation and terminology used varies between fields, even when the underlying concept is the same.

In Section 2.2, we cover models of coalition formation in which coalitions need not have a fixed size. Models of this type are almost always defined as *hedonic games*, which are the subject of a sizeable area of research. Many of the concepts and terminology associated with hedonic games are also used in other models of coalition formation, some of which we consider subsequently. For this reason we consider these models first.

In Section 2.3, we cover models of coalition formation in which a restriction exists on the sizes of possible coalitions. In most of the existing models of this type, for example in the *Stable Roommates problem* (SR), the restriction is that any feasible coalition has some fixed size. The terminology used in the literature to refer to these models can vary. For the sake of consistency, we standardise terminology. For example, we use the term “matching”, meaning a set of coalitions, while other authors use the terms *assignment* [10] or *partition* [7] for a similar purpose.

2.2 Hedonic games and stable partitions

A well-studied model of coalition formation is the *hedonic game*. The defining characteristic of a hedonic game is *hedonic preferences*. Generally speaking, hedonic preferences mean that “every agent only cares about which agents are in its coalition, but does not care how agents in other coalitions are grouped together” [11, 12]. This contrasts with other so-called co-operative games that involve divisible goods, which are shared among the agents based on the coalitions formed. *The Handbook of Computational Social Choice* [11] defines a hedonic game as follows:

Definition 2.1 ([11]). Hedonic game

Let N be a finite set of agents. A *coalition* is a non-empty subset of N . Let $\mathcal{N}_i = \{S \subseteq N : i \in S\}$ be the set of all coalitions (subsets of N) that include agent $i \in N$. A *coalition partition* π is a partition of the agent set N into disjoint coalitions. A *hedonic game* or *hedonic coalition formation game* is a pair (N, \succsim) , where \succsim is a *preference profile* that specifies for every agent $i \in N$ a reflexive, complete, and transitive relation \succsim_i on \mathcal{N}_i . We call \succsim_i a *preference relation*.

An *instance* of a hedonic game is a specific set of agents and a preference profile \succsim . The exact representation of \succsim can vary. For example, one possibility is that for each agent i , \succsim_i is an ordered list containing every possible coalition in \mathcal{N}_i . For some partition π and some agent i , we denote by $\pi(i)$ the coalition containing i . For any two agents i and j , if $j \in \pi(i)$ then we say that i is a *partner* of j . For some agent i and any two coalitions S and T in \mathcal{N}_i , we write $S \sim_i T$ if $S \succsim_i T$ and $T \succsim_i S$ and write $S \succ_i T$ if $S \succsim_i T$ and $T \not\succsim_i S$. If $S \succsim_i T$ then we say that i *weakly prefers* S to T . If $S \sim_i T$ then we say that i is *indifferent* between S to T . If $S \succ_i T$ then we say that i *strictly prefers* S to T . For some agent i and any coalition S , if $S \succsim_i T$ for any other coalition T then we say that S is one of i 's *most-preferred* coalitions. Each agent's preference between two coalition partitions depends only on their assigned coalition in each, so for any agent i and coalition partitions π and π' , we write $\pi \succsim_i \pi'$ if $\pi(i) \succsim_i \pi'(i)$, $\pi \succ_i \pi'$ if $\pi(i) \succ_i \pi'(i)$, and $\pi \sim_i \pi'$ if $\pi(i) \sim_i \pi'(i)$.

In a given hedonic game, we may ask if there exists a coalition partition in which each agent is satisfied. In formal terms, we may consider the existence of a coalition partition that meets some fixed *solution concept*. One well-studied solution concept is *individual rationality*. We say that coalition partition π is *individually rational* if there exists no agent i who strictly prefers the *individual coalition* $\{i\}$ to $\pi(i)$. Some other relevant solution concepts are:

- *Perfection*. A coalition partition is *perfect* if each agent belongs to one of its most-preferred coalitions [13]. This is a very strong solution concept and in general, a given

hedonic game need not contain a perfect partition [11].

- *Pareto optimality*. Given coalition partitions π and π' , we say π' *Pareto dominates* π if $\pi' \succeq_i \pi$ for each agent $i \in N$ and there exists some agent $j \in N$ where $\pi' \succ_j \pi$. A coalition partition π is *Pareto optimal* if no coalition partition exists that Pareto dominates π [14].
- *Core stability* (also known as *stability*). We say coalition partition π is *core stable* if no set of agents $S \subseteq N$ exists such that $S \succ_i \pi(i)$ for each agent $i \in S$. In other words, no set of agents have a common incentive to deviate from the coalition partition and form a new coalition. If such a set S exists then we call it a *blocking coalition*.
- *Envy-freeness*. Suppose we are given some coalition partition π . If there exists some pair of agents $i, j \in N$ where $\pi(i) \neq \pi(j)$ and i strictly prefers $(\pi(j) \setminus \{j\}) \cup \{i\}$ to $\pi(i)$ then we say that i *has envy for* j . We say that π is *envy-free* if no such pair exists [15].
- *Justified envy-freeness*. Suppose we are given some coalition partition π . If there exists some pair of agents $i, j \in N$ where i envies j and $(\pi(j) \setminus \{j\}) \cup i \succ_k \pi(j)$ for each agent $k \in \pi(j) \setminus \{j\}$, then we say that i *has justified envy for* j . In other words, i envies j and each partner of j in π would be strictly better off if i were to replace j in $\pi(j)$. We say that π is *justified envy-free* if no such pair exists [16].
- *Weakly justified envy-freeness*. Suppose we are given some coalition partition π . If there exists some pair of agents $i, j \in N$ where i envies j and $(\pi(j) \setminus \{j\}) \cup i \succeq_k \pi(j)$ for each agent $k \in \pi(j) \setminus \{j\}$, then we say that i *has weakly justified envy for* j . In other words, i envies j and each partner of j in π would be weakly better off if i were to replace j in $\pi(j)$. We say that π is *weakly justified envy-free* if no such pair exists [16].

Some solution concepts generalise others. For example, by definition any coalition partition that is core stable must also be individually rational, and any perfect partition must also satisfy all of the other concepts that we have described. In fact, a remarkable hierarchy of solution concepts exists [11, 15, 16]. Part of this hierarchy, involving the seven solution concepts described so far, is illustrated in Figure 2.1.

A wide range of solution concepts have been proposed and studied in the setting of hedonic games. Many (such as Pareto optimality) have roots in game theory and economics [11]. In a well-cited 2013 article, Aziz et al. [15] considered a variety of solution concepts and applied them to a specific type of hedonic game. One such concept is *popularity*. They defined a popular partition π as a partition in which for every alternative partition π' , the number of agents that prefer π' to π is at least the number of agents that prefer π to π' . We shall discuss later (in Sections 2.3.2 and 2.3.4) some other research that involves popular partitions.

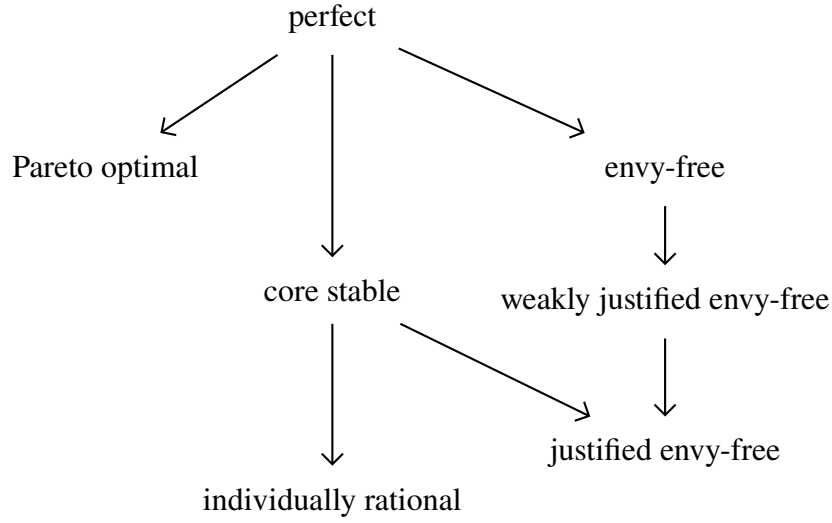


Figure 2.1: Part of the known hierarchy of solution concepts in hedonic games. In the diagram, an arrow points from one concept to another if any partition that satisfies the former must also satisfy the latter. Adapted from the *Handbook of Computational Social Choice* [11].

In some hedonic games, it is assumed that the agents' preferences have some additional structure, which dictates the representation of the preference profile \succsim . For example, it may be that a hedonic game models agents in the real world whose preference between coalitions is based on some underlying preference over the agents themselves. The exact representation of \succsim is also important when considering computational aspects of hedonic games.

The simplest system of preference representation is to suppose that each agent has a *preference list* over the $2^{|N|-1}$ possible coalitions in \mathcal{N}_i , in order from the most-preferred to the least-preferred. This system is known as *lists of coalitions* (LCs) [17]. In the literature, unless otherwise specified, it is usually assumed that agents provide LCs. Alternatively, agents might list only the coalitions from the most-preferred to the individual coalition, which is known as *Individually Rational Lists of Coalitions* (IRLCs) [17]. IRLCs might be used if the solution concept implies individual rationality, since no agent i need ever consider a coalition partition less-preferred than the individual coalition $\{i\}$.

LCs and IRLCs involve *ordinal* preferences, meaning that they describe a preference between alternatives but do not quantify the extent that one alternative may be preferred over another [18]. If preferences do involve such a quantification then they are *cardinal*. For example, we could assume that each agent i assigns a numeric valuation to every possible coalition in \mathcal{N}_i .

For some applications LCs or IRLCs may not be practical since they require agents to list up to $2^{|N|-1}$ possible coalitions. For example, consider a hedonic game model used by 50 cooperative robots [1]. Using LCs might involve each robot listing $\sim 5.6 \times 10^{14}$ coalitions, which is likely to be challenging in practice. In such an application, it might be better

to assume that the agents' preferences have a specific structure. For example, we could assume that each agent has a preference list over the other (individual) agents. We say that this representation is more succinct since each preference list has length $|N| - 1$. This representation might also reflect a natural structure in the preferences of real people [19].

To formalise such a hedonic game, in which agents have preference lists over the other agents, we must be able to infer from each agent's preference list a preference over coalitions. One way to do this systematically is to use a *set extension rule*. Set extension rules have been well studied outside the context of coalition formation, both in general [20] and in other areas of computational social choice, such as multi-winner voting [14]. A number of set extension rules have been considered in the hedonic games literature, which we shall introduce shortly.

We first introduce some new notation. For an agent i let P_i be a preference list over other (individual) agents. Thus, P_i describes a strict weak order \succsim_i over $N \setminus \{i\}$. For any two agents j and k we say that i *prefers* j to k , denoted by $j \succ_i k$, if j precedes k in P_i . It may be that the preference list P_i contains *ties*. A tie is a set of agents $T \subseteq N \setminus \{i\}$ where $j \sim_i k$ for any pair of agents $j, k \in T$ and, for any $l \in T$ and $m \in N \setminus (T \cup \{i\})$, either $l \succ_i m$ or $m \succ_i l$. If a preference list contains no ties then we say it is *strict*. Strict preference lists therefore represent a *total order*. Given a set of agents S and an agent i where $i \notin S$, let $\mathcal{W}_i(S)$ be the least-preferred agent in S according to P_i (or an arbitrary least-preferred agent if P_i contains ties). We define $\mathcal{B}_i(S)$ analogously, as the most-preferred agent in S according to P_i (or an arbitrary most-preferred agent if P_i contains ties).

We now introduce two well-known set extension rules, in which an agent compares two coalitions based on either the most-preferred or least-preferred agent in each coalition. Using \mathcal{W} -preferences [8], each agent i prefers some coalition S to another coalition T if $\mathcal{W}_i(S) \succ_i \mathcal{W}_i(T)$, and is otherwise indifferent. A related set extension rule is \mathcal{B} -preferences [7], in which i prefers S to T if $\mathcal{B}_i(S) \succ_i \mathcal{B}_i(T)$. Usually an additional rule is added, so that i prefers S to T if either $\mathcal{B}_i(S) \succ_i \mathcal{B}_i(T)$, or $\mathcal{B}_i(S) \sim_i \mathcal{B}_i(T)$ and $|S| < |T|$. Hajduková [21] noted the motivation for this additional rule: if there is no condition placed on the cardinalities of S and T then, given an arbitrary instance of a hedonic game using \mathcal{B} -preferences, the *grand coalition partition* $\{N\}$ is a perfect partition.

Cechlárová and Hajduková [19] reason that a strong motivation for studying \mathcal{W} - and \mathcal{B} -preferences in hedonic games comes from more general work of Kannai and Peleg [22] in social choice. Motivated by voting theory, Kannai and Peleg demonstrate that any set extension rule that satisfies two natural axioms makes any agent i indifferent between any set S and $\{\mathcal{B}_i(S), \mathcal{W}_i(S)\}$. Barberà, Bossert, and Pattanaik discuss this result and some related work in *Ranking Sets of Objects*, Chapter 17 of the *Handbook of Utility Theory* [20]. It remains open whether Kannai and Peleg's axiomatic result can be further applied to the setting of hedonic games.

Another popular system is *additively separable preferences*, which are the defining characteristic of an *Additively Separable Hedonic Game* (ASHG) [23, 24]. Additively separable preferences are a form of cardinal preferences and can be defined in terms of *valuation* functions. Each agent i has a valuation function $v_i(j) : N \setminus \{i\} \mapsto \mathbb{R}$ (where \mathbb{R} is the set of real numbers). Given an agent i and a set of agents $S \subseteq N$, we say that agent i has *utility* $u_i(S) = \sum_{j \in S \setminus \{i\}} v_i(j)$ in S . For two coalitions S and T , we define $S \succ_i T$ if $u_i(S) > u_i(T)$ and $S \sim_i T$ if $u_i(S) = u_i(T)$. Let $u_i(\pi)$ be short for $u_i(\pi(i))$. We also define the *utilitarian welfare* of a partition π as $u(\pi) = \sum_{i \in N} u_i(\pi)$. Various other measures related to “welfare” have been defined in ASHG [15]. Additively separable preferences generalise *separable preferences* [21], which can be defined as a restriction of additively separable preferences in which $v_i(j) \in \{-1, 0, 1\}$ for any two agents i and j . We remark that a similar model to an ASHG has been studied as a *Weighted Graph Game* [25]. It is also possible to define other restrictions of additively separable preferences. For example, we say that valuations are *symmetric* if $v_i(j) = v_j(i)$ for any two agents i and j . We say that valuations are *binary* (also termed *simple* [26]) if $v_i(j) \in \{0, 1\}$. Similarly, we say that valuations are *ternary* if $v_i(j) \in \{0, 1, 2\}$.

In a seminal 2002 article, Bogomolnaia and Jackson [27] focused on solution concepts in AHSGs that involve the movement of individual agents away from their assigned coalitions (such as individual rationality). They observed that, for some of these solution concepts, if preferences are symmetric then a satisfactory partition must exist. The proof of this observation follows from the fact that any agent’s movement (or *deviation*) produces a partition with strictly higher utilitarian welfare. This type of proof has since been termed a “potential function” argument [11]. Since 2002, similar arguments have since been used to show new results in some other types of hedonic games [28].

In 2007, Huang [29] proposed a restricted variant of additively separable preferences called *Precedence by Ordinal Number* (PON), which is related to Borda scores [11]. In PON, each agent has an ordinal preference list, and $v_i(j)$ is defined to be the rank, beginning from one, of j in the preference list of i .

Another variant of additively separable preferences is *fractional preferences*, in which the utility of a coalition is based on the average valuation over all agents in that coalition. Aziz et al. [26] presented an extensive survey of so-called *Fractional Hedonic Games* (FHGs) in 2019. We remark that if we require any feasible coalition to have a fixed size k , then the definitions of an ASHG and an FHG are effectively equivalent.

From an algorithmic perspective, there are three main computational problems associated with a given model of a hedonic game. The first is a decision problem known as the *existence problem*. For some fixed solution concept, it asks if a given instance of a hedonic game contains a coalition partition meeting that concept. The second is the *construction problem*,

which is the search problem of either finding a coalition partition that meets a given solution concept or reporting that no such partition exists. The third is the *verification problem*. For a fixed solution concept and fixed instance of a hedonic game, it asks if a given partition in that instance meets that concept. These three problems are closely related. For example, if the existence problem is NP-complete, for some model of a hedonic game, then it follows immediately that the verification problem is solvable in polynomial time and the construction problem is NP-hard.

The system of preference representation is particularly meaningful when considering these computational problems. Consider the core-stability existence problem for a model of a hedonic game in which preferences are represented using IRLCs. Encoding the agents' preferences requires $O(2^{|N|})$ space, but since they form part of the problem input it is possible to scan every preference list in linear time with respect to the size of the input. This observation means that the corresponding verification problem belongs to the complexity class P. Ballester [30] showed that the corresponding existence problem is NP-hard and thus NP-complete. In some cases, more succinct systems of preference representation lead to coNP-complete verification problems [31] and Σ_2^P -complete (NP^{NP}-complete) existence problems [32, 33].

In research on the algorithmics of hedonic games, the goal is often to show that, for a given preference representation and solution concept, the verification, existence, or construction problems are either solvable in polynomial time or are computationally hard. For example, in 2001 Cechlárová and Romero-Medina [7] considered the core-stability existence problem in hedonic game models (referred to as the *Stable Partition problem*) using \mathcal{B} - and \mathcal{W} -preferences. They showed that for \mathcal{B} -preferences, a core stable partition must exist, and can be found in polynomial time. Later, in 2004, Cechlárová and Hajduková considered the analogous model using \mathcal{W} -preferences. They showed that in that model, a core stable partition may not exist, but that a polynomial-time algorithm exists that can either find a core stable partition or report that none exist. Interestingly, even though the definitions of \mathcal{B} - and \mathcal{W} -preferences are similar, the algorithms used in both settings are significantly different.

In 2004, Ballester [30] showed that the existence problem is NP-complete for a number of hedonic games involving systems of preferences in which agents have preference lists over all possible coalitions. In particular, he showed that in a hedonic game using LCs, the core-stability existence problem is NP-complete.

In 2010, Sung and Dimitrov [34] studied the existence of core stable partitions in ASHG. They showed that an ASHG may not contain a core stable partition and that the existence problem is strongly NP-hard, but left open the question of whether it belongs to NP. They also presented hardness results relating to other solution concepts in ASHG. In their 2011 paper, Aziz et al. [15] strengthened Sung and Dimitrov's result, showing that the

core stability existence problem is strongly NP-hard even when valuations are symmetric. In 2013, Woeginger [32] resolved this open question and showed the existence problem is Σ_2^P -complete.

In 2019, Gairing and Savani [35] considered certain solution concepts in ASHG with symmetric preferences, where for each solution concept a potential function argument can be used to show that a satisfactory partition always exists. They observed that all of the associated construction problems can be modelled as local search problems in the class PLS. Notably, they showed that many are also PLS-complete. These results are interesting because it is unlikely that any problem in PLS is NP-hard (which would imply $NP = coNP$), but it is also believed that no PLS-complete problem can be solved in polynomial time [36].

In their 2019 article on FHGs, Aziz et al. [26] showed that a core stable partition may not exist in a given FHG, even when valuations are binary and symmetric, and that the corresponding existence problem is Σ_2^P -complete. They also presented some positive results for FHGs with binary and symmetric preferences.

In their 2013 paper, Aziz et al. [15] considered the existence of envy-free partitions in ASHG. They noted that the *singleton coalition partition* $\{\{\alpha_1\}, \{\alpha_2\}, \dots, \{\alpha_{|N|}\}\}$ is trivially envy-free and thus considered the existence of partitions that simultaneously satisfy envy-freeness as well as other solution concepts. In 2018, Ueda [37] considered envy-freeness and justified envy-freeness in a hedonic game model using LCs. He observed in this model that both the singleton partition and the grand partition $\{N\}$ are also envy-free, although there exist instances in which no “non-trivial” coalition partition is envy-free, and additional instances in which no non-trivial partition is justified envy-free. He also observed that core stability implies justified envy-freeness. In 2019, Barrot and Yokoo [16] noted Ueda’s observation and continued exploring the existence of coalition partitions that satisfy a combination of solution concepts, including envy-freeness, weakly justified envy-freeness, and justified envy-freeness. As well as some non-existence results for ASHG, they also considered other models involving more general systems of preference representation. Notably, they presented results relating to the existence of such coalition partitions in a setting in which preferences either satisfy *top responsiveness* or *bottom responsiveness*, two restrictions already well-established in the hedonic games literature.

2.3 Coalitions of restricted size

2.3.1 Two-dimensional matching and roommates

Historically, most of the research involving fixed-size coalitions relates to models in which agents are to be paired into coalitions of size two. We call such models *two-dimensional*. The

study of such models is closely related to the area of *matching under preferences*. In his 2013 book, Manlove [18] presents a broad survey of the literature of matching under preferences. This area is also related to the concept of matching in graph theory [38].

A seminal model of two-dimensional matching under preferences was introduced in 1962 by Gale and Shapley [39]. A set of applicants are applying individually to a set of colleges. Each applicant produces a strict preference list of colleges from most-preferred to least-preferred. Similarly, each college produces a strict preference list of students. Each college may offer multiple places but each student must apply to exactly one college. The authors considered how best to match applicants to college places. They asked if there exists an assignment of students to college places such that no two applicants α and β , assigned to colleges A and B , constitute a *blocking pair*, meaning that β prefers A to B , and A prefers β to α . They called such an assignment a *stable matching*, and provided an efficient algorithm that can construct a stable matching. Interestingly, they also showed that this algorithm is, in a sense, optimal for the applicants: “every applicant is at least as well off under the assignment given . . . as he would be under any other stable assignment”. Although this problem is stated in terms of colleges and applicants, it is commonly referred to as the *Hospitals-Residents problem* (HR) [18].

In the same paper, Gale and Shapley also considered the restriction of HR when hospitals admit exactly one student. They proposed a heterosexual marriage metaphor involving a set of n men and n women, who are to be matched into n pairs. For this reason this problem is known as the *Stable Marriage problem* (SM) [18]. They also described a third problem, known as the *Stable Roommates problem* (SR), which can be defined as a generalisation of SM. In SR, there exists a single set of agents, who have strict preference lists over all other agents. The goal, as for SM, is to construct a matching in which no two agents prefer each other to their respective assigned partners [40]. SR can thus be equivalently viewed as a hedonic game in which agents provide LCs (see Section 2.2) and any feasible coalition has size two. Gale and Shapley showed that, in contrast to HR (and SM), there exist instances of SR that contain no stable matching. In 1976, Knuth [41, Problem 12] asked if a stable matching can be found in polynomial time in a given instance of SR. This question was finally resolved by Irving [40] in 1985 who presented a polynomial-time algorithm that can decide if a given instance of SR contains a stable matching, and constructs one if so.

Since 1962, a multitude of research in economics and computing science has been influenced by concepts and theory from Gale and Shapley. The 2012 Nobel prize in Economic Sciences was awarded to Shapley and Roth [42] for their work on the theory and application of matching under preferences. The citation describes Shapley’s theoretical research, including the Gale-Shapley algorithm, and the practical work of Roth, who successfully applied this theory to the assignment of doctors to hospital positions in the USA.

Much research in the area of matching under preferences involves variants and generalisations of HR, SM and SR [18]. For problems that involve preference lists, one natural generalisation is to allow *incomplete lists*. In general terms this means that for each agent certain alternatives in their preference list are unacceptable, meaning that in no feasible matching is any agent assigned an unacceptable alternative. The variant of SM with incomplete lists is known as SMI [18]. In SMI, incomplete preference lists characterise unacceptable pairs of men and women. Thus, it may be impossible to produce a stable matching involving all men and all women, so we generalise the definition of a stable marriage and allow agents to be *unmatched*, meaning they have no partner in a given matching. A blocking pair now consists of a man m and a woman w that find each other acceptable, where: (1) either m is unmatched or m prefers w to his partner, and (2) either w is unmatched or w prefers m to her partner. The definition of SRI is analogous. The introduction of incomplete preference lists in SMI and SRI, and in other problems of matching under preferences, has generated significant interest [18].

2.3.2 Multidimensional matching of a multipartite agent set

In this section we focus on problem models that involve matching a multipartite set of agents. An early model of this type was proposed in 1976 by Knuth [41, Problem 11], who asked if SM can be extended to three sets, for example men, women and dogs. It is not immediately clear how this should be done, and various different models have since been proposed. In all of them, a feasible coalition must contain exactly one man, one woman, and one dog, and thus have size three. We classify such models as *three-dimensional*. A close connection also exists to the *Three-Dimensional Matching* and *Partition into Triangles* problems, which can both be stated in terms of graph theory and do not involve agents with preferences. For general information on Three-Dimensional Matching and Partition into Triangles we recommend Garey and Johnson's classic textbook [43].

An early formalism of Knuth's idea was proposed by Alkan [44] in 1988. He proposed the following model (we update the terminology and notation). There exists a set N of $3n$ agents and a preference list P_{α_i} for each agent $\alpha_i \in N$. Let P be the collection of preference lists P_{α_i} for each agent α_i . Each agent has one of three types, which are called *man*, *woman*, and *dog*. There are n agents of each type, and the agents of each type are labelled $U = \{u_1, u_2, \dots, u_n\}$, $W = \{w_1, w_2, \dots, w_n\}$, and $D = \{d_1, d_2, \dots, d_n\}$ respectively. A *family* is a 3-tuple $(u_i, w_j, d_k) \in U \times W \times D$. A *matching* is a set of families where each agent in N is contained in exactly one family. Given an agent α_i and a matching M , we denote by $M(\alpha_i)$ the family in M that contains α_i . Each agent's preference list P_{α_i} describes a strict order over every pair of agents containing one agent of each of the other two types. In other words, each agent has a strict preference over all possible coalitions that they may belong to. We say that each agent $u_i \in U$ *prefers* a family (u_i, w_j, d_k) to a family $(u_i, w_{j'}, d_{k'})$ if (w_j, d_k) precedes $(w_{j'}, d_{k'})$ in

P_{u_i} . Analogous statements are true for each $w_j \in W$ and each $d_k \in D$. Given a matching M , we say that a family f is *blocking* if each agent α_i in f prefers their pair of partners in f to their pair of partners in $M(\alpha_i)$. A matching is *stable* if it does not contain a blocking family. Alkan presented an example instance of this model that contains no stable matching.

Seemingly independently of Alkan, Ng and Hirschberg [10] defined the same model in 1991 as the *Three-Gender Stable Marriage problem* (3GSM). They showed that, in contrast with (two-dimensional) SM, it is NP-complete to decide if a given instance (N, P) of 3GSM contains a stable matching. Subramanian [45] provided an alternative proof of this result in 1994, in a paper exploring an interesting relationship between certain stable matching problems and the so-called *Network Stability* problem. As defined by Subramanian, a network is similar to a circuit (defined in the standard way) except its underlying graph need not be acyclic. The Network Stability problem asks if it possible to assign boolean values to the arcs in a given network such that all gates are simultaneously satisfied. In his book, Manlove [18] reviews Subramanian's results, and also discusses some related subsequent results on the relationship between stable matching and Network Stability.

In his 2007 paper, Huang [29] proposed a variant of 3GSM in which each agent u_i has a strict preference list over W and a strict preference list over D , and compares two pairs (w_j, d_k) and $(w_{j'}, d_{k'})$ based on the sum of the ranks of $w_j, w_{j'}, d_k$, and $d_{k'}$ in these lists. Huang called this system *Precedence by Ordinal Number* (PON, which we also discussed in Section 2.2). Huang also considered the restriction of 3GSM in which preferences are *consistent*. In this system, each man u_i has underlying strict preference lists, $P_{u_i}^W$ and $P_{u_i}^D$, over the agents in W and D respectively. For any man u_i , the strict preference list P_{u_i} , which is a total order over pairs, must be a linear extension of the product order over $W \times D$ with respect to $P_{u_i}^W$ and $P_{u_i}^D$. In other words, $(w_j, d_k) >_i (w_{j'}, d_{k'})$ if and only if either (1) w_j precedes $w_{j'}$ in $P_{u_i}^W$ and $d_k = d_{k'}$, (2) d_k precedes $d_{k'}$ in $P_{u_i}^D$ and $w_j = w_{j'}$, or (3) w_j precedes $w_{j'}$ in $P_{u_i}^W$ and d_k precedes $d_{k'}$ in $P_{u_i}^D$ [18]. Similar statements are true for each woman, who has underlying lists over the agents in U and D , and each dog, who has two underlying lists over the agents in U and W . Huang showed that, in a given instance of 3GSM in which preferences are either consistent, or are additively separable and follow the PON restriction, a stable matching may not exist and the associated decision problem is NP-complete. Noting that in the PON variant agents may be indifferent between pairs of partners, he also proposed and studied a hierarchy of solution concepts related to stability. In each of the solution concepts, some number of agents in any blocking family f need only be indifferent between f and their assigned family. He showed that the existence problems relating to so-called *strongly-stable*, *super-stable*, and *ultra-stable* matchings in the PON system are all also NP-complete.

We remark that the idea of consistency is conceptually similar to the principle of *monotonic preferences* (or synonymously, *independence* [20]), one of the axioms studied by Kannai and Peleg [22] in 1984. It is also unclear whether Kannai and Peleg's result (discussed in

Section 2.2) can be related to the specific setting of 3GSM.

Another special case of 3GSM was proposed by Danilov [46], in 2003. In this model, each man u_i has an underlying preference list over the set of women W and, in the preference list P_{u_i} over $W \times D$, the pair (w_j, d_k) precedes $(w_{j'}, d_{k'})$ only if w_j precedes $w_{j'}$ in the underlying list of u_i . Similarly, each woman w_j has an underlying preference list over the set of men U , and in the preference list P_{w_j} over $U \times D$, the pair (u_i, d_k) precedes $(u_{i'}, d_{k'})$ only if u_i precedes $u_{i'}$ in the underlying list of w_j . No assumption is made about the preferences of the agents in D . Danilov showed that, in this case, Gale and Shapley's algorithm for SM can be used to find a stable matching.

In 2004, Boros et al. [47] considered another model that can also be defined as a special case of 3GSM. The authors noted that in Danilov's model the preferences are *acyclic*, and proposed a model with *lexicographically cyclic* preferences, in which the types have a cyclic order in which U precedes W , W precedes D , and D precedes U . In this model, each man u_i has an underlying preference list over $W \times D$, where (w_j, d_k) precedes $(w_{j'}, d_{k'})$ in P_{u_i} only if w_j precedes $w_{j'}$ in the underlying list of u_i . Similarly, each woman w_j has an underlying preference list over the set of dogs D , and in the preference list P_{w_j} over $U \times D$, the pair (u_i, d_k) precedes $(u_{i'}, d_{k'})$ only if d_k precedes $d_{k'}$ in the underlying list of w_j . Similarly, each dog d_k has an underlying preference list over the set of men U , and in the preference list P_{d_k} over $U \times W$, the pair (u_i, w_j) precedes $(u_{i'}, w_{j'})$ only if u_i precedes $u_{i'}$ in the underlying list of d_k . Boros et al. showed that in this model a stable matching must exist if $n \leq 2$ but need not exist if $n \geq 3$. They also proposed another model of *purely cyclic* preferences, which is very closely related but not, strictly speaking, a special case of 3GSM (since agents may be indifferent between families). This model has since been termed *Three-Dimensional Stable Matching with Cyclic preferences* (3-DSM-CYC, sometimes 3DSM [48] or c3DSM [49]).

3-DSM-CYC can be defined identically to 3GSM except the preference list P_{u_i} of each agent u_i is over the individual agents in W and is strict, the preference list P_{w_j} of each agent w_j is over the individual agents in D and is strict, and the preference list P_{d_k} of each agent d_k is over the individual agents in U and is strict. As in the model of Boros et al. [47], the types have a cyclic order in which U precedes W , W precedes D , and D precedes U . In 3-DSM-CYC, each man $u_i \in U$ prefers any pair (w_j, d_k) to any pair $(w_{j'}, d_{k'})$ if w_j precedes $w_{j'}$ in P_{u_i} , each woman $w_j \in W$ prefers any pair (u_i, d_k) to any pair $(u_{i'}, d_{k'})$ if d_k precedes $d_{k'}$ in P_{w_j} , and each dog $d_k \in D$ prefers any pair (u_i, w_j) to any pair $(u_{i'}, w_{j'})$ if u_i precedes $u_{i'}$ in P_{d_k} . A family f is *blocking* if each agent α_i in f prefers the agent of the next type in f to the agent of the next type in $M(\alpha_i)$ (with respect to the cyclic order).

Since 2004, 3-DSM-CYC has generated a great deal of interest, particularly in the area of matching under preferences. In their 2004 paper, Boros et al. [47] showed that if $n = 3$ then a stable matching must exist, but left open the case for $n \geq 4$.

In 2006, Eriksson et al. [50] extended this result to $n = 4$. They conjectured that, based on evidence from computer search, any instance (N, P) of 3-DSM-CYC contains a stable matching. Moreover, they conjectured that the minimum number of stable matchings over all instances of size n increases with n . In 2009, Biró and McDermid [48] studied two variants of 3-DSM-CYC. One involved incomplete preference lists, which is known as 3-DSMI-CYC, and the other involved complete preference lists with ties. They showed that for both variants a stable matching may not exist and the associated decision problem is NP-complete. In 2018, Escamocher and O’Sullivan [51] considered a restricted set of instances of 3-DSM-CYC in which all agents of one type have the same *master* preference list. They showed that the number of stable matchings in such an instance is exponential in n . They combined this result with an empirical study, which indicated that such instances contain the fewest stable matchings among all instances of the same size. They therefore conjectured that the number of stable matchings in an arbitrary instance is in fact exponential in n . In 2019, Pashkovich and Poirrier [49] extended the result of Eriksson et al. and showed that if $n = 5$ then a stable matching must exist. Pashkovich and Poirrier formulated instances of 3-DSM-CYC as instances of the *Satisfiability* problem, and solved them using a SAT solver.

Generalisations of 3-DSM-CYC that involve more than three types have also been studied. In 2016, Hofbauer [52] extended the result of Boros et al. [47] to show, for any $k \geq 3$, that any instance of k -DSM-CYC in which there are at most $k + 1$ agents of each type contains a stable matching, which can be found in polynomial time.

Building on previous results involving incomplete preferences, Lam and Plaxton showed in 2019 that, contrary to all previous conjectures, for any $k \geq 3$ there exist a family of instances of k -DSM-CYC that contain no stable matching, and the associated existence problem is NP-complete [5]. We remark that for $k = 3$, Lam and Plaxton identified an instance with $n = 90$ that contains no stable matching.

Nevertheless, k -DSM-CYC, and 3-DSM-CYC in particular, have continued to attract attention. In 2020, Pittel [53] presented a probabilistic analysis of k -DSM-CYC, and showed that the expected number of stable matchings in a random instance increases with n as $(n \log n)^{k-1}$. In 2022, Lerner [54] made some interesting further discoveries. His main result was an example instance of 3-DSMI-CYC in which $n = 3$ that contains no stable matching, and showed that this instance is minimal since any instance in which $n < 3$ must contain a stable matching. He also presented an instance of 3-DSM-CYC in which $n = 20$ that contains no stable matching, leaving the existence of a smaller instance with no stable matching as an intriguing open problem.

Recently in 2022, Cseh and Peters [55] considered further solution concepts related to popularity in the models of 3-DSMI-CYC and 3-DSM-CYC. They presented a number of NP-hardness results related to 3-DSMI-CYC and, notably, a polynomial-time algorithm for

a special case of 3-DSM-CYC that can find a so-called $A \cup B$ -popular matching in a given instance of 3-DSM-CYC. Also in 2022, Cseh et al. [56] presented a paper in which they develop, and then analyse, a collection of Constraint Programming models for 3-DSM-CYC. They also integrated models for problems involving fair matchings, for some common definitions of fairness.

2.3.3 Multidimensional roommates

Just as 3GSM and 3-DSM-CYC generalise (two-dimensional) SM to three dimensions, other models have been proposed that generalise SR to three (or more) dimensions. We classify such models, in which the set of agents is homogenous, as models of *Three-Dimensional Roommates* (3DR). Other models involve coalitions of a fixed size $k \geq 3$ and have been referred to as *multidimensional roommates*.

As defined originally, (two-dimensional) SR could be viewed as a hedonic game in which any feasible coalition has size two. The solution concept corresponds exactly to core stability (although research on SR predates much of the research on hedonic games and core stability). SR has since been generalised to higher dimensions which, in a similar way, correspond to hedonic games.

The earliest known model of this type is the *Three-Person Stable Assignment problem* (3PSA), which was proposed by Ng and Hirschberg [10] in 1991, as a counterpart to 3GSM. An instance of 3PSA comprises a set N of $3n$ agents and a strict preference list P_{α_i} of each agent α_i over all pairs of agents in $N \setminus \{\alpha_i\}$. Note that in terms of hedonic games, this system of preferences is equivalent to LCs (see Section 2.2). Let P be the collection of preference lists P_{α_i} for each agent α_i . A *triple* is an unordered set of three agents and that a *matching* is a partition of N into n triples. Given an agent α_i and a matching M , we denote by $M(\alpha_i)$ the triple in M that contains α_i . For any agent α_i and two triples r and s , we say that α_i prefers r to s , denoted $r \succ_{\alpha_i} s$, if $r \setminus \{\alpha_i\}$ precedes $s \setminus \{\alpha_i\}$ in P_{α_i} . Given a matching M , we say that a triple t is *blocking* if each agent α_i in t prefers t to $M(\alpha_i)$. A matching is *stable* if it does not admit a blocking triple. Ng and Hirschberg proved that, as in the case of 3GSM, a given instance (N, P) of 3PSA may not contain a stable matching and the associated existence problem is NP-complete. We remark that, unlike a general hedonic game, a blocking triple has size three, so the verification problem for both variants is solvable in $O(|N|^3)$ time so this existence problem belong to NP.

As well as the aforementioned work on 3GSM, in his 2007 paper Huang [29] considered two variants of 3PSA. In the first variant, he supposed that agents provide preference lists over pairs of agents (i.e. agents have LCs) that are *consistent*. The definition of consistent preferences in 3PSA is analogous to the definition of consistent in 3GSM (discussed in Section 2.3.2). In

3PSA, we say that preferences are consistent in an instance of 3PSA if each agent α_i has an underlying strict preference list P'_{α_i} over the agents in $N \setminus \{\alpha_i\}$. For any agent α_i , the strict preference list P_{α_i} , which is a total order over pairs, must be a linear extension of the product order over $\{(\alpha_j, \alpha_k) : (\alpha_j, \alpha_k) \in (N \setminus \{\alpha_i\})^2\}$ with respect to P'_{α_i} . In the second variant, he considers additively separable preferences under a PON restriction (see Section 2.2). He showed that, in a given instance of 3PSA in which the preferences are either consistent, or are additively separable and follow the PON restriction, a stable matching may not exist and the associated decision problem is NP-complete. As in the case of 3GSM, Huang also showed that various existence problems relating to so-called strongly-stable, super-stable, and ultra-stable matchings are NP-complete.

Later in 2007, Iwama et al. [57] considered another closely-related three-dimensional generalisation of SR. They also considered stable matchings characterised by the absence of a blocking triple. They supposed that each agent α_i has an strict preference list P_{α_i} over all agents in $N \setminus \{\alpha_i\}$, and defined implicitly a set extension rule that is similar to consistent preferences (as defined by Huang). In the model of Iwama et al., any agent α_i prefers some triple r where $\alpha_i \in r$ to another triple s where $\alpha_i \in s$ and $s \neq r$ if $r \setminus \{\alpha_i\}$ precedes $s \setminus \{\alpha_i\}$ in the product order $\{(\alpha_j, \alpha_k) : (\alpha_j, \alpha_k) \in (N \setminus \{\alpha_i\})^2\}$ with respect to P_{α_i} . We remark that, like \mathcal{B} and \mathcal{W} -preferences, this rule defines a partial order over possible coalitions even though each agent has a strict preference list over possible partners. Interestingly, this set extension rule is almost identical to the construction of a so-called power-ordered set from a partially ordered set as defined by Bossong and Schweigert [58] the previous year, in the context of formalised decision making. In 2011, Delort, Spanjaard and Weng [59] applied the aforementioned *Bossong-Schweigert extension* rule to the problem of committee forming, which is closely related to hedonic games. In 2015, Lang et al. [60] applied a generalised version of the Bossong-Schweigert rule to a model of a hedonic game.

In 2008, Iwama et al. [61] considered an optimisation problem using the same model as in their 2007 paper, and showed that it was APX-hard.

In 2009, Arkin et al. [62] presented another variant of 3PSA called *Geometric 3D-SR*. In this model, agents have additively separable preferences. The agents' valuations are derived from their relative positions in a metric space. The authors also defined a more general version of stability, specific to additively separable preferences, called α -stability, in which an agent prefers one pair to another pair if the sum of relative distances to the agents in the first pair is at least α times smaller than that of the second pair. The authors showed that a 2-stable matching always exists and can be found in polynomial time. They also presented an instance in which no (1-)stable matching exists but left open the complexity of the corresponding existence problem. In 2013, Deineko and Woeginger [63] resolved this open question by showing that this existence problem is NP-complete. In fact, Chen and Roy [64] later strengthened this result (in 2022) to show that *Geometric kD-SR* is NP-complete even when the metric space

is the Euclidean plane.

In 2014, Ostrovsky and Rosenbaum [65] revisited both 3GSM and 3PSA. For each model they considered two related optimisation problems. The first, called *Maximally Stable Matching* (MSM), involves finding a matching with the maximum number of non-blocking triples. The second, called *Maximum Stable Sub-matching* (MSS), involves finding a *sub-matching* that involves only a subset of the agents. The problem is to find a submatching of maximum size, i.e. involving as many agents as possible, which is stable when considering only blocking triples involving agents in the submatching. They showed, in the context of both 3GSM and 3PSA, that MSM and MSS are NP-hard to approximate within some fixed constant factor (i.e. that they are APX-hard [66]). Positively, they also described a simple greedy algorithm that returns constant factor approximations for both MSM and MSS for both 3GSM and 3PSA.

In 2020, Boehmer and Elkind [67] considered a number of different models of *multidimensional roommates*. In each model 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 existence problems are NP-hard, although also that many are solvable in linear time if the room size is a fixed constant. Notably, they presented an integer programming-based algorithm that can solve the stability existence problem in polynomial time.

Also in 2020, Bredereck et al. [68] considered two variations of multidimensional roommates that involve either a *master list* or *master poset*, from which all the agents' preference lists are derived. In the case of a master list, each agent obtains their preference list by deleting from the master list any pair that contains themselves. Surprisingly, the authors discovered that a stable matching may not exist in this model even if all agents' preference lists are derived from a single master list. They presented two positive results relating to restrictions of the problem involving a master poset, although they also showed, for either a master list or master poset, that in general the stability existence problem is either NP-hard or $W[1]$ -hard, in terms of three well-motivated parameters.

2.3.4 Other models

Various other models have been proposed in the literature that involve coalitions of restricted, but not necessarily fixed, size. For example, some problem models only require the size of any feasible coalition to be within some lower and upper bound. In this section we review a selection of such models.

In 2011, Aziz et al. [13] considered Pareto optimal coalition partitions, in a variety of hedonic game variants. In one variant, which has since been termed a *flatmate game*, any coalition must have size at most three. They showed that the Pareto optimality existence problem is

NP-hard in this particular model. The authors also presented polynomial-time solvability results in a model involving \mathcal{W} -preferences and for a model in which the maximum size of any coalition is two. In 2020, Brandt and Bullinger [69] presented a paper studying the existence of popular partitions in a number of hedonic game variants, including flatmate games. They applied a variety of preference structures and presenting results relating to partitions that are popular, *strongly popular*, and *mixed popular* (a *mixed popular partition* is in fact a distribution over coalition partitions).

In 2015, Wright and Vorobeychik [70] studied the *Team formation problem*, an instance of which resembles an ASHG. Taking a game-theoretic approach, they assumed that agents act competitively and strategically. They compared four different algorithms in terms of so-called *strategy-proofness*, welfare, and fairness, focussing on experimental performance rather than algorithmic complexity. Notably, their model includes constraints on the sizes of coalitions with a lower and upper bound. Other research works, mostly in the fields of economics and operations research, have since studied the Team formation problem [71].

In 2018, Sless et al. [4] proposed a model that can be viewed as a type of ASHG with symmetric preferences. They argued that a strong practical motivation exists for considering coalitions of restricted size, and thus focused on the existence of coalition partitions that contain exactly k coalitions, for some fixed $k \geq 1$. They presented a number of theoretical and empirical results relating to this model. They showed, using a connection to the so-called Min- k -Cut problem, that the problem of finding a partition with maximum utilitarian welfare can be solved in polynomial time in the restricted case in which k is fixed and there are, in a precise sense, relatively few negative edges. Otherwise, they showed that this construction problem is NP-hard. They also presented a polynomial-time solvability result for a problem in which a central organiser can add edges to the instance. Notably, they proposed a heuristic-based procedure for the problem of finding a core stable partition that maximises utilitarian welfare. They showed that this procedure performs well on instances derived from real-world data from social networks.

In 2019, Cseh et al. [31] considered Pareto optimal matchings in a general model that is comparable to a hedonic game. In this model there is a set of rooms with integer sizes, and any coalition must be allocated to exactly one room where the size of the room is exactly the size of the coalition. They studied two specific variants of this model, applying \mathcal{B} - and \mathcal{W} -preferences. They showed if \mathcal{B} -preferences are used and the agents' preference lists are strict then a polynomial-time algorithm based on serial dictatorship can be used to construct Pareto optimal matchings in polynomial time. They also showed that, in a number of other circumstances, a Pareto optimal matching may not exist and that in many cases the associated existence problems are either NP-hard or NP-complete.

In 2022, Li et al. [72], considered a model that can be viewed as a generalisation of multi-

dimensional roommates with binary and symmetric additively separable preferences. Rather than a fixing the size of a coalition, a partition of the set of n agents must be *balanced*, meaning the number of coalitions $k \leq n$ is fixed and $\lfloor n/k \rfloor \leq |S| \leq \lceil n/k \rceil$ for any coalition S in a feasible partition. They studied an approximation of envy-freeness, termed EF- r , in which the utility gained by any envious agent may be up to r in a feasible partition, for some fixed $r \geq 0$ (our definition of envy-freeness is thus EF-0). Interestingly, they applied results from discrepancy theory to show that an approximate-envy free partition with a particular fixed asymptotic bound must exist, and can be found in polynomial time. They also considered restricted sets of instances, such those in which the underlying structure is a tree. They showed that in such an instance, an EF-1 partition must exist and can be found in polynomial time. A coalition partition of $3n$ agents in which every coalition has size three is by definition balanced, so some of the algorithmic results relating to approximate envy-free partitions [72, Theorems 9 and 10] also apply in a more restricted model of 3DR with additively separable preferences (such as the model that we define in Chapter 6).

Also in 2022, Bilò et al. [73] proposed another model, which can be viewed as a type of ASHG. They analogized their model to a dinner party situation in which n agents are assigned to exactly k tables. They argued that Nash stability and core stability might not make sense in such a setting, since there is no free table. Instead, they considered three successively weaker solution concepts involving two agents that swap places. In a partition that is *strictly swap stable*, no two agents α_i, α_j exist where if α_i and α_j swap places then the utilities of both agents strictly increase. In a partition that is *swap stable*, no two agents α_i, α_j exist where if α_i and α_j swap places then the utility of α_i strictly increases and the utility of α_j does not decrease. In a partition π that is *swap stable under transferable utilities*, no two agents α_i, α_j exist where if α_i and α_j swap places, in a new partition π' , then $u_{\alpha_i}(\pi') + u_{\alpha_j}(\pi') > u_{\alpha_i}(\pi) + u_{\alpha_j}(\pi)$. They remarked that a variation of these solution concepts had been previously studied in the context of the Stable Marriage problem as *exchange stability* [74]. We remark here that envy-freeness implies strict swap stability. In their paper, Bilò et al. show that for any of the three concepts, a feasible partition must exist and can be found by iteratively executing improving swaps from an arbitrary starting partition. Notably, they showed that the relevant existence problem is PLS-complete in general but the iterative process converges in polynomial time if preferences are binary. They also analysed the problem of maximising utilitarian welfare, and the so-called *price of anarchy*, and *price of stability*, which are defined as follows. For any of the three stability concepts, the price of anarchy (stability) is the worst- (best-)case ratio between the utilitarian welfare of an arbitrary stable matching and the maximum possible utilitarian welfare over all possible partitions.

Chapter 3

Approximability of Three-Dimensional Stable Matching with Cyclic Preferences

3.1 Introduction

In this chapter we study the approximability of the Three-Dimensional Stable Matching problem with Cyclic Preferences (3-DSM-CYC, also known as 3DSM [48] or c3DSM [49]).

As we saw in Chapter 2, the question of whether every instance of 3-DSM-CYC contains a stable matching was open for several decades. It was only in 2019 that Lam and Plaxton [5] showed that a given instance of 3-DSM-CYC need not contain a stable matching and that the associated decision problem is NP-complete. A natural next step is to consider approximately stable matchings, as we do here. To our knowledge, we present the first theoretical results on the approximability of 3-DSM-CYC.

In this chapter we consider the optimisation problem of finding a matching with the maximum number of non-blocking families, which we call the 3-DSM-CYC Maximally Stable Matching problem (3-DSM-CYC-MSM).

We begin, in Section 3.2, by showing that an existing approximation algorithm for 3GSM-MSM, which is a closely related problem, can be used to devise a $9/4$ -approximation algorithm for 3-DSM-CYC-MSM (Theorem 3.1). We then show that a simple algorithm based on serial dictatorship gives an improved approximation ratio of $6/5$ (Theorem 3.2).

Next, in Section 3.3, we consider a situation in which there exists a master preference list over all agents of one type, and quantify the similarity between the preference list of any agent of the previous type and that master list, in terms of a specific distance metric. We extend the approximation algorithm for 3-DSM-CYC-MSM and show that if the maximum

distance is sufficiently small then as it is further reduced, the approximation ratio of the algorithm decreases from $6/5$ to 1 (Theorem 3.3). As a corollary, we show that if every agent of one type has the same preference list then the algorithm returns a matching that is stable (Corollary 3.1, which is also implied by a result of Escamocher and O’Sullivan [51]).

Finally, in Section 3.4, we recap on our results and discuss some directions for future work.

We proceed with some formal definitions and notation. An instance of 3-DSM-CYC comprises a set N of $3n$ agents and a strict preference list for each agent α_i , labelled P_{α_i} . Each agent in N has one of three *types*, which we call *man*, *woman*, and *dog*. There are n agents of each type, and the agents of each type are labelled $U = \{u_1, u_2, \dots, u_n\}$, $W = \{w_1, w_2, \dots, w_n\}$, and $D = \{d_1, d_2, \dots, d_n\}$. The types have a cyclic order in which W follows U , D follows W , and U follows D . Each agent’s preference list P_{α_i} describes a strict order all agents in the next type. We say that an agent α_i *prefers* β_j to β_k if β_j precedes β_k in the preference list P_{α_i} of α_i . A *family* is a 3-tuple $(u_i, w_j, d_k) \in U \times W \times D$. A *matching* is a set of families where each agent in N is contained in exactly one family. Given an agent α_i and a matching M , we denote by $M(\alpha_i)$ the family in M that contains α_i . Given a matching M , we say that a family f is *blocking* if each agent α_i in f prefers the agent of the next type in f to the agent of the next type in $M(\alpha_i)$. A matching is *stable* if it does not contain a blocking family. Let P be the collection of preference lists P_{α_i} for each agent α_i . For any instance (N, P) of 3-DSM-CYC and any matching M , we denote by $\text{bf}(M, (N, P)) \subseteq U \times W \times D$ the set of families that block M in (N, P) . Conversely, we denote by $\text{nbf}(M, (N, P)) = (U \times W \times D) \setminus \text{bf}(M, (N, P))$ the set of families that do not block M in (N, P) . When the instance in question is clear from context, we simply write $\text{bf}(M)$ or $\text{nbf}(M)$. Formally, 3-DSM-CYC-MSM is the optimisation variant of 3-DSM-CYC in which the objective is to maximise $|\text{nbf}(M, (N, P))|$.

3.2 Unrestricted preferences

In this section we consider the approximability of 3-DSM-CYC-MSM in the general case. We first apply an existing polynomial-time approximation algorithm for a related problem and construct a polynomial-time $9/4$ -approximation algorithm. We then improve on this result and show that a polynomial-time algorithm based on serial dictatorship constitutes a $6/5$ -approximation algorithm.

There is a close relationship between 3-DSM-CYC and the *Three-Gender Stable Marriage Problem* (3GSM, introduced in Chapter 2). Recall that in 3-DSM-CYC, each agent has a strict preference list over all agents of the next type, and compares two families based on the relative rank of the two agents of the next type in each family. In 3GSM, each agent has a strict preference list over all possible families that they belong to, and compares any two families using on this preference list. Although neither problem is a strict generalisation of

the other, we show that, given an instance (N, P) of 3-DSM-CYC, it is possible to construct an instance (N, P') of 3GSM with the same set of agents such that for any matching M , if a family blocks M in (N, P) then it also blocks M in (N, P') . From this result it follows that an existing 9/4-approximation algorithm for 3GSM-MSM, which is defined analogously to 3-DSM-CYC-MSM, can be applied to construct a 9/4-approximation algorithm for 3-DSM-CYC-MSM. The existing 9/4-approximation algorithm for 3GSM-MSM was presented by Rosenbaum in 2016 [75] and is called Algorithm AMSM. It is an iterative greedy algorithm which involves, in each iteration, selecting a family that once added to the matching, intersects the maximum number of non-blocking families.

Theorem 3.1. *There exists a polynomial-time 9/4-approximation algorithm for 3-DSM-CYC-MSM.*

Proof. The approximation algorithm for 3-DSM-CYC-MSM involves constructing a corresponding instance of 3GSM-MSM, which has the same set of agents, and running Algorithm AMSM [75].

We first describe how to construct, in polynomial time, a corresponding instance (N, P') of 3GSM such that $|\text{nbf}(M, (N, P))| \geq |\text{nbf}(M, (N, P'))|$ for any matching M . For each agent u_i in U let P'_{u_i} be the ordered list of tuples in $W \times D$ such that if w_j precedes w_i in P_{u_i} then every family containing w_j appears before every family containing w_i . The relative order of the families in P'_{u_i} that contain the same agent in W is arbitrary. Construct the preferences of each agent in W and D symmetrically. Now, suppose M is an arbitrary matching in (N, P) and (u_i, w_j, d_k) is a family that blocks M in (N, P) . We claim that this family also blocks M in (N, P') . First, assume without loss of generality that $M(u_i) = (u_i, w_i, d_i)$. By definition, it must be that w_j appears before w_i in P_{u_i} . It follows by the construction of P' that (u_i, w_j, d_k) appears before (u_i, w_i, d_i) in P'_{u_i} and thus that (u_i, w_j, d_k) also blocks M in (N, P') . It follows immediately that $|\text{bf}(M, (N, P'))| \geq |\text{bf}(M, (N, P))|$ and thus that $|\text{nbf}(M, (N, P))| \geq |\text{nbf}(M, (N, P'))|$, as required.

Since Rosenbaum's [75] analysis of Algorithm AMSM shows that $|\text{nbf}(M, (N, P'))| \geq 4n^3/9$, we can conclude that $|\text{nbf}(M, (N, P))| \geq 4n^3/9$. \square

We now consider an algorithm for 3-DSM-CYC-MSM based on serial dictatorship, which we call Algorithm `cyclicSerialDictatorship`, shown in Algorithm 3.1. In the context of computational social choice, serial dictatorship refers to a type of iterative algorithm used to assign resources to agents. Typically, in each iteration a dictator is selected who is then assigned their most-preferred resource that is still available. The classical application of serial dictatorship is for the problem of *House allocation* [11] but it has also been applied to hedonic games [76] and problems involving coalitions of fixed size [31].

The algorithm we present here for 3-DSM-CYC-MSM follows the pattern of serial dictatorship. First, an arbitrary yet-unmatched agent in U is selected and labelled u_i . Next, u_i selects their most-preferred yet-unmatched agent in W which is labelled w_j . Finally, w_j selects their most-preferred yet-unmatched agent in D which is labelled d_k . The family (u_i, w_j, d_k) is then added to the matching. A similar algorithm was in fact used by Hofbauer [52] in order to construct stable matchings in the setting of k -DSM-CYC where $n \leq k + 1$.

Algorithm 3.1 Algorithm cyclicSerialDictatorship

Input: an instance (N, P) of 3-DSM-CYC

Output: a matching M in (N, P)

```

 $V \leftarrow N$ 
 $M \leftarrow \emptyset$ 
while  $|V| > 0$  do
   $u_i \leftarrow$  any agent in  $U \cap V$ 
   $w_j \leftarrow$  the most-preferred agent in  $W \cap V$  according to  $P_{u_i}$ 
   $d_k \leftarrow$  the most-preferred agent in  $D \cap V$  according to  $P_{w_j}$ 
   $M \leftarrow M \cup \{(u_i, w_j, d_k)\}$ 
   $V \leftarrow V \setminus \{u_i, w_j, d_k\}$ 
end while
return  $M$ 

```

It is straightforward to show that Algorithm cyclicSerialDictatorship returns a matching M in polynomial time. We now analyse its approximation ratio and show that our analysis is tight. This involves placing an upper bound on the number of blocking families in a matching M returned by the algorithm. To do this, we consider each family in M in the order that they were added to M in the algorithm and count only the blocking families that intersect that family and do not intersect any previous family. To simplify the analysis, without loss of generality suppose the (u_1, w_1, d_1) was the first family added to M , (u_2, w_2, d_2) was the second family added to M , and so on.

Theorem 3.2. *There exists a polynomial-time $6/5$ -approximation algorithm for 3-DSM-CYC-MSM.*

Proof. There are exactly n iterations of the while loop so it is straightforward to show that the algorithm runs in polynomial time. For each i where $1 \leq i \leq n$, let $F_i = (u_i, w_i, d_i)$ be the family added to M in the i^{th} iteration. Let U_i , W_i , and D_i be the set of agents in $U \cap V$, $W \cap V$, and $D \cap V$ respectively at the start of the i^{th} iteration. It follows that $U_i = \{u_i, u_{i+1}, \dots, u_n\}$, $W_i = \{w_i, w_{i+1}, \dots, w_n\}$, and $D_i = \{d_i, d_{i+1}, \dots, d_n\}$.

For each i where $1 \leq i \leq n$, let S_i be the set of families that block M , have a non-empty intersection with F_i , and have an empty intersection with F_j for every $1 \leq j < i$. It follows that $S_n = \emptyset$ since any family that blocks M and intersects the final family F_n must contain

some agent not in F_n , which must belong to some previous family F_j where $1 \leq j < n$. We can now define $\text{bf}(M)$ in terms of S_i :

$$\text{bf}(M) = \bigcup_{i=1}^{n-1} S_i .$$

By definition, the sets S_i are pairwise disjoint so it follows that

$$|\text{bf}(M)| = \sum_{i=1}^{n-1} |S_i| . \quad (3.1)$$

We now place an upper bound on $|S_i|$ for any i where $1 \leq i \leq n-1$. For any such i , consider the i^{th} iteration of the while loop. By the algorithm, w_i is the most-preferred agent in W_i according to P_{u_i} . It follows that any family that blocks M and contains u_i must also contain some agent in W that has already been added to a family F_j in M where $j < i$. By the definition of S_i , it follows that no family in S_i contains u_i . Similarly, since d_i is the most-preferred agent in D_i according to P_{w_i} , any family in S_i that contains w_i must also contain some agent in D that has already been added to a family F_j in M , where $j < i$. It follows similarly that no family in S_i contains w_i .

It remains that every family in S_i contains d_i . Consider an arbitrary family (u_j, w_k, d_i) in S_i . By the definition of S_i it must be that $j > i$ and $k > i$. Note that for any choice of u_j , by assumption u_j prefers w_k to w_j . It follows by the algorithm that $w_k \notin W_j$ and thus $k < j$.

In conclusion, we have shown that for any family $(u_j, w_k, d_i) \in S_i$ it must be that $i < j \leq n$ and $i < k < j$. We can now count $|S_i|$ by considering each possible value of j , from $i+1$ to n inclusive and each possible value of k , of which there are $j-i-1$. It follows that

$$\begin{aligned} |S_i| &\leq \sum_{j=i+1}^n (j-i-1) \\ &= \frac{(n-i)(n-i-1)}{2} . \end{aligned} \quad (3.2)$$

Now

$$\begin{aligned} |\text{bf}(M)| &= \sum_{i=1}^{n-1} |S_i| && \text{Equation 3.1} \\ &\leq \sum_{i=1}^{n-1} \frac{(n-i)(n-i-1)}{2} && \text{by Inequality 3.2} \\ &= \frac{n^3}{6} - \frac{n^2}{2} + \frac{n}{3} . \end{aligned} \quad (3.3)$$

Suppose M^* is a matching in (N, P) with the maximum number of non-blocking families. The approximation ratio of the algorithm is thus

$$\begin{aligned}
 \frac{|\text{nbf}(M^*)|}{|\text{nbf}(M)|} &\leq \frac{n^3}{|\text{nbf}(M)|} \\
 &= \frac{n^3}{n^3 - |\text{bf}(M)|} && \text{by the definition of nbf} \\
 &\leq \frac{6n^2}{5n^2 + 3n - 2} && \text{by Inequality 3.3} \\
 &\leq \frac{6}{5} && \text{since } n \geq 1.
 \end{aligned}$$

□

It is desirable to show that the analysis is tight, for example by constructing an instance (N, P) such that there exists some execution of the algorithm that returns a matching M for which $|\text{nbf}(M^*)|/|\text{nbf}(M)| = 6/5$, where M^* is some matching in (N, P) with the maximum number of non-blocking families. We show that this analysis is tight asymptotically, by constructing an instance \mathcal{I}_n for some fixed $n \geq 1$, where the approximation ratio obtained by Algorithm `cyclicDicatorship` on \mathcal{I}_n in the worst case is $6/5 - o(1)$.

The structure of the preferences of the agents in \mathcal{I}_n corresponds directly to the counting argument used in the proof of Theorem 3.2. For any fixed n , construct \mathcal{I}_n as follows. Let $U = \{u_1, u_2, \dots, u_n\}$, $W = \{w_1, w_2, \dots, w_n\}$, and $D = \{d_1, d_2, \dots, d_n\}$, and for each i where $1 \leq i \leq n$ let

$$\begin{aligned}
 P_{u_i} &: w_1 \ w_2 \ \dots \ w_n \\
 P_{w_i} &: d_1 \ d_2 \ \dots \ d_n \\
 P_{d_i} &: u_n \ u_{n-1} \ \dots \ u_1
 \end{aligned}$$

In the algorithm the selection of each agent in $U \cap V$ is arbitrary, so suppose the algorithm selects u_1 in the first iteration, u_2 in the second iteration, and so on. It follows that the first family added to M is (u_1, w_1, d_1) and $V = N \setminus \{u_1, w_1, d_1\}$ at the start of the second iteration. It then follows that the next family added to M is (u_2, w_2, d_2) . In general, in the i^{th} iteration w_i must be the most-preferred agent in $W \cap V$ according to P_{u_i} and d_i must be the most-preferred agent in $D \cap V$ according to P_{w_i} . It follows that the algorithm returns $M = \{(u_1, w_1, d_1), (u_2, w_2, d_2), \dots, (u_n, w_n, d_n)\}$. As in the proof of Theorem 3.2, for each i where $1 \leq i \leq n$, let F_i be the family (u_i, w_i, d_i) selected in the i^{th} iteration and let S_i be the set of families that block M , have a non-empty intersection with F_i and an empty intersection with F_j for every $1 \leq j < i$. Note that as before, $S_n = \emptyset$. As before, it can be shown that for any i where $1 \leq i \leq n$, no family in S_i contains u_i and no family in S_i contains w_i . It

remains that each family in S_i contains d_i . Note that by the construction of \mathcal{I}_n , d_i prefers to u_i each agent u_j where $j > i$. Moreover, any such u_j prefers to w_j each agent w_k where $k < j$, and any such w_k prefers to d_k each agent d_i where $i < k$. It follows that S_i contains the family (u_j, w_k, d_i) for each j where $i < j \leq n$ and each k where $i < k < j$. As in the proof of Theorem 3.2, it is straightforward to show that no other families are contained in S_i . It follows that since

$$S_i = \{(u_j, w_k, d_i) : i < j \leq n \text{ and } i < k < j\}$$

it must be that

$$|S_i| = \sum_{j=i+1}^n (j - i - 1)$$

which shows that the upper bound on $|S_i|$ shown in Inequality 3.2 in Theorem 3.2 is tight. The same argument used in the proof of Theorem 3.2 then shows that

$$|\text{nbf}(M)| = \frac{5n^3}{6} + \frac{n^2}{2} - \frac{n}{3}. \quad (3.4)$$

We now show that a stable matching exists in \mathcal{I}_n . Let $M^* = \{(u_i, w_{n-i+1}, d_{n-i+1}) : 1 \leq i \leq n\}$. Suppose for a contradiction that M^* is not stable and thus some family (u_j, w_k, d_i) blocks M^* in (N, P) . By the definition of a blocking family, u_j prefers w_k to w_{n-j+1} . By the preference list of u_j it follows that $k < n - j + 1$. Similarly, since w_k prefers d_i to d_k by the preference list of w_k it must be that $i < k$. Since $k < n - j + 1$ and $i < k$ it follows that $j < n - i + 1$. By the construction of P_{d_i} , it follows that u_j appears after u_{n-i+1} in P_{d_i} . This is a contradiction since by the definition of a blocking family d_i must prefer u_j to its assigned partner of the next type, u_{n-i+1} . It follows that M^* is stable. Now

$$\begin{aligned} \frac{|\text{nbf}(M^*)|}{|\text{nbf}(M)|} &= \frac{n^3}{|\text{nbf}(M)|} && \text{since } M^* \text{ is stable} \\ &= \frac{6n^2}{5n^2 + 3n - 2} && \text{by Equation 3.4} \\ &= \frac{6}{5} - \frac{18n - 12}{25n^2 + 15n - 10} \\ &= \frac{6}{5} - o(1) \end{aligned}$$

which shows that the analysis is asymptotically tight.

Since all agents of the same type in \mathcal{I}_n had the same preference list, it was straightforward to construct a stable matching. In the next section, we consider instances in which the preference lists of all agents of at least one type are derived from a master list, and consider the problem

of finding a matching with the maximum number of non-blocking families.

3.3 Preferences derived from a master list

In this section, we consider a situation in which the preference lists of all agents in at least one of the sets U, W, D are in some way similar. Specifically, we consider a situation in which there exists a master preference list over all agents of one type, and quantify the similarity between the preference list of any agent of the previous type and that master list, in terms of a specific distance metric. We extend Algorithm `cyclicSerialDictatorship` for this problem and show that if this distance is sufficiently small then as the distance is reduced the approximation ratio of the algorithm decreases from $6/5$ to 1 .

The assumption of a master list has been made in a number of problems involving matching under preferences [18], including the multidimensional roommates problem [68]. In the setting of 3-DSM-CYC, Escamocher and O’Sullivan [51] showed in 2018 that if all agents of one type have the same preference list then the number of stable matchings in that instance is exponential in n . In a paper on Constraint Programming models for 3-DSM-CYC, Cseh et al. [56] showed that the serial dictatorship-style algorithm that we present here can also be used to construct *strongly stable* matchings in 3-DSM-CYC.

Suppose hereafter that all agents of at least one type have preferences derived from a master list. Without loss of generality, suppose the preference lists of all agents in D are close to, relative to a measure that will be defined, some master list \hat{P} over all agents in U .

We quantify the similarity between the preference list of each agent in D to the master list \hat{P} . Here we use the *Kendall tau distance* [6] (also known as *bubblesort distance* or *Kemeny distance* [11]) two linear orders over some set S . For any two elements s_i, s_j in S , we say that (s_i, s_j) is a *discordant pair* between L_1 and L_2 if $s_i \succ_{L_1} s_j$ and $s_j \succ_{L_2} s_i$. The Kendall tau distance $\tau(L_1, L_2)$ between L_1 and L_2 is the number of discordant pairs between L_1 and L_2 . The Kendall tau distance can be equivalently defined as the number of swaps made by the bubblesort algorithm when sorting L_1 according to the order of the elements in L_2 (or vice-versa) [6]. By definition, $0 \leq \tau(L_1, L_2) \leq \binom{n}{2}$ for any L_1 and L_2 .

Suppose the maximum Kendall tau distance between \hat{P} and the preference list P_{d_i} of any agent $d_i \in D$ is c .

Theorem 3.3. *If $c \leq n$ then there exists a polynomial-time algorithm for 3-DSM-CYC-MSM with approximation ratio $6/(6 - (3d^2 - 2d^3))$, where $d = c/n$.*

Proof. The algorithm is a variation of Algorithm `cyclicSerialDictatorship`. The only difference is that in each round, instead of selecting a dictator agent in U arbitrarily, the most-preferred yet-unmatched agent in U is selected according to the master list \hat{P} . Intuitively, it

follows that u_i is among the most-preferred agents in the preference list of d_k and thus the number of blocking families that contain d_k is minimised.

It is straightforward to show that this algorithm returns a matching M in polynomial time. Without loss of generality, assume that $\hat{P} : u_1 u_2 \dots u_n$ and that (u_1, w_1, d_1) was the first family added to M , (u_2, w_2, d_2) was the second family added to M , and so on.

For an arbitrary i where $1 \leq i \leq n - 1$, consider the i^{th} iteration of the while loop. As before in the proof of Theorem 3.2, it must be that each family in S_i contains d_i . Also as before, for any family (u_j, w_k, d_i) in S_i it must be that $i < j \leq n$ and $i < k < j$ so

$$\begin{aligned} |S_i| &\leq \sum_{j=i+1}^n (j - i - 1) \\ &= \frac{(n - i)(n - i - 1)}{2} . \end{aligned} \quad (3.5)$$

We now show that because of the master list, if $i < n - c$ then the upper bound on $|S_i|$ shown in Inequality 3.5 can be improved. Consider an arbitrary i where $1 \leq i < n - c$.

Consider some family (u_j, w_k, d_i) in S_i . As before in the proof of Theorem 3.2 it must be that $i < j$ and $i < k < j$.

Suppose for a contradiction that $j > i + c$ and thus that $j \geq i + c + 1$. We shall identify a number of discordant pairs between P_{d_i} and the master list \hat{P} . By assumption, the position of u_i in the master list \hat{P} is exactly i . Suppose \hat{i} is the position of u_i in P_{d_i} . It follows that the number of discordant pairs between P_{d_i} and \hat{P} that contain u_i is at least $|\hat{i} - i|$. Similarly, the number of discordant pairs between P_{d_i} and \hat{P} that contain u_j is at least $|\hat{j} - j|$ where \hat{j} is the position of u_j in P_{d_i} . At most one discordant pair contains both u_j and u_i so the total number of discordant pairs containing either u_j or u_i is at least

$$\begin{aligned} |\hat{i} - i| + |\hat{j} - j| - 1 &= |\hat{i} - i| + |j - \hat{j}| - 1 \\ &\geq \hat{i} - i + j - \hat{j} - 1 \\ &\geq \hat{i} + c - \hat{j} && \text{since } j \geq i + c + 1 \\ &\geq c + 1 && \text{since } \hat{j} < \hat{i} \end{aligned}$$

which contradicts the definition of c as the maximum number of discordant pairs between the preference list of any agent in D and the master list \hat{P} . It follows that $j \leq i + c$. In conclusion,

we have shown that $i < j \leq i + c$ and $i < k < j$. It follows that, for any i where $1 \leq i < n - c$,

$$\begin{aligned} |S_i| &\leq \sum_{j=i+1}^{i+c} (j - i - 1) \\ &= \frac{c(c-1)}{2} . \end{aligned} \quad (3.6)$$

By Inequality 3.1,

$$\begin{aligned} |\text{bf}(M)| &= \sum_{i=1}^{n-1} |S_i| && \text{as before in the proof of Theorem 3.2} \\ &\leq \sum_{i=1}^{n-c-1} \frac{c(c-1)}{2} + \sum_{i=n-c}^{n-1} \frac{(n-i)(n-i-1)}{2} && \text{by Inequalities 3.5 and 3.6} \\ &= \frac{1}{2}c(n-c-1)(c-1) + \frac{1}{6}(c^3 - c) . \end{aligned} \quad (3.7)$$

Suppose M^* is a matching in (N, P) with the maximum number of non-blocking families. By the definition of nbf,

$$\begin{aligned} |\text{nbf}(M)| &= n^3 - |\text{bf}(M)| \\ &\geq n^3 - \frac{1}{2}c(n-c-1)(c-1) - \frac{1}{6}(c^3 - c) && \text{by Inequality 3.7} \end{aligned} \quad (3.8)$$

so the approximation ratio is

$$\begin{aligned} \frac{|\text{nbf}(M^*)|}{|\text{nbf}(M)|} &\leq \frac{n^3}{|\text{nbf}(M)|} \\ &\leq \frac{6n^3}{6n^3 - 3nc^2 + 2c^3 + 3nc - 2c} && \text{by Inequality 3.8} \\ &\leq \frac{6n^3}{6n^3 - 3nc^2 + 2c^3} && \text{since } n \geq 1 \\ &= \frac{6n^3}{6n^3 - c^2(3n - 2c)} \\ &= \frac{6}{6 - (3d^2 - 2d^3)} && \text{where } d = c/n. \end{aligned}$$

□

A consequence of Inequality 3.7 in the proof of Theorem 3.3 is the following corollary. This corollary is also implied by Escamocher and O'Sullivan's [51] result that the number of stable matchings in such an instance of 3-DSM-CYC is exponential in n .

Corollary 3.1. *Any instance of 3-DSM-CYC contains a stable matching, which can be found in polynomial time, if all agents of at least one type have the same preference list.*

Proof. Assume without loss of generality that all agents in D have the same preference list. Suppose M is a matching returned by the variant of Algorithm cyclicSerialDictatorship described in Theorem 3.3. By Inequality 3.7 in the proof of Theorem 3.3, since $c = 0$ it must be that $|\text{bf}(M)| = 0$ and thus that M is stable. \square

As before in the case of unrestricted preferences, we show that this analysis is tight asymptotically by constructing an instance \mathcal{I}_n for some fixed $n \geq 1$ where for any fixed $c \leq n$ the approximation ratio obtained by the algorithm is $6/(6 - (3d^2 - 2d^3)) - o(1)$, where $d = c/n$. For any fixed n , construct \mathcal{I}_n as follows. Let $U = \{u_1, u_2, \dots, u_n\}$, $W = \{w_1, w_2, \dots, w_n\}$, and $D = \{d_1, d_2, \dots, d_n\}$, and for each i where $1 \leq i \leq n$ let

$$\hat{P} : u_1 \ u_2 \ \dots \ u_n$$

and

$$\begin{aligned} P_{u_i} &: w_1 \ w_2 \ \dots \ w_n \\ P_{w_i} &: d_1 \ d_2 \ \dots \ d_n \end{aligned}$$

To construct P_{d_i} for each $d_i \in D$, first construct $P_{d_i} : u_1 \ u_2 \ \dots \ u_n$. If $i \leq n - c$ then shift u_i in P_{d_i} to the right by c places so that it appears just after u_{i+c} . If $i > n - c$ then shift u_i in P_{d_i} to the right by $n - i$ places so that it appears last. Note that now d_i prefers to u_i any agent u_j where either $j < i$ or $i + 1 \leq j \leq i + c$.

Notice that for any $d_i \in D$ each discordant pair between \hat{P} and P_{d_i} comprises (u_i, u_{i+j}) where $j \leq c$. It follows that there are exactly c discordant pairs between \hat{P} and P_{d_i} for each d_i .

It is straightforward to show that the algorithm returns a matching $M = \{(u_1, w_1, d_1), (u_2, w_2, d_2), \dots, (u_n, w_n, d_n)\}$. Let F_i be the family (u_i, w_i, d_i) selected in the i^{th} iteration and let S_i be the set of families that block M , have a non-empty intersection with F_i and an empty intersection with F_j for every $1 \leq j < i$. Note that $S_n = \emptyset$.

As before in the proof of Theorem 3.3, we consider two cases. First, suppose $i < n - c$. As before, it can be shown that no family in S_i contains u_i and no family in S_i contains w_i . It follows that each family in S_i contains d_i . As we noted, by the construction of \mathcal{I}_n it must be that d_i prefers to u_i each agent u_j where either $j < i$ or $i + 1 \leq j \leq i + c$. By the definition of S_i no family in S_i contains any agent u_j where $j < i$ so it follows that any family in S_i contains d_i as well as some u_j where $i + 1 \leq j \leq i + c$. Moreover, any such u_j prefers to w_j each agent w_k where $k < j$, and any such w_k prefers to d_k each agent d_i where $i < k$. It follows that S_i contains the family (u_j, w_k, d_i) for each j where $i + 1 \leq j \leq i + c$ and each k

where $i < k < j$. By the proof of Theorem 3.3, it is straightforward to show that no other families are contained in S_i . It follows that

$$S_i = \{(u_j, w_k, d_i) : i + 1 \leq j \leq i + c, i < k < j\}$$

and thus that

$$|S_i| = \sum_{j=i+1}^{i+c} (j - i - 1) .$$

Second, suppose $i \geq n - c$. It follows in this case that

$$|S_i| = \sum_{j=i+1}^n (j - i - 1) .$$

It follows, as in the proof of Theorem 3.3, that

$$\begin{aligned} |\text{bf}(M)| &= \sum_{i=1}^{n-c-1} \frac{c(c-1)}{2} + \sum_{i=n-c}^{n-1} \frac{(n-i)(n-i-1)}{2} \\ &= \frac{1}{2}c(n-c-1)(c-1) + \frac{1}{6}(c^3 - c) \end{aligned}$$

which shows that the upper bound on $|S_i|$ shown in Inequality 3.2 in Theorem 3.2 is tight. It then follows that

$$|\text{nbf}(M)| = n^3 - \frac{1}{2}c(n-c-1)(c-1) - \frac{1}{6}(c^3 - c) . \quad (3.9)$$

To see that at least one stable matching M^* exists in \mathcal{I}_n , consider a new instance \mathcal{I}'_n with sets U' , W' , and D' constructed as for \mathcal{I}_n except permuting the types of the agents so that every agent in D' has the same preference list. It follows by Corollary 3.1 that \mathcal{I}'_n contains a stable matching, which can be relabelled to reveal a stable matching M^* in \mathcal{I}_n .

Now

$$\begin{aligned} \frac{|\text{nbf}(M^*)|}{|\text{nbf}(M)|} &= \frac{n^3}{|\text{nbf}(M)|} && \text{since } M^* \text{ is stable} \\ &= \frac{6n^3}{6n^3 - 3nc^2 + 2c^3 + 3nc - 2c} && \text{by Equation 3.9} \end{aligned}$$

$$\begin{aligned}
&= \frac{6n^3}{6n^3 - 3nc^2 + 2c^3} - \frac{18cn^4 - 12cn^3}{36n^6 + (18c - 36c^2)n^4} \\
&\quad + (24c^3 - 12c)n^3 \\
&\quad + (9c^4 - 9c^3)n^2 \\
&\quad + (6c^3 - 12c^5 + 6c^4)n \\
&\quad + 4c^6 - 4c^4 \\
&= \frac{6n^3}{6n^3 - 3nc^2 + 2c^3} - o(1) \\
&= \frac{6}{6 - (3d^2 - 2d^3)} - o(1) \quad \text{where } d = c/n
\end{aligned}$$

which shows that the analysis is tight asymptotically.

3.4 Summary and open problems

In this chapter we considered the approximability of the 3-DSM-CYC Maximally Stable Matching problem (3-DSM-CYC-MSM). We first presented a $9/4$ -approximation algorithm based on an existing algorithm for 3GSM-MSM, which is a closely related problem [75]. We then presented a $6/5$ -approximation algorithm based on serial dictatorship, and showed that our analysis is tight asymptotically. Finally, we considered a situation in which the preference lists of all agents of at least one type are derived from some master list, and modified the aforementioned serial dictatorship algorithm for this setting. We considered the maximum Kendall tau distance c between any such agent's list and the master list, and showed that if $c \leq n$ then the modified algorithm has an approximation ratio of $6/(6 - (3d^2 - 2d^3))$, where $d = c/n$, which is tight asymptotically.

As we saw in Chapter 2, the history of 3-DSM-CYC spans several decades and it continues to generate interesting research. Given the recent result of Lam and Plaxton [5] (who showed that stable matchings need not exist in general, and the associated decision problem is NP-complete), it seems most natural to consider either the approximability of 3-DSM-CYC, as we do here, or parameterised complexity.

In the optimisation version of 3-DSM-CYC that we studied here, the objective is to maximise the number of non-blocking families. Of course, one could also define a complementary problem in which the objective is to minimise the number of blocking families, which is arguably more natural. In fact, similar optimisation problems, in which the objective involves minimising the number of blocking coalitions, have been studied in relation to other problems of matching under preferences [77, 78, 79]. Another possibility is to construct a sub-matching of maximum cardinality such that no three agents in families in the sub-matching form a

blocking family in the sub-matching. Rosenbaum [75] refers to the analogous problem for 3GSM as the *3G Maximum Stable Sub-matching problem* (3G-MSS).

Although the definitions of the maximisation and minimisation variants of 3-DSM-CYC are complementary, it seems as if the difficulty of characterising instances with no stable matching makes tackling the latter variant more challenging.

Chapter 4

Three-Dimensional Stable Roommates with \mathcal{B} -preferences

4.1 Introduction

In this chapter we study a new model of fixed-size coalition formation, which we call the *Three-Dimensional Roommates with \mathcal{B} -preferences* (3DR-B). This model is closely related to some of the existing models of Three-Dimensional Roommates (3DR) discussed in Chapter 2, and in particular the *Three-Person Stable Assignment problem* (3PSA) [10] and the model of Iwama et al. [57]. As in the model of Iwama et al., in 3DR-B each agent has a strict preference list over all other agents. A specific set extension rule is then used to infer from each agent's preference list that agent's preferences over coalitions. In 3DR-B, that set extension rule is known as \mathcal{B} -preferences [19]. Using \mathcal{B} -preferences, any agent prefers some triple S to another triple T if the most-preferred agent in S is preferred to the most-preferred agent in T . In this chapter we consider in 3DR-B the existence of, and complexity of finding, matchings that are stable.

We first show, in Section 4.2, that a given instance of 3DR-B may not contain a stable matching and that the associated decision problem is NP-complete (Theorem 4.1). This contrasts with an analogous model in which coalitions need not have a fixed size, in which a stable matching must always exist and can be found in polynomial time [7]. It may seem intuitive that the additional requirement of fixed-size coalitions makes this particular problem harder to solve, and this result gives an example of a model in which that intuition holds. It also leads to interesting directions for subsequent work, for example involving approximately stable matchings, or alternative constraints on the size of feasible coalitions.

We then consider, in Section 4.3, a closely related optimisation problem in which the objective is to construct a matching that is, in terms of a specific measure, as stable as possible. We begin

by showing that an existing approximation algorithm for a different model of 3DR can be used to devise a $9/4$ -approximation algorithm for the 3DR-B problem (Theorem 4.2). We then show that a simple algorithm based on serial dictatorship gives an improved approximation ratio of $3/2$ (Theorem 4.3). Interestingly, this algorithm can be viewed as an adaptation of the algorithm developed by Cechlárová and Romero-Medina [7] that can be used to construct a stable matching in the analogous model in which coalitions need not have a fixed size.

Next, in Section 4.4, we consider the problem of identifying the smallest instance of 3DR-B that contains no stable matching. We show that such an instance contains at least 9 (Theorem 4.4) and at most 15 agents (Theorem 4.5) but leave determining the precise number of agents as an open problem.

Finally, in Section 4.5, we recap on our results and discuss some directions for future work.

We proceed with some formal definitions and notation. An instance of 3DR-B comprises a set N of $3n$ agents and a preference list of each agent α_i , labelled P_{α_i} , that describes a strict order over all agents in $N \setminus \{\alpha_i\}$. We say that an agent α_i *prefers* α_j to α_k , denoted $\alpha_j \succ_{\alpha_i} \alpha_k$, if α_j precedes α_k in P_{α_i} . A *triple* is an unordered set of three agents. In order to compare triples, agents in an instance of 3DR-B use \mathcal{B} -*preferences*, which are defined as follows. For any agent α_i and set of agents $S \subseteq N$ we denote by $\mathcal{B}_{\alpha_i}(S)$ the most-preferred agent in $S \setminus \{\alpha_i\}$ according to α_i . For any agent α_i and any two triples r and s , we say that α_i *prefers* r to s , denoted $r \succ_{\alpha_i} s$, if $\mathcal{B}_{\alpha_i}(r) \succ_{\alpha_i} \mathcal{B}_{\alpha_i}(s)$. A *matching* is a partition of N into n triples. Given an agent α_i and a matching M , we denote by $M(\alpha_i)$ the triple in M that contains α_i . Given a matching M , we say that a triple t is *blocking* if each agent α_i in t prefers t to $M(\alpha_i)$. A matching is *stable* if it does not contain a blocking triple. Let $\mathcal{B}_{\alpha_i}(M)$ be short for $\mathcal{B}_{\alpha_i}(M(\alpha_i))$. Let P be the collection of preference lists P_{α_i} for each agent α_i . For any instance (N, P) of 3DR-B and any matching M , we denote by $\text{bt}(M, (N, P)) \subseteq \binom{N}{3}$ the set of triples that block M in (N, P) . Conversely, we denote by $\text{nbt}(M, (N, P)) = \binom{N}{3} \setminus \text{bt}(M, (N, P))$ the set of triples that do not block M in (N, P) . When the instance in question is clear from context, we simply write $\text{bt}(M)$ or $\text{nbt}(M)$.

4.2 Deciding existence

In this section we show that deciding if a given instance of 3DR-B contains a stable matching is NP-complete. The reduction presented here is from the Three-Dimensional Stable Matching problem with Cyclic preferences (3-DSM-CYC, defined in Chapter 3), which is NP-complete [5].

Our reduction is similar to the reduction from 3GSM to the Three-Person Stable Assignment problem (3PSA, defined in Chapter 2) used by Ng and Hirschberg [10]. Our reduction, like

Ng and Hirschberg's, is from a problem involving a multipartite set of agents (in their case, 3GSM) to a problem involving a homogenous set of agents (in their case, 3PSA). Informally, the idea is to use the same set of agents in both problem instances but to design the preference lists of the agents in the constructed instance in such a way that each triple in any stable matching in the constructed instance must contain exactly one agent of each type. It is then straightforward to show that a stable matching exists in the instance of the latter problem if and only if a stable matching exists in the instance of the former problem. In fact, a similar technique was also used to reduce from (two-dimensional) SM to (two-dimensional) SR by Gusfield and Irving in 1989 [80, Lemma 4.1.1].

The reduction from 3-DSM-CYC to 3DR-B is as follows. Suppose (N', P') is an arbitrary instance of 3-DSM-CYC, in which the sets of agents of each type are labelled U , W , and D . We shall construct an instance (N, P) of 3DR-B. First construct three new 'sentinel' agents u_0, w_0, d_0 in N where:

$$\begin{aligned} P_{u_0} &: [W \text{ in arbitrary order }] w_0 [D \text{ in arbitrary order }] d_0 [U \text{ in arbitrary order }] \\ P_{w_0} &: [D \text{ in arbitrary order }] d_0 [U \text{ in arbitrary order }] u_0 [W \text{ in arbitrary order }] \\ P_{d_0} &: [U \text{ in arbitrary order }] u_0 [W \text{ in arbitrary order }] w_0 [D \text{ in arbitrary order }] \end{aligned}$$

Next, add each agent in $N' = U \cup W \cup D$ to N and for each i where $1 \leq i \leq n$ let:

$$\begin{aligned} P_{u_i} &: [P'_{u_i}] w_0 [D \text{ in arbitrary order }] d_0 [U \setminus \{u_i\} \text{ in arbitrary order }] u_0 \\ P_{w_i} &: [P'_{w_i}] d_0 [U \text{ in arbitrary order }] u_0 [W \setminus \{w_i\} \text{ in arbitrary order }] w_0 \\ P_{d_i} &: [P'_{d_i}] u_0 [W \text{ in arbitrary order }] w_0 [D \setminus \{d_i\} \text{ in arbitrary order }] d_0 \end{aligned}$$

This completes the construction of (N, P) . Partition N into three sets U', W', D' where $U' = U \cup \{u_0\}$, $W' = W \cup \{w_0\}$, and $D' = D \cup \{d_0\}$. Note that in the constructed instance (N, P) of 3DR-B:

- Any agent in U' prefers any agent in U' to any agent in W' and also prefers any agent in W' to any agent in D' (and similarly for any agent in W' or D').
- For any agent in N , u_0 is the least-preferred agent in U' , w_0 is the least-preferred agent in W' and d_0 is the least-preferred agent in D' .

It is straightforward to show that this reduction can be performed in polynomial time. To prove that the reduction is correct we show that the 3DR-B instance (N, P) contains a stable matching if and only if the 3-DSM-CYC instance (N', P') contains a stable matching.

We first show that if the 3-DSM-CYC instance (N', P') then the 3DR-B instance (N, P) contains a stable matching.

Lemma 4.1. *If (N', P') contains a stable matching then (N, P) contains a stable matching.*

Proof. Suppose M' is a stable matching in (N', P') . Let $M = \{u_0, w_0, d_0\} \cup \{(u_i, w_j, d_k) : (u_i, w_j, d_k) \in M'\}$.

Towards a contradiction, suppose M is not stable in (N, P) and that $\{\alpha_i, \alpha_j, \alpha_k\}$ blocks M in (N, P) . It must be that either $\alpha_i \in U'$, $\alpha_i \in W'$, or $\alpha_i \in D'$. Assume without loss of generality that $\alpha_i \in U'$. It follows that $\mathcal{B}_{\alpha_i}(M) \in W'$ and thus either $\alpha_j \in W'$ or $\alpha_k \in W'$. Suppose without loss of generality that $\alpha_j \in W'$. A similar argument then shows that $\alpha_k \in D'$ so we relabel $\{\alpha_i, \alpha_j, \alpha_k\}$ as $\{u_{i'}, w_{j'}, d_{k'}\}$ where $u_{i'} \in U'$, $w_{j'} \in W'$, and $d_{k'} \in D'$. Since $w_{j'} \succ_{u_{i'}} \mathcal{B}_{u_{i'}}(M)$ and, by the construction of M' , $\mathcal{B}_{u_{i'}}(M) \in W'$, it must be that $j' \neq 0$. A similar argument shows that $k' \neq 0$ and $i' \neq 0$. It then follows that the family $(u_{i'}, w_{j'}, d_{k'})$ blocks M in (N', P') , which is a contradiction. \square

We now show, using a sequence of lemmas, that if the 3DR-B instance (N, P) contains a stable matching then the 3-DSM-CYC instance (N', P') contains a stable matching. The complication here is that a triple in a stable matching in (N, P) need not contain exactly one agent in each of U' , W' , and D' . Nevertheless, we show that an arbitrary stable matching in (N, P) has a relatively constrained structure and can thus be modified such that each triple contains exactly one agent from each of U, W, D . It is then straightforward to construct a stable matching in the 3-DSM-CYC instance (N', P') .

We say that a triple of three agents in N is *mixed* if it does not contain exactly one agent in each of U' , W' , and D' . Without loss of generality assume that the number of mixed triples in M is minimal.

Suppose X is an arbitrary element of $\{U', W', D'\}$. Note that by definition, the number of agents in X in non-mixed triples in M is $n - |X|$. It follows that the number of agents in X in mixed triples in M is $|X|$ and thus the average number of agents in X in each mixed triple in M is $|X|/|X| = 1$.

Lemma 4.2. *If (N, P) contains a stable matching M then no triple in M contains three agents in exactly one of U' , W' , and D' .*

Proof. Assume without loss of generality that M contains some mixed triple $t_1 = \{u_{i_1}, u_{i_2}, u_{i_3}\}$ where $u_{i_1}, u_{i_2}, u_{i_3} \in U'$. Since t_1 contains more agents in U' than the average number of agents in U' in each mixed triple in M , it follows that there exists some mixed triple $t_2 \in M$ that contains fewer than the average number of agents in U' in each mixed triple in M . Since the average number of agents in U' in each mixed triple in M is 1 it must be that t_2 contains no agent in U' . Consider the possible contents of t_2 . If either $t_2 = \{w_{j_1}, d_{k_1}, d_{k_2}\}$ where $w_{j_1} \in W'$ and $d_{k_1}, d_{k_2} \in D'$, $t_2 = \{w_{j_1}, w_{j_2}, d_{k_1}\}$ where $w_{j_1}, w_{j_2} \in W'$ and $d_{k_1} \in D'$, or $t_2 = \{d_{k_1}, d_{k_2}, d_{k_3}\}$ where $d_{k_1}, d_{k_2}, d_{k_3} \in D'$, then $\{u_{i_1}, u_{i_2}, d_{k_1}\}$ blocks M' , which is a

contradiction. If $t_2 = \{w_{j_1}, w_{j_2}, w_{j_3}\}$ where $w_{j_1}, w_{j_2}, w_{j_3} \in W'$ then $\{w_{j_1}, w_{j_2}, u_{i_1}\}$ blocks M , which is a contradiction. \square

Lemma 4.3. *Arbitrarily label U', W', D' as X, Y, Z . If (N, P) contains a stable matching M that contains a mixed triple $\{x_{i_1}, x_{i_2}, y_{j_1}\}$ where $x_{i_1}, x_{i_2} \in X$ and $y_{j_1} \in Y$ then M also contains some mixed triple $\{y_{j_2}, z_{k_1}, z_{k_2}\}$ where $y_{j_2} \in Y$ and $z_{k_1}, z_{k_2} \in Z$.*

Proof. Without loss of generality assume that $X = U', Y = W'$, and $Z = D'$. Suppose for a contradiction that M contains some mixed triple $\{u_{i_1}, u_{i_2}, w_{j_1}\}$ where $u_{i_1}, u_{i_2} \in U'$ and $w_{j_1} \in W'$ and does not contain any triple $\{w_{j_2}, d_{k_1}, d_{k_2}\}$ where $w_{j_2} \in W'$ and $d_{k_1}, d_{k_2} \in D'$. Observe that $\{u_{i_1}, u_{i_2}, w_{j_1}\}$ contains more agents in U' than the average number of agents in U' in each mixed triple in M and fewer agents in D' than the average number of agents in D' in each mixed triple in M . It follows that there exists some mixed triple $t_1 \in M$ in which the number of agents in U' is 0 and some mixed triple $t_2 \in M$ in which the number of agents in D' is at least 2. By Lemma 4.2, the number of agents in t_2 in D' is exactly 2. By assumption, no triple $\{w_{j_2}, d_{k_1}, d_{k_2}\}$ exists in M so the only possibility is that $t_1 \neq t_2$ and $t_1 = \{w_{j_3}, w_{j_4}, d_{k_3}\}$ where $w_{j_3}, w_{j_4} \in W'$ and $d_{k_3} \in D'$ and $t_2 = \{d_{k_4}, d_{k_5}, u_{i_3}\}$ where $d_{k_4}, d_{k_5} \in D'$ and $u_{i_3} \in U'$. Now $\{u_{i_3}, w_{j_1}, d_{k_3}\}$ blocks M' , which is a contradiction. \square

Lemma 4.4. *If (N, P) contains a stable matching M then no triple in M is mixed.*

Proof. Assume for a contradiction that M contains at least one mixed triple. By Lemma 4.2, no triple contains three agents in any one of U', W' , and D' . Assume then without loss of generality that M contains some mixed triple $t_1 = \{u_{i_1}, u_{i_2}, w_{j_1}\}$ where $u_{i_1}, u_{i_2} \in U'$ and $w_{j_1} \in W'$. By Lemma 4.3 it follows that there exists some other mixed triple $\{w_{j_2}, d_{k_1}, d_{k_2}\}$ in M where $w_{j_2} \in W'$ and $d_{k_1}, d_{k_2} \in D'$.

We first claim that every mixed triple in M either contains two agents in U' and one agent in W' or contains two agents in D' and one agent in W' . If not, by Lemma 4.3 there are two possible cases: either M contains two triples $\{u_{i_3}, w_{j_3}, w_{j_4}\}, \{u_{i_4}, d_{k_3}, d_{k_4}\}$ or M contains two triples $\{d_{k_3}, w_{j_3}, w_{j_4}\}, \{u_{i_3}, u_{i_4}, d_{k_4}\}$, where in either case $u_{i_3}, u_{i_4} \in U'$, $w_{j_3}, w_{j_4} \in W'$, and $d_{k_3}, d_{k_4} \in D'$. In the former case, $\{u_{i_4}, w_{j_3}, d_{k_1}\}$ blocks M . In the latter case, $\{u_{i_4}, w_{j_1}, d_{k_3}\}$ blocks M .

Now consider w_0 . If $M(w_0)$ is not mixed then $M(w_0) = \{u_{i_3}, w_0, d_{k_3}\}$ where $u_{i_3} \in U'$ and $d_{k_3} \in D'$ and thus $\{u_{i_3}, w_{j_1}, d_{k_1}\}$ blocks M , which is a contradiction. It remains that $M(w_0)$ is mixed. Since every mixed triple either contains two agents in U' and one agent in W' or contains two agents in D' and one agent in W' , without loss of generality assume that either $j_1 = 0$ or $j_2 = 0$. Suppose firstly that $j_2 = 0$. It follows that $\{d_{k_1}, d_{k_2}, w_{j_1}\}$ blocks M' , which is a contradiction. It remains that $j_1 = 0$. To show a contradiction, we now construct a new matching \hat{M} in which the number of mixed triples in \hat{M} is strictly fewer than the number

of mixed triples in M . Consider $P_{w_{j_2}}$. It must be that either $d_{k_1} \succ_{w_{j_2}} d_{k_2}$ or $d_{k_2} \succ_{w_{j_2}} d_{k_1}$. Suppose without loss of generality that $d_{k_1} \succ_{w_{j_2}} d_{k_2}$. Now consider

$$\hat{M} = (M \setminus \{\{u_{i_1}, u_{i_2}, w_{j_1}\}, \{d_{k_1}, d_{k_2}, w_{j_2}\}\}) \cup \{\{u_{i_1}, w_0, d_{k_1}\}, \{u_{i_2}, w_{j_2}, d_{k_2}\}\}$$

in which:

- $\mathcal{B}_{u_{i_1}}(\hat{M}) = \mathcal{B}_{u_{i_1}}(M) = w_0$
- $\mathcal{B}_{w_0}(\hat{M}) = d_{k_1}$ and $\mathcal{B}_{w_0}(M) \in U'$ and so $\hat{M} \succ_{w_0} M$
- $\mathcal{B}_{d_{k_1}}(\hat{M}) = u_{i_1}$ and $\mathcal{B}_{d_{k_1}}(M) \in W'$ and so $\hat{M} \succ_{d_{k_1}} M$
- $\mathcal{B}_{u_{i_2}}(\hat{M}) = w_{j_2} \succ_{u_{i_2}} w_0 = \mathcal{B}_{u_{i_2}}(M)$
- $\mathcal{B}_{w_{j_2}}(\hat{M}) = \mathcal{B}_{w_{j_2}}(M) = d_{k_1}$
- $\mathcal{B}_{d_{k_2}}(\hat{M}) = \mathcal{B}_{d_{k_2}}(M) = w_{j_2}$.

It follows that any triple that blocks \hat{M} in (N, P) also blocks M in (N, P) . Thus, since M is stable, \hat{M} is also stable. The number of mixed triples in \hat{M} is exactly one fewer than in M , which contradicts our assumption that M is a stable matching in (N, P) with the minimal number of mixed triples. \square

Lemma 4.5. *If (N, P) contains a stable matching then (N, P) contains a stable matching \hat{M} such that \hat{M} contains $\{u_0, w_0, d_0\}$ and no triple in \hat{M} is mixed.*

Proof. Let S be the set of triples in M that each contain at least one agent in $\{u_0, w_0, d_0\}$. By definition, $1 \leq |S| \leq 3$. If $|S| = 1$, then $\{u_0, w_0, d_0\} \in M$ so $\hat{M} = M$. If $|S| = 3$ then by Lemma 4.4 it must be that $S = \{\{u_0, w_{j_1}, d_{k_1}\}, \{u_{i_2}, w_0, d_{k_2}\}, \{u_{i_3}, w_{j_3}, d_0\}\}$, where $u_{i_2}, u_{i_3} \in U'$, $w_{j_1}, w_{j_2} \in W'$ and $d_{k_1}, d_{k_2} \in D'$. Now $\{u_{i_2}, w_{j_3}, d_{k_1}\}$ blocks M , which is a contradiction. It remains that $|S| = 2$. By Lemma 4.4, no triple in M is mixed, so there are three possible cases: either $S = \{\{u_0, w_0, d_k\}, \{u_i, w_j, d_0\}\}$, $S = \{\{u_0, w_j, d_0\}, \{u_i, w_0, d_k\}\}$, or $S = \{\{u_0, w_j, d_k\}, \{u_i, w_0, d_0\}\}$, where in any case $u_i \in U'$, $w_j \in W'$, and $d_j \in D'$. In any case, let:

$$\hat{M} = (M \setminus S) \cup \{\{u_i, w_j, d_k\}, \{u_0, w_0, d_0\}\}.$$

Now:

- either $\mathcal{B}_{u_i}(\hat{M}) \succ_{u_i} \mathcal{B}_{u_i}(M)$ or $\mathcal{B}_{u_i}(\hat{M}) = \mathcal{B}_{u_i}(M)$
- either $\mathcal{B}_{w_j}(\hat{M}) \succ_{w_j} \mathcal{B}_{w_j}(M)$ or $\mathcal{B}_{w_j}(\hat{M}) = \mathcal{B}_{w_j}(M)$

- either $\mathcal{B}_{d_k}(\hat{M}) >_{d_k} \mathcal{B}_{d_k}(M)$ or $\mathcal{B}_{d_k}(\hat{M}) = \mathcal{B}_{d_k}(M)$.

It follows that any triple that blocks \hat{M} in (N, P) also blocks M in (N, P) . Thus, since M is stable, \hat{M} is also stable. \square

Lemma 4.6. *If (N, P) contains a stable matching then (N', P') contains a stable matching.*

Proof. By Lemma 4.5, there exists a stable matching \hat{M} that contains $\{u_0, w_0, d_0\}$ in which no triple is mixed. We claim that

$$M' = \{(u_i, w_j, d_k) : \{u_i, w_j, d_k\} \in \hat{M}\}$$

is a stable matching in (N', P') . Suppose for a contradiction that the family (u_i, w_j, d_k) blocks M' in (N', P') . It follows that the triple $\{u_i, w_j, d_k\}$ blocks \hat{M} in (N, P) , which is a contradiction. \square

We have now shown that the 3DR-B instance (N, P) contains a stable matching if and only if the 3-DSM-CYC instance (N', P') contains a stable matching. This shows that the reduction is correct.

Theorem 4.1. *Deciding if a given instance of 3DR-B contains a stable matching is NP-complete.*

Proof. This decision problem belongs to NP since the stability of a given matching M can be verified in polynomial time, as follows. For each triple $r \in \binom{N}{3}$ consider each agent α_i in r compare $\mathcal{B}_{\alpha_i}(r)$ and $\mathcal{B}_{\alpha_i}(M)$ using P_{α_i} . If at least one agent α_i in each triple r does not prefer $\mathcal{B}_{\alpha_i}(r)$ to $\mathcal{B}_{\alpha_i}(M)$ then M is stable.

We have presented a polynomial-time reduction from 3-DSM-CYC, which is NP-complete [5]. Given an arbitrary instance (N', P') of 3-DSM-CYC, the reduction constructs an instance (N, P) of 3DR-B. By use the reduction described to construct (N, P) . In Lemmas 4.1 and 4.6 we showed that (N, P) contains a stable matching if and only if (N', P') contains a stable matching and thus that this decision problem is NP-hard. \square

4.3 Approximation

The *3DR-B Maximally Stable Matching problem* (3DR-B-MSM) is the following optimisation problem: given an instance of 3DR-B, find a matching with the maximum number of non-blocking triples. Formally, 3DR-B-MSM is a maximisation problem in which any instance (N, P) of 3DR-B-MSM is also instance of 3DR-B, a solution is a matching in (N, P) , and

the measure is $|\text{nbt}(M, (N, P))|$. We showed in Theorem 4.1 that deciding if an instance of 3DR-B contains a matching with $\binom{3n}{3}$ non-blocking triples is NP-complete, so it follows that 3DR-B-MSM is NP-hard. In this section we present two approximation algorithms for 3DR-B-MSM. The first is a direct application of an existing result relating to 3PSA. The second is a novel serial dictatorship-style algorithm.

There is a close relationship between 3DR-B and the Three-Person Stable Assignment problem (3PSA, introduced in Chapter 2). Recall that in 3DR-B, each agent has a strict preference list over the other $3n - 1$ agents. \mathcal{B} -preferences are then used to infer each agent's preferences over triples. In 3PSA, each agent instead has a strict preference list over the $\binom{3n-1}{2}$ triples that they may belong to. We show that, given an instance (N, P) of 3DR-B, it is possible to construct an instance (N, P') of 3PSA with the same set of agents such that for any matching M , if a triple blocks M in (N, P) then it also blocks M in (N, P') . From this result it follows that an existing 9/4-approximation algorithm for 3PSA-MSM [75], which is defined analogously to 3DR-B-MSM, can be applied to construct a 9/4-approximation algorithm for 3DR-B-MSM. The existing 9/4-approximation algorithm for 3PSA-MSM was presented by Rosenbaum in 2016 [75] and is called Algorithm ASA. It is an iterative greedy algorithm which involves, in each iteration, selecting a triple that once added to the matching, intersects the maximum number of non-blocking triples.

The design of our 9/4-approximation algorithm for 3DR-B-MSM, which makes use of the relationship between 3PSA and 3DR-B, is essentially the same as the 9/4-approximation algorithm that we described in Chapter 3 for 3DSM-CYC-MSM. As we saw in Chapter 3, the latter algorithm makes use of the analogous relationship between 3GSM and 3DSM-CYC (in fact, Algorithm AMSM is also essentially the same as Algorithm ASA [75]).

Theorem 4.2. *There exists a polynomial-time 9/4-approximation algorithm for 3DR-B-MSM.*

Proof. The approximation algorithm for 3DR-B-MSM involves constructing a corresponding instance of 3PSA-MSM, which has the same set of agents, and running Algorithm ASA [75].

We first describe how to construct, in polynomial time, a corresponding instance (N, P') of 3PSA such that $|\text{bt}(M, (N, P))| \geq |\text{bt}(M, (N, P'))|$ for any matching M . For each agent $\alpha_i \in N$, let P'_{α_i} be the list of all 2-agent subsets of $N \setminus \{\alpha_i\}$ in lexicographic order with respect to P_{α_i} . Now, suppose M is an arbitrary matching in (N, P) and r is a triple that blocks M in (N, P) . We will show that r also blocks M in (N, P') . For any α_k in r it must be that $\mathcal{B}_{\alpha_k}(r)$ precedes $\mathcal{B}_{\alpha_k}(M)$ in P_{α_k} . By the construction of P'_{α_k} as the lexicographic order of P_{α_k} , it must be that r precedes $M(\alpha_k)$ in P'_{α_k} and thus that r also blocks M in (N, P') . It follows that $|\text{bt}(M, (N, P'))| \geq |\text{bt}(M, (N, P))|$ and thus that $|\text{nbt}(M, (N, P))| \geq |\text{nbt}(M, (N, P'))|$, as required.

Since Rosenbaum's [75] analysis of Algorithm ASA shows that $|\text{nbt}(M, (N, P'))| \geq 4\binom{3n}{3}/9$, we can conclude that $|\text{nbt}(M, (N, P))| \geq 4\binom{3n}{3}/9$. \square

We now present an algorithm with an improved approximation ratio. This algorithm, called Algorithm serialDictatorship, is based on serial dictatorship. It is a variation of Algorithm cyclicSerialDictatorship that we presented for 3-DSM-CYC in Chapter 3. The accompanying analysis is loosely based on the analysis of Algorithm ASA given by Rosenbaum [75].

Algorithm 4.1 Algorithm serialDictatorship

Input: an instance (N, P) of 3DR-B

Output: a matching M in (N, P)

```

 $U \leftarrow N$ 
 $M \leftarrow \emptyset$ 
while  $|U| > 0$  do
   $d_1 \leftarrow$  an arbitrary agent in  $U$ 
   $d_2 \leftarrow \mathcal{B}_{d_1}(U)$ 
   $d_3 \leftarrow \mathcal{B}_{d_2}(U \setminus \{d_1\})$ 
   $M \leftarrow M \cup \{\{d_1, d_2, d_3\}\}$ 
   $U \leftarrow U \setminus \{d_1, d_2, d_3\}$ 
end while
return  $M$ 

```

It is straightforward to show that Algorithm serialDictatorship returns a matching M in polynomial time. We now analyse its approximation ratio in the same way we analysed the approximation ratio of Algorithm cyclicSerialDictatorship for Theorem 3.2 in Chapter 3, and show that our analysis is tight. We consider each triple in M in the order that they were added to M in the algorithm and count only the blocking triples that intersect that triple and do not intersect any previous triple.

Theorem 4.3. *There exists a polynomial-time $3/2$ -approximation algorithm for 3DR-B-MSM.*

Proof. Namely Algorithm serialDictatorship. By the pseudocode, there are n iterations of the while loop so it is straightforward to show that the algorithm runs in polynomial time. For each i where $1 \leq i \leq n$, let d_1^i, d_2^i, d_3^i be the agents labelled d_1, d_2, d_3 respectively in that iteration, $D_i = \{d_1^i, d_2^i, d_3^i\}$, and U_i be the set of agents in U at the start of that iteration. Note that by the algorithm, $|U_i \setminus D_i| = 3n - 3i$.

For each i where $1 \leq i \leq n$, let S_i be the set of triples that block M , have a non-empty intersection with D_i , and have an empty intersection with D_j for every $j < i$. It follows that $S_n = \emptyset$ since any triple that blocks M and intersects the final triple D_n must contain some

agent not in D_n , which must belong to some previous triple D_j where $1 \leq j < n$. We can now define $\text{bt}(M)$ in terms of S_i :

$$\text{bt}(M) = \bigcup_{i=1}^{n-1} S_i .$$

By definition, the sets S_i are pairwise disjoint so it follows that

$$|\text{bt}(M)| = \sum_{i=1}^{n-1} |S_i| . \quad (4.1)$$

We now place an upper bound on $|S_i|$ for any i where $1 \leq i \leq n-1$. For any such i , consider the i^{th} iteration of the while loop. By the algorithm, $\mathcal{B}_{d_1^i}(U_i) = d_2^i$. It follows that any triple that blocks M and contains d_1^i contains some agent in $N \setminus (U_i \cup D_i)$ that has already been added to some triple D_j in M where $j < i$. Thus, no triple in S_i contains d_1^i . Similarly, by the algorithm it must be that either $\mathcal{B}_{d_2^i}(U_i) = d_3^i$ or $\mathcal{B}_{d_2^i}(U_i) = d_1^i$, so any triple that blocks M and contains d_2^i contains some agent in $N \setminus (U_i \cup D_i)$, so likewise no triple in S_i contains d_2^i . It follows that any triple in S_i contains d_3^i and two other agents in N . By the definition of S_i , any triple in S_i has an empty intersection with D_j for any $j < i$ so it must be that any triple in S_i contains d_3^i as well as two agents in $U_i \setminus D_i$. Since $|U_i \setminus D_i| = 3n - 3i$ it follows that

$$|S_i| \leq \binom{3n - 3i}{2} . \quad (4.2)$$

By definition,

$$\begin{aligned} |\text{nbt}(M)| &= \binom{3n}{3} - |\text{bt}(M)| \\ &= \binom{3n}{3} - \sum_{i=1}^{n-1} |S_i| && \text{by Equation 4.1} \\ &\geq \binom{3n}{3} - \sum_{i=1}^{n-1} \binom{3n - 3i}{2} && \text{by Inequality 4.2} \\ &= 3n^3 - \frac{3n^2}{2} - \frac{n}{2} . \end{aligned} \quad (4.3)$$

Suppose M^* is a matching in (N, P) with the maximum number of non-blocking triples. The

approximation ratio of the algorithm is thus

$$\begin{aligned}
 \frac{|\text{nbt}(M^*)|}{|\text{nbt}(M)|} &\leq \binom{3n}{3} \frac{1}{|\text{nbt}(M)|} && \text{since } |\text{nbt}(M^*)| \leq \binom{3n}{3} \\
 &\leq \frac{9n^2 - 9n + 2}{6n^2 - 3n - 1} && \text{by Inequality 4.3} \\
 &\leq \frac{3}{2} && \text{since } n \geq 1.
 \end{aligned}$$

□

It is desirable to show that this analysis is tight, by constructing an instance (N, P) of 3DR-B and showing that there exists some execution of Algorithm serialDictatorship that returns a matching M for which $|\text{nbt}(M^*)|/|\text{nbt}(M)| = 3/2$, where M^* is some matching in (N, P) with the maximum number of non-blocking triples. We show that the analysis is tight asymptotically, by constructing an instance \mathcal{I}_n of 3DR-B for some fixed $n \geq 1$, where the approximation ratio obtained by Algorithm serialDictatorship on \mathcal{I}_n is $3/2 - o(1)$ in the worst case. The proof follows the same pattern as the analysis of Algorithm cyclicSerialDictatorship in Chapter 3.

The structure of the preferences of the agents in \mathcal{I}_n corresponds directly to the counting argument used in the proof of Theorem 4.3. For any fixed n , construct \mathcal{I}_n as follows. First, let $N = \{\alpha_1, \alpha_2, \dots, \alpha_{3n}\}$. Next, for each agent $\alpha_j \in N$, construct P_{α_j} so that it lists every agent in $N \setminus \{\alpha_j\}$ in ascending order of subscript. Finally, for each i where $1 \leq i \leq n$, modify $P_{\alpha_{3i}}$ by shifting α_{3i-2} and α_{3i-1} to the right so that α_{3i-2} is the second-to-last agent in P_{α_i} and α_{3i-1} is the last agent in P_{α_i} .

As in the proof of Theorem 4.3, for each i where $1 \leq i \leq n$, let d_1^i, d_2^i, d_3^i be the agents labelled d_1, d_2, d_3 respectively in that iteration, $D_i = \{d_1^i, d_2^i, d_3^i\}$, and U_i be the set of agents in U at the start of that iteration. Since the selection of agents in U is arbitrary, suppose $d_1^i = \alpha_1$. It follows that $d_2^i = \alpha_2$ and $d_3^i = \alpha_3$ and thus $U_2 = N \setminus \{\alpha_1, \alpha_2, \alpha_3\}$. Similarly, the second triple chosen is $D_2 = \{\alpha_4, \alpha_5, \alpha_6\}$. In general, it follows that $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}, \dots, \{\alpha_{3n-2}, \alpha_{3n-1}, \alpha_{3n}\}\}$ and $U_i = \bigcup_{j=i}^n \{\alpha_{3j-2}, \alpha_{3j-1}, \alpha_{3j}\}$. Note that now, for any $\alpha_k \in N$, α_k prefers to $\mathcal{B}_{\alpha_k}(M)$ any agent α_{3i} where $3i < k$.

As in the proof of Theorem 4.3, for each i where $1 \leq i \leq n$, let S_i be the set of triples that block M , have a non-empty intersection with D_i , and have an empty intersection with D_j for every $j < i$. By definition it follows that $S_i \subseteq \binom{U_i}{3}$ and $S_n = \emptyset$. As in the proof of Theorem 4.3, it can be shown that for any i where $1 \leq i \leq n$, no triple in S_i contains α_{3i-2} and no triple in S_i contains α_{3i-1} . It remains that each triple in S_i contains α_{3i} as well as two agents in $U_i \setminus D_i$, which we label α_k and α_l . Since $U_i \setminus D_i = \bigcup_{j=i+1}^n \{\alpha_{3j-2}, \alpha_{3j-1}, \alpha_{3j}\}$, it must be that $k \geq 3i + 1$ and $l \geq 3i + 1$. Since $k \neq l$ assume without loss of generality that

$k \geq 3i + 2$. It follows that $\mathcal{B}_{\alpha_{3i}}(\{\alpha_k, \alpha_l\}) = \alpha_k$ precedes both α_{3i-1} and α_{3i-2} in $P_{\alpha_{3i}}$. As we noted, since $3i < k$ it must be that α_{3i} precedes $\mathcal{B}_{\alpha_k}(M)$ in P_{α_k} . Similarly, since $3i < l$ it must be that α_{3i} precedes $\mathcal{B}_{\alpha_l}(M)$ in P_{α_l} . It follows that the triple $\{\alpha_{3i}, \alpha_k, \alpha_l\}$ blocks M and hence also belongs to S_i . Since the selection of α_k, α_l as two agents in $U_i \setminus D_i$ was arbitrary it follows that

$$S_i = \{\{\alpha_{3i}, \alpha_k, \alpha_l\} : \alpha_k, \alpha_l \in U_i \setminus D_i\} .$$

It follows immediately that

$$|S_i| = \binom{3n - 3i}{2}$$

which shows that the upper bound on $|S_i|$ in Inequality 4.2 in Theorem 4.3 is tight. The same argument used in the proof of Theorem 4.3 then shows that

$$|\text{nbt}(M)| = 3n^3 - \frac{3n^2}{2} - \frac{n}{2} . \quad (4.4)$$

We now show that a stable matching exists in \mathcal{I}_n . Let

$$M^* = \{\alpha_1, \alpha_2, \alpha_{3n}\} \cup \bigcup_{i=1}^{n-1} \{\{\alpha_{3i}, \alpha_{3i+1}, \alpha_{3i+2}\}\} .$$

Note first that by the construction of M^* , $\mathcal{B}_{\alpha_2}(M^*) = \mathcal{B}_{\alpha_{3n}}(M^*) = \alpha_1$, $\mathcal{B}_{\alpha_1}(M^*) = \alpha_2$. Note also that for any i where $1 \leq i \leq n - 1$, $\mathcal{B}_{\alpha_{3i}}(M^*) = \alpha_{3i+1}$, $\mathcal{B}_{\alpha_{3i+1}}(M^*) = \alpha_{3i+2}$, and $\mathcal{B}_{\alpha_{3i+2}}(M^*) = \alpha_{3i+1}$. It follows in general, by the construction of \mathcal{I}_n , that for any two agents α_j and α_k if α_k precedes $\mathcal{B}_{\alpha_j}(M^*)$ in P_{α_j} then $k < j$. Consider an arbitrary triple $\{\alpha_i, \alpha_j, \alpha_k\} \in \binom{N}{3}$ where $i < j < k$. It follows that $\mathcal{B}_{\alpha_i}(\{\alpha_j, \alpha_k\})$ succeeds $\mathcal{B}_{\alpha_i}(M)$ in P_{α_i} and thus that this triple does not block M^* . It follows that M^* is stable. Now

$$\begin{aligned} \frac{|\text{nbt}(M^*)|}{|\text{nbt}(M)|} &= \binom{3n}{3} \frac{1}{|\text{nbt}(M)|} && \text{since } M^* \text{ is stable} \\ &= \frac{9n^2 - 9n + 2}{6n^2 - 3n - 1} && \text{by Equation 4.4} \\ &= \frac{3}{2} - o(1) \end{aligned}$$

which shows that the analysis is tight asymptotically.

4.4 Towards a minimal “no” instance

As in the case of 3-DSM-CYC, it seems difficult to characterise instances of 3DR-B that do not contain a stable matching. With respect to the respective existence problems, which are decision problems, we call such instances “no” instances. A significant open question for both 3-DSM-CYC and 3DR-B involves the minimal number of agents required to construct such an instance. The smallest such instance of 3-DSM-CYC uses 60 agents ($n = 20$) [54]. In this section, we show that the smallest such instance of 3DR-B contains at least 9 and at most 15 agents, but leave determining the precise number as an open problem.

We first show, in Theorem 4.4, that any instance of 3DR-B with at most 6 agents must contain a stable matching. It follows that at least 9 agents are required to construct an instance of 3DR-B that contains no stable matching.

Theorem 4.4. *In any instance (N, P) of 3DR-B, if $|N| \leq 6$ then (N, P) contains a stable matching.*

Proof. If $|N| = 3$ then any matching is stable so suppose $|N| = 6$. Consider the directed graph $G = (N, A)$ where N is the set of agents and $(\alpha_i, \alpha_j) \in A$ if $\mathcal{B}_{\alpha_i}(N) = \alpha_j$. We shall analyse the structure of G in a case analysis, and in each case identify a stable matching M . Since the out-degree of each agent in G is 1, and by definition G contains no self-loops, it must be that each weakly connected component of G contains at least two agents and at least one directed cycle. If G contains:

- Three components $\{\alpha_1, \alpha_2\}, \{\alpha_3, \alpha_4\}, \{\alpha_5, \alpha_6\}$, each of size two, then $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 2, 4, 5\}$.
- Two components $\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}$ each of size three, then $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each $\alpha_i \in N$.
- Two components $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, \{\alpha_5, \alpha_6\}$ then there are two possible cases. In the first, the longest directed cycle among $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ contains two agents. In this case, assume without loss of generality that this cycle is $\{\alpha_1, \alpha_2\}$ and that $(\alpha_3, \alpha_1) \in A$. Now $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 2, 5, 6\}$. In the second, the longest directed cycle among $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ contains three or more agents. In this case, we may assume without loss of generality that $(\alpha_1, \alpha_2) \in A$ and $(\alpha_2, \alpha_3) \in A$. Now $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 2, 5, 6\}$.
- One component $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$. If the longest directed cycle contains:

- Six agents then $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 2, 4, 5\}$.
- Five agents then assume without loss of generality that this cycle is $\{\alpha_1, \alpha_2, \dots, \alpha_5\}$. Now $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 2, 4, 5\}$.
- Four agents then assume without loss of generality that this cycle is $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ and that $(\alpha_5, \alpha_1) \in A$. Consider $\mathcal{B}_{\alpha_6}(N)$. If $(\alpha_6, \alpha_1) \in A$ then $M = \{\{\alpha_1, \alpha_5, \alpha_6\}, \{\alpha_2, \alpha_3, \alpha_4\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{2, 3, 5, 6\}$. If either $(\alpha_6, \alpha_2) \in A$ or $(\alpha_6, \alpha_3) \in A$ then $M = \{\{\alpha_1, \alpha_4, \alpha_5\}, \{\alpha_2, \alpha_3, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{2, 4, 5, 6\}$. If $(\alpha_6, \alpha_4) \in A$ then $M = \{\{\alpha_1, \alpha_2, \alpha_5\}, \{\alpha_3, \alpha_4, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 3, 5, 6\}$.
- Three agents then assume without loss of generality that this cycle is $\{\alpha_1, \alpha_2, \alpha_3\}$. Now $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 2, 3\}$ and $\{\alpha_1, \alpha_2, \alpha_3\} \in M$.
- Two agents then assume without loss of generality that this cycle is $\{\alpha_1, \alpha_2\}$ and also that $(\alpha_3, \alpha_1) \in A$. Now $M = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_6\}\}$ is stable since $\mathcal{B}_{\alpha_i}(M) = \mathcal{B}_{\alpha_i}(N)$ for each i where $i \in \{1, 2, 3\}$ and $\{\alpha_1, \alpha_2, \alpha_3\} \in M$.

□

We now show, in Theorem 4.5, that there exists an instance of 3DR-B with 15 agents that contains no stable matching. That instance, which we denote by (N', P') , is illustrated in Figure 4.1 and can be constructed as follows. First, for each r where $1 \leq r \leq 3$ construct a set of five agents in N' labelled $Q_r = \{q_r^1, q_r^2, \dots, q_r^5\}$, which we refer to as a *pentagadget*. To simplify the description of the valuations in each pentagadget, in what follows we write $i \oplus y$ to denote $((i + y - 1) \bmod 5) + 1$. For each r where $1 \leq r \leq 3$ and each i where $1 \leq i \leq 5$ let

$$P_{q_r^i} : q_r^{i \oplus 1} q_r^{i \oplus 2} q_r^{i \oplus 3} q_r^{i \oplus 4} \text{ [} N' \setminus Q_r \text{ in arbitrary order]}$$

Suppose M is an arbitrary matching in (N', P') . We shall eventually show that M is not stable. For any agent $q_r^i \in N'$, we say that q_r^i is *external* if $\mathcal{B}_{q_r^i}(M) \notin Q_r$.

Lemma 4.7. *At least one pentagadget contains at least two external agents.*

Proof. Since each pentagadget contains five agents it must be that Q_1 contains at least one external agent. Without loss of generality assume that q_1^1 is external. If Q_1 contains two external agents, including q_1^1 , then the lemma statement holds, so suppose q_1^1 is the

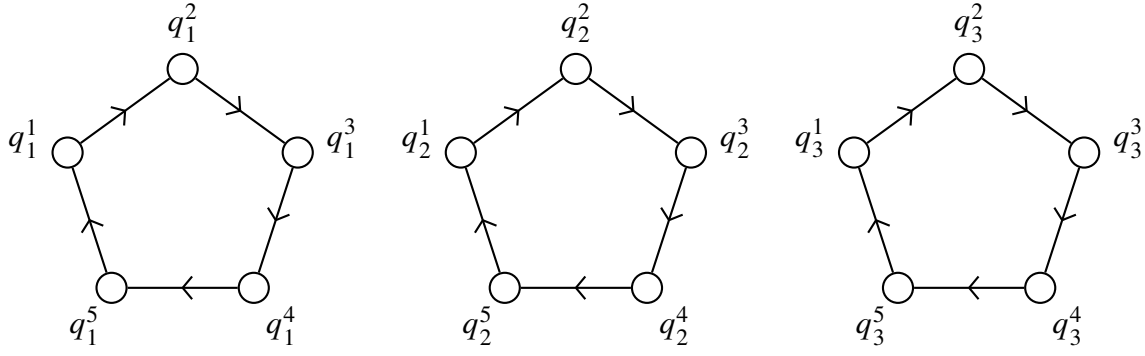


Figure 4.1: A representation of the instance (N', P') of 3DR-B that contains no stable matching. An arrow exists from q_r^i to q_r^j if $\mathcal{B}_{q_r^i}(N') = q_r^j$.

only external agent in Q_1 . It follows that $\{q_1^{i_1}, q_1^{i_2}, q_r^{j_1}\}, \{q_1^{i_3}, q_1^{i_4}, q_s^{k_1}\} \in M$ where $\{i_1, i_2, i_3, i_4\} = \{2, 3, 4, 5\}$, $r, s \in \{2, 3\}$, and $1 \leq j_1, k_1 \leq 5$. If $r = s$ then Q_r contains two external agents, namely $q_r^{j_1}$ and $q_r^{k_1}$, and thus the lemma statement holds, so suppose $r \neq s$. Assume without loss of generality that $q_r^{j_1} = q_2^1$ and $q_s^{k_1} = q_3^1$.

By definition, q_2^1 is external. If Q_2 contains two external agents, including q_2^1 , then the lemma statement holds, so suppose q_2^1 is the only external agent in Q_2 . It follows that $\{q_2^{l_2}, q_2^{l_3}, q_t^{j_2}\}, \{q_2^{l_4}, q_2^{l_5}, q_u^{k_2}\} \in M$, where $\{l_2, l_3, l_4, l_5\} = \{2, 3, 4, 5\}$, $t, u \in \{1, 3\}$, and $1 \leq j_2, k_2 \leq 5$. If $t = u$ then Q_t contains two external agents, namely $q_t^{j_2}$ and $q_t^{k_2}$, and thus the lemma statement holds, so it must be that $t \neq u$. Assume without loss of generality that $t = 1$ and $u = 3$. Now Q_3 contains two external agents, namely q_3^1 and $q_3^{k_2}$, and thus the lemma statement holds. \square

Theorem 4.5. *There exists an instance of 3DR-B with 15 agents that contains no stable matching.*

Proof. Namely the instance (N', P') . Recall that M is an arbitrary matching in (N', P') . By Lemma 4.7, assume without loss of generality that Q_1 contains at least two external agents, which we label $q_1^{i_1}$ and $q_1^{i_2}$. By the symmetry of each pentagadget, without loss of generality assume that $i_1 = 1$ and either $i_2 = 2$ or $i_2 = 3$. In the former case, $\{q_1^1, q_1^2, q_1^5\}$ blocks M , since $q_1^2 \succ_{q_1^1} \mathcal{B}_{q_1^1}(M)$, $\mathcal{B}_{q_1^2}(\{q_1^1, q_1^5\}) \succ_{q_1^2} \mathcal{B}_{q_1^2}(M)$, and $q_1^1 \succ_{q_1^5} \mathcal{B}_{q_1^5}(M)$. In the latter case, $\{q_1^1, q_1^2, q_1^3\}$ blocks M , since $q_1^2 \succ_{q_1^1} \mathcal{B}_{q_1^1}(M)$, $q_1^3 \succ_{q_1^2} \mathcal{B}_{q_1^2}(M)$, and $\mathcal{B}_{q_1^3}(\{q_1^1, q_1^2\}) \succ_{q_1^3} \mathcal{B}_{q_1^3}(M)$. \square

4.5 Summary and open problems

In this chapter we studied a new formalism of 3DR, involving \mathcal{B} -preferences, which we called 3DR-B. We considered in 3DR-B the existence of, and complexity of finding, matchings that are stable.

Our first result was that a given instance of 3DR-B may not contain a stable matching and that the associated decision problem is NP-complete. We then considered a closely related optimisation problem, which we called 3DR-B-MSM, in which the objective is to construct, in a given instance of 3DR-B, a matching with the maximum number of non-blocking triples. We first devised a $9/4$ -approximation algorithm for 3DR-B-MSM based on an existing algorithm for 3PSA-MSM, which is a closely related problem [75]. Improving upon this approximation, we then presented a $3/2$ -approximation algorithm based on serial dictatorship, and showed that our analysis is tight asymptotically. Finally, we considered the problem of identifying the smallest instance of 3DR-B that contains no stable matching. We showed that such an instance must have between 9 and 15 agents, inclusive.

We now present some open problems specifically involving stability in 3DR-B. More general problems, involving solution concepts other than stability and other models of fixed-size coalition formation, are discussed in Chapter 9.

An immediate open problem is to improve the bounds on the the smallest instance of 3DR-B that contains no stable matching. We have shown in Section 4.4 that such an instance contains at least 9 and at most 15 agents, but the precise number of agents remains open. To fully resolve this open question, it will be necessary to either prove that every some fixed size strictly greater than 6 contains a stable matching (as in the proof of Theorem 4.4), demonstrate that some instance with between 9 and 12 agents contains no stable matching (in a similar way to the proof of Theorem 4.5), or both.

In Chapter 3, we presented an approximation algorithm for a restriction of 3-DSM-CYC-MSM in which the preferences of some agents were derived from a master list. We conjecture that a similar algorithm also exists for a restriction of 3DR-B-MSM in which the preferences of all agents are derived from a master list. In 2020, Brederick et al. [68] considered a similar situation for a multidimensional generalisation of 3GSM [10], and it may be that some of their results or techniques can also be applied to 3DR-B.

A closely related objective is to estimate the probability that a random instance of 3DR-B contains a stable matching. Pittel’s [53] probabilistic analysis of k -DSM-CYC and Pittel and Irving’s [81] analysis of two-dimensional Stable Roommates (SR) are two possible starting points. A hybrid of theoretical and empirical techniques might also be informative, as it was in Escamocher and O’Sullivan’s [51] paper, which considered the same question in the setting to 3-DSM-CYC.

As well as investigating the existence of an improved approximation algorithm for 3DR-B-MSM, it might be possible to prove an inapproximability result for this problem. For example, it would be very informative to prove that the approximation ratio of Algorithm `serialDictatorship` is tight, by showing that no $(3/2 - \varepsilon)$ -approximation algorithm exists for 3DR-B-MSM unless $P = NP$. Alternatively, it might be easier to prove the weaker result

that 3DR-B-MSM is APX-hard, meaning that there exists some constant factor ε such that no $(1 + \varepsilon)$ -approximation algorithm exists for 3DR-B-MSM, unless $P = NP$ [66]. A starting point towards the latter result could be to modify the reduction of Iwama et al. [61] from a variant of *Maximum 3D Matching* (Max 3DM) to an optimisation problem defined in a related model of 3DR (which is described in Chapter 2). Another possibility is to adapt the reduction of Rosenbaum [75] from Max 3DM to 3PSA-MSM.

Various alternative optimisation problems and measures can also be defined that relate to stable matchings and 3DR-B. One possibility is to construct a sub-matching of maximum cardinality such that no three agents in triples in the sub-matching form a blocking triple in the sub-matching. Rosenbaum [75] refers to the analogous problem for 3PSA as the *3PSA Maximum Stable Sub-matching problem* (3PSA-MSS). A second possibility is to consider α -stability [62] (discussed in Chapter 2) in the setting of 3DR-B. For example, for some fixed $\alpha \geq 1$, we could say that a matching M is α -stable if for any agent α_i and any triple t where $\alpha_i \in t$ the increase in rank in P_{α_i} from $\mathcal{B}(M)$ to $\mathcal{B}(t)$ is at most α . We could also then define an optimisation problem in which the objective is to find an α -stable matching for a minimum such α . A third possibility is to define a complementary problem in which the objective is to minimise the number of blocking triples, which is arguably more natural. Similar optimisation problems, in which the objective is to minimise the number of blocking pairs, have been studied in the context of (two-dimensional) Stable Roommates [78].

Chapter 5

Three-Dimensional Stable Roommates with \mathcal{W} -preferences

5.1 Introduction

In this chapter we study another new model of fixed-size coalition formation, which we call the *Three-Dimensional Roommates with \mathcal{W} -preferences* (3DR-W). This model is in a sense dual to the *Three-Dimensional Roommates with \mathcal{B} -preferences* (3DR-B), which we considered in Chapter 4. As in 3DR-B, in 3DR-W each agent supplies a strict preference list over all other agents and uses a set extension rule to compare coalitions. In 3DR-W, the set extension rule used is known as \mathcal{W} -preferences [8] and means that any agent prefers some triple S to another triple T if the least-preferred agent in S is more preferred than the least-preferred agent in T . In this chapter we consider in 3DR-W the existence of, and complexity of finding, matchings that are stable.

We first show, in Section 5.2, that a given instance of 3DR-W may not contain a stable matching and that the associated decision problem is NP-complete (Theorem 5.1). As in the case of 3DR-B, this also contrasts with an analogous model involving \mathcal{W} -preferences in which coalitions need not have a fixed size, in which a stable matching must exist and can be found in polynomial time [8]. It may seem intuitive that the additional requirement of fixed-size coalitions makes this particular problem harder to solve, and this result gives another example of a model in which that intuition holds.

We then consider, in Section 5.3, a closely related optimisation problem in which the objective is to construct a matching that is, in terms of a specific measure, as stable as possible. We show that an existing approximation algorithm for a different model of 3DR can be used to devise a $9/4$ -approximation algorithm for this problem (Theorem 5.2).

Finally, in Section 5.4, we recap on our results and discuss some directions for future work.

An instance of 3DR-W comprises a set N of $3n$ agents and a preference list of each agent α_i , labelled P_{α_i} , that describes a strict order over all agents in $N \setminus \{\alpha_i\}$. We say that an agent α_i *prefers* α_j to α_k , denoted $\alpha_j \succ_{\alpha_i} \alpha_k$, if α_j precedes α_k in P_{α_i} . A *triple* is an unordered set of three agents. In order to compare triples, agents in an instance of 3DR-W use \mathcal{W} -*preferences*, which are defined as follows. For any agent α_i and set of agents $S \subseteq N$ we denote by $\mathcal{W}_{\alpha_i}(S)$ the least-preferred agent in $S \setminus \{\alpha_i\}$ according to α_i . For any agent α_i and any two triples r and s , we say that α_i *prefers* r to s , denoted $r \succ_{\alpha_i} s$, if $\mathcal{W}_{\alpha_i}(r) \succ_{\alpha_i} \mathcal{W}_{\alpha_i}(s)$. A *matching* is a partition of N into n triples. Given an agent α_i and a matching M , we denote by $M(\alpha_i)$ the triple in M that contains α_i . Given a matching M , we say that a triple t is *blocking* if each agent α_i in t prefers t to $M(\alpha_i)$. A matching is *stable* if it does not contain a blocking triple. Let $\mathcal{W}_{\alpha_i}(M)$ be short for $\mathcal{W}_{\alpha_i}(M(\alpha_i))$. Let P be the collection of preference lists P_{α_i} for each agent α_i . For any instance (N, P) of 3DR-W and any matching M , we denote by $\text{bt}(M, (N, P)) \subseteq \binom{N}{3}$ the set of triples that block M in (N, P) . Conversely, we denote by $\text{nbt}(M, (N, P)) = \binom{N}{3} \setminus \text{bt}(M, (N, P))$ the set of triples that do not block M in (N, P) . When the instance in question is clear from context, we simply write $\text{bt}(M)$ or $\text{nbt}(M)$.

5.2 Deciding existence

In this section we show that deciding if a given instance of 3DR-W contains a stable matching is NP-complete. The reduction is from a restricted case of *Partition Into Triangles* (PIT, Problem 5.1) [43].

Problem 5.1. Partition Into Triangles (PIT)

Input: a simple undirected graph $G = (W, E)$ where $|W| = 3q$ for some integer q

Question: Can the vertices of G be partitioned into q disjoint sets $X = \{X_1, X_2, \dots, X_q\}$, each set containing exactly three vertices, such that each $X_p = \{w_i, w_j, w_k\} \in W$ where $1 \leq p \leq q$ is a triangle?

Given a graph simple undirected graph $G = (W, E)$, we say that a set of three vertices $\{w_i, w_j, w_k\}$ is a *triangle* if $\{w_i, w_j\} \in E$, $\{w_i, w_k\} \in E$, and $\{w_j, w_k\} \in E$. Let $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ be the set of triangles in G . We reduce from the special case of PIT in which \mathcal{T} has a *system of distinct representatives* (SDR), that is, a set $Z = \{z_1, z_2, \dots, z_m\}$ of m distinct vertices where $z_i \in T_i$ for each i where $1 \leq i \leq m$. We refer to this restricted problem as *PIT-SDR*. It can be verified that in the reduction shown by Garey and Johnson [43, Theorem 3.7] from *Exact Cover by Three-sets* to PIT that the constructed graph admits an SDR¹. It follows that

¹In the reduction of Garey and Johnson, every triangle in G is contained in exactly one subset gadget. For each subset gadget $c_i \in C$, the SDR contains the vertices labelled “ $a_i[1]$ ”, “ $a_i[2]$ ”, “ $a_i[4]$ ”, “ $a_i[5]$ ”, “ $a_i[7]$ ”, “ $a_i[8]$ ”, and “ $a_i[6]$ ”.

PIT-SDR is NP-complete.

The reduction from PIT-SDR is as follows. Suppose $G = (W, E)$ is an arbitrary graph. We shall construct an instance (N, P) of 3DR-W. For each vertex i where $1 \leq i \leq 3q$ construct a set of seven agents $H_i = \{h_i^1, h_i^2, \dots, h_i^7\}$, which we refer to as the i^{th} heptagadget. Let H^7 be the set $\bigcup_{1 \leq i \leq 3q} h_i^7$. We shall now construct the preferences of the agents in each heptagadget. First, for each heptagadget H_i construct the agents' preferences as follows. Note that in the following construction “...” denotes all remaining agents in an arbitrary order.

$$\begin{array}{ll}
 h_i^1 : & h_i^3 \ h_i^7 \ h_i^4 \ h_i^6 \ h_i^2 \ h_i^5 \ \dots \\
 h_i^2 : & h_i^7 \ h_i^1 \ h_i^3 \ h_i^6 \ h_i^4 \ h_i^5 \ \dots \\
 h_i^3 : & h_i^4 \ h_i^1 \ h_i^7 \ h_i^6 \ h_i^2 \ h_i^5 \ \dots \\
 h_i^4 : & h_i^6 \ h_i^7 \ h_i^1 \ h_i^3 \ h_i^2 \ h_i^5 \ \dots \\
 h_i^5 : & h_i^7 \ h_i^6 \ h_i^1 \ h_i^4 \ h_i^2 \ h_i^3 \ \dots \\
 h_i^6 : & h_i^1 \ h_i^3 \ h_i^7 \ h_i^4 \ h_i^2 \ h_i^5 \ \dots \\
 h_i^7 : & \underbrace{[\ h_j^7 \in (H^7 \setminus \{h_i^7\}) : \{w_i, w_j\} \in E \]}_{\text{proper part}} \ h_i^6 \ h_i^4 \ h_i^1 \ h_i^3 \ h_i^2 \ h_i^5 \ \dots
 \end{array}$$

We now reorder the proper part of the preference list of each agent in H^7 . Our aim is to ensure that for any three agents $h_i^7, h_j^7, h_k^7 \in H^7$ if the three corresponding vertices w_i, w_j, w_k form a triangle in G then one of the three agents is the least-preferred agent in the proper part of the preference list of at least one of the other two agents. To do this, first identify the set of triangles $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ and then construct an SDR of \mathcal{T} labelled $Z = \{z_1, z_2, \dots, z_m\}$. Note that Z can be constructed in polynomial time as a maximum matching in a bipartite graph. Next, for each i where $1 \leq a \leq m$ consider the triangle $T_a = \{w_i, w_j, w_k\}$ in G , labelling the representative vertex z_a in Z as w_i and the other two vertices as w_j and w_k arbitrarily. Since $\{w_i, w_j\} \in E$ it must be that h_j^7 appears in the proper part of $P_{h_i^7}$. Reorder the proper part of $P_{h_i^7}$ such that h_j^7 is now the least-preferred agent in the proper part. Note that by the definition of an SDR, no preference list is modified more than once and thus we have achieved our aim. This completes the construction of (N, P) .

It is straightforward to show that this reduction can be performed in polynomial time. To prove that the reduction is correct we show that a stable matching exists in the 3DR-W instance (N, P) if and only if a partition into triangles exists in the PIT-SDR instance G .

We first show that if the PIT-SDR instance G contains a partition into triangles then the 3DR-W instance (N, P) contains a stable matching.

Lemma 5.1. *If G contains a partition into triangles then (N, P) contains a stable matching.*

Proof. Suppose $\mathcal{X} = \{X_1, X_2, \dots, X_q\}$ is a partition into triangles in G . For each triangle $X_p = \{w_i, w_j, w_k\} \in \mathcal{X}$ construct in M the triples $\{h_i^7, h_j^7, h_k^7\}$, $\{h_i^1, h_i^3, h_i^4\}$, $\{h_i^2, h_i^5, h_i^6\}$, $\{h_j^1, h_j^3, h_j^4\}$, $\{h_j^2, h_j^5, h_j^6\}$, $\{h_k^1, h_k^3, h_k^4\}$ and $\{h_k^2, h_k^5, h_k^6\}$.

To show that M is stable, we consider each agent in an arbitrary heptagadget H_i .

First, consider h_i^7 . Suppose for a contradiction that h_i^7 belongs to a triple that blocks M . By the construction of M , $\mathcal{W}_{h_i^7}(M)$ belongs to the proper part of $P_{h_i^7}$ so it must be that this blocking triple comprises three agents $\{h_i^7, h_j^7, h_k^7\}$ where $\{w_i, w_j\} \in E$ and $\{w_i, w_k\} \in E$. A symmetric argument shows that h_i^7 must appear in the proper part of $P_{h_j^7}$ and thus that $\{w_i, w_j, w_k\}$ is a triangle in G . Suppose the triangle $\{w_i, w_j, w_k\}$ is labelled T_a where $1 \leq a \leq m$ and assume without loss of generality that w_i is the representative vertex z_a in the SDR Z of \mathcal{T} . It follows that $\{h_i^7, h_j^7, h_k^7\}$ contains the least-preferred agent in the proper part of $P_{h_i^7}$. Assume without loss of generality that h_j^7 is the least-preferred agent in the proper part of $P_{h_i^7}$. This contradicts the fact that h_j^7 must appear before $\mathcal{W}_{h_i^7}(M)$, which also belongs to the proper part of $P_{h_i^7}$. It remains that h_i^7 does not belong to a triple that blocks M .

Next, consider h_i^3 . No triple is preferred by h_i^3 to $M(h_i^3)$ so h_i^3 does not belong to a blocking triple. Similarly, the only triple that h_i^1 prefers to $M(h_i^1)$ contains h_i^7 , which we have shown does not belong to a blocking triple. It follows that h_i^1 also does not belong to a blocking triple. Any triple h_i^4 prefers to $M(h_i^4)$ and contains h_i^1 or h_i^3 is not blocking, and the only triple that h_i^4 prefers to $M(h_i^4)$ that contains neither h_i^1 nor h_i^3 is $\{h_i^4, h_i^6, h_i^7\}$. Since h_i^7 does not belong to a blocking triple it follows thus that h_i^4 does not belong to a blocking triple. Similarly, the only triples that h_i^2 , h_i^5 , and h_i^6 prefer to $M(h_i^2)$, $M(h_i^5)$, and $M(h_i^6)$ contain at least one of h_i^1 , h_i^3 , h_i^4 , and h_i^7 so are not blocking. It follows that neither h_i^2 , h_i^5 , nor h_i^6 belong to blocking triples. \square

We now show, using a sequence of lemmas, that if the 3DR-W instance (N, P) contains a stable matching then the PIT-SDR instance G contains a partition into triangles.

In the reduction, for some matching M we say that some agent h_i^r is *internal in M* if $M(h_i^r) \subset H_i$, some agent $h_i^7 \in H^7$ is *proper in M* if $\mathcal{W}_{h_i^7}(M)$ belongs to the proper part of $P_{h_i^7}$ and some agent $h_i^r \in N$ is *external in M* if h_i^r is neither proper nor internal. It follows that every agent in H_i is either proper, internal, or external in M . We will eventually show that in M no agent is external and every agent in H^7 is proper, from which the existence of a partition into triangles is straightforward to show.

Lemma 5.2. *If (N, P) contains a stable matching M then each heptagadget H_i contains exactly six agents that are internal in M .*

Proof. By definition, the number of internal agents in M in H_i is divisible by three. It follows that if H_i does not contain six agents that are internal in M it contains at most three agents

that are internal in M . Suppose for a contradiction that H_i contains at most three agents that are internal in M . Since by definition H_i contains at most one proper agent in M , namely h_i^7 , it follows that H_i contains at least three agents that are external in M , which we label h_i^r , h_i^s , and h_i^t . By the definition of an external agent it must be that $\mathcal{W}(\{h_i^s, h_i^t\}) \succ_{h_i^r} \mathcal{W}_{h_i^r}(M)$, $\mathcal{W}(\{h_i^r, h_i^t\}) \succ_{h_i^s} \mathcal{W}_{h_i^s}(M)$, and $\mathcal{W}(\{h_i^r, h_i^s\}) \succ_{h_i^t} \mathcal{W}_{h_i^t}(M)$. It follows that $\{h_i^r, h_i^s, h_i^t\}$ blocks M , which is a contradiction. \square

Lemma 5.3. *If (N, P) contains a stable matching M then no agent in N is external in M .*

Proof. Consider an arbitrary heptagadget H_i , which by Lemma 5.2 contains exactly six agents that are internal in M . The remaining agent is either external or proper in M . Suppose for a contradiction that the remaining agent h_i^r is external in M . If $r \neq 5$ then it must be that h_i^5 is internal in M and thus $M(h_i^5) = \{h_i^5, h_i^s, h_i^t\}$ where $s, t \in \{1, 2, 3, 4, 6, 7\}$. Notice that by the design of the constructed instance (N, P) , it must be that $h_i^r \succ_{h_i^s} h_i^5$ and $h_i^r \succ_{h_i^t} h_i^5$. Moreover, since h_i^r is external in M it follows that $\mathcal{W}_{h_i^r}(\{h_i^s, h_i^t\}) \succ_{h_i^r} \mathcal{W}_{h_i^r}(M)$ and thus $\{h_i^r, h_i^s, h_i^t\}$ blocks M , which is a contradiction. It remains that $r = 5$. In this case we consider the two triples in M that contain the six agents in H_i that are internal in M . We enumerate the $\binom{6}{2}/2$ possible such pairs of triples and show that in any case M is not stable. Since all agents in both triples belong to H_i , in the following table we shorten $\{h_i^r, h_i^s, h_i^t\}$ to $\{r, s, t\}$.

triples in M	M is blocked by
$\{1, 2, 3\}, \{4, 6, 7\}$	$\{1, 3, 6\}$
$\{1, 2, 4\}, \{3, 6, 7\}$	$\{1, 3, 4\}$
$\{1, 2, 6\}, \{3, 4, 7\}$	$\{1, 4, 6\}$
$\{1, 2, 7\}, \{3, 4, 6\}$	$\{1, 3, 7\}$
$\{1, 3, 4\}, \{2, 6, 7\}$	$\{4, 6, 7\}$
$\{1, 3, 6\}, \{2, 4, 7\}$	$\{1, 3, 4\}$
$\{1, 3, 7\}, \{2, 4, 6\}$	$\{4, 6, 7\}$
$\{1, 4, 6\}, \{2, 3, 7\}$	$\{1, 3, 7\}$
$\{1, 4, 7\}, \{2, 3, 6\}$	$\{4, 6, 7\}$
$\{1, 6, 7\}, \{2, 3, 4\}$	$\{1, 3, 4\}$

\square

We remark that from the proof of Lemma 5.3 it is straightforward to identify a (minimal) instance of 3DR-W that contains no stable matching, by using six agents $h_i^1, h_i^2, h_i^3, h_i^4, h_i^6$, and h_i^7 from a single gadget H_i .

Lemma 5.4. *If (N, P) contains a stable matching then G contains a partition into triangles.*

Proof. Suppose M is a stable matching in (N, P) . Recall that $|N| = 21q$. By Lemma 5.2, there are $18q$ internal agents that by definition belong to $6q$ triples in M . By Lemma 5.3, none of the remaining $3q$ agents are external and thus there are exactly $3q$ proper agents. It follows that there are q triples in M that each contain three proper agents. By the construction of the proper part of the preference list of each agent in H^7 , each of these q triples corresponds to a triangle in G . It follows that this set of q triples corresponds directly to a partition into triangles in G . \square

We have now shown that the 3DR-W instance (N, P) contains a stable matching if and only if a partition into triangles exists in G . This shows that the reduction is correct.

Theorem 5.1. *Deciding if a given instance of 3DR-W contains a stable matching is NP-complete.*

Proof. It is straightforward to show that this decision problem belongs to NP. We have presented a polynomial-time reduction from a restricted version of Partition Into Triangles, known as PIT-SDR, which we showed was NP-complete. Given an arbitrary instance G of PIT-SDR, the reduction constructs an instance (N, P) of 3DR-W. Lemmas 5.1 and 5.4 shows that (N, P) contains a stable matching if and only if G contains a partition into triangles and thus that this decision problem is NP-hard. \square

5.3 Approximation

The *3DR-W Maximally Stable Matching problem* (3DR-W-MSM) is the following optimisation problem: given an instance of 3DR-W, find a matching with the largest possible number of non-blocking triples. Formally, 3DR-W-MSM is a maximisation problem in which any instance (N, P) of 3DR-W-MSM is also instance of 3DR-W, a solution is a matching in (N, P) , and the measure is $|\text{nbt}(M, (N, P))|$. We showed in Theorem 5.1 that deciding if an instance of 3DR-W contains a matching with $\binom{3n}{3}$ non-blocking triples is NP-complete, so it follows that 3DR-W-MSM is NP-hard. In this section we present a polynomial-time algorithm that can approximate 3DR-W-MSM with an approximation ratio of $9/4$.

The approximation algorithm is analogous to the corresponding algorithm we presented before for 3DR-B-MSM in Theorem 4.2 in Chapter 4. As before, we show that it is possible to construct an instance (N, P') of 3PSA with the same set of agents such that for any matching M , if a triple blocks M in (N, P) then it also blocks M in (N, P') . We can then show, as before, that the approximation ratio of the algorithm for 3DR-W-MSM is $9/4$.

Theorem 5.2. *There exists a polynomial-time $9/4$ -approximation algorithm for 3DR-W-MSM.*

Proof. The proof is analogous to Theorem 4.2 in Chapter 4, except (N, P') is constructed such that for any agent α_i , P'_{α_i} is the list of all 2-agent subsets of $N \setminus \{\alpha_i\}$ in colexicographic order [38] with respect to P_{α_i} . Suppose M is an arbitrary matching in (N, P) and r is a triple that blocks M in (N, P) . We will show that r also blocks M in (N, P') . For any α_k in r it must be that $\mathcal{W}_{\alpha_k}(r) >_{P_{\alpha_k}} \mathcal{W}_{\alpha_k}(M)$. By the construction of P'_{α_k} as the colexicographic order of P_{α_k} , it must be that $r >_{P'_{\alpha_k}} M(\alpha_k)$ and thus that r also blocks M in (N, P') . It follows that $|\text{bt}(M, (N, P'))| \geq |\text{bt}(M, (N, P))|$ and thus that $|\text{nbt}(M, (N, P))| \geq |\text{nbt}(M, (N, P'))|$, as required. \square

5.4 Summary and open problems

In this chapter we formalised a new model of 3DR, involving \mathcal{W} -preferences, which we called 3DR-W. We considered in 3DR-W the existence of, and complexity of finding, matchings that are stable.

We first showed that, as in the case of 3DR-B (see Chapter 4), a given instance of 3DR-W may not contain a stable matching and the associated decision problem is NP-complete. Next, we considered a closely related optimisation problem, which we called 3DR-W-MSM, in which the objective is to construct, in a given instance of 3DR-B, a matching with the maximum number of non-blocking triples. Finally, we presented a 9/4-approximation algorithm for 3DR-W-MSM built on an existing algorithm for 3PSA-MSM, which is a closely related problem [75].

We now present some open problems specifically involving stability in 3DR-W. More general problems, involving solution concepts other than stability and other models of fixed-size coalition formation, are discussed in Chapter 9.

Although the definitions of 3DR-B and 3DR-W are similar, and the stability existence problem is NP-complete in both models, it is unclear whether the two models are further related. For example, in Chapter 4 we presented a 3/2-approximation algorithm for 3DR-B-MSM, but it seems difficult to design such an algorithm for 3DR-W-MSM. One possible starting point is Rosenbaum's 9/4-approximation algorithm for 3PSA-MSM, Algorithm ASA. Another idea is to adapt Irving's [40] algorithm for (two-dimensional) Stable Roommates. In fact, Irving's algorithm is the basis of another polynomial-time algorithm for an analogous model involving \mathcal{W} -preferences [8], in which coalitions need not have a fixed size.

Alternatively, it might be possible to prove an inapproximability result for 3DR-W-MSM. For example, if one could show that no 3/2-approximation algorithm exists for 3DR-W-MSM, unless $\text{P} = \text{NP}$, then a striking difference would be revealed between 3DR-B and 3DR-W. To this end, it might be useful to consider the reductions of Iwama et al. [61] and Rosenbaum

[75], which show that two optimisation problems that are defined in related models of 3DR are both APX-hard (both results are discussed in Chapter 2).

It would be very informative to estimate the probability that a random instance of 3DR-W contains a stable matching. In particular, if this probability could be compared between the models of 3DR-B and 3DR-W. As we noted in Chapter 4, both 3-DSM-CYC [53] and two-dimensional Stable Roommates (SR) [81] have been studied from a probabilistic perspective, which are possible starting points. Alternatively, an empirical approach might be informative, as it has been for 3-DSM-CYC [51]. In this direction, it might be useful to formulate the problem of finding a stable matching in a given instance of 3DR-W as an integer programming model.

Various other optimisation problems and measures can be defined in relation to stability and 3DR-W. In Chapter 4 we proposed some alternatives in the setting of 3DR-B, all of which can be defined analogously for 3DR-W.

Chapter 6

Three-Dimensional Stable Roommates with Additively Separable Preferences

6.1 Introduction

In this chapter we formalise a model of Three-Dimensional Roommates (3DR) involving additively separable preferences, which we call 3DR-AS. We consider in 3DR-AS the existence of, and complexity of finding, matchings that are stable, under three possible restrictions of the agents' preferences.

A strong motivation exists for a model of 3DR with additively separable preferences. The first such model, of which 3DR-AS is a generalisation, was first proposed by Huang [29] in 2007, who noted the natural definition and relative practicality of additively separable preferences compared to other possible systems of preference representation. For example, such a model could be applied to a social network graph involving a symmetric “friendship” relation between users [4]. Another special case of 3DR-AS is *Geometric 3D-SR* [62] (see Chapter 2). In a sense, all of these models can also be considered as a special type of additively separable hedonic game [15, 34], which have received much attention in the literature (and are also discussed in Chapter 2).

We begin in Section 6.2 with some preliminary definitions and results.

We then show, in Section 6.3, that any instance of 3DR-AS with binary and symmetric preferences must contain a stable matching, and present a polynomial-time algorithm that can construct a stable matching in a given such instance (Theorem 6.1). We then consider the problem of finding a stable matching with maximum utilitarian welfare, given an instance in which preferences are binary and symmetric. We show that this optimisation problem is

NP-hard (Theorem 6.2) but also that the algorithm for constructing a stable matching in this setting can be modified to yield a 2-approximation algorithm (Theorem 6.3).

Next, we complement the previous tractability results with two hardness results. The first, shown in Section 6.4, is that a stable matching need not exist in general, and the associated decision problem is NP-complete even when preferences are binary and not necessarily symmetric (Theorem 6.4). The second, shown in Section 6.5, is that the same decision problem is NP-complete even when preferences are ternary and symmetric (Theorem 6.5).

Finally, in Section 6.6, we recap on our contribution and discuss some directions for future work.

6.2 Preliminaries

An instance of 3DR-AS comprises a set N of $3n$ agents with additively separable preferences over triples, which we define as follows. Each agent α_i supplies a *valuation function* $v_{\alpha_i} : N \setminus \{\alpha_i\} \mapsto \mathbb{Z}$. Given agent α_i , let the *utility* of any set $S \subseteq N$ be $u_{\alpha_i}(S) = \sum_{\alpha_j \in S \setminus \{\alpha_i\}} v_{\alpha_i}(\alpha_j)$. A *triple* is an unordered set of three agents. We say that $\alpha_i \in N$ *prefers* some triple r to another triple s if $u_{\alpha_i}(r) > u_{\alpha_i}(s)$. A *matching* is a partition of N into n triples (note that we shall slightly modify this definition in Section 6.3.1). Given an agent α_i and a matching M , we denote by $M(\alpha_i)$ the triple in M that contains α_i . Given a matching M , we say that a triple t is *blocking* if each agent α_i in t prefers t to $M(\alpha_i)$. A matching is *stable* if it does not contain a blocking triple. An agent's preference between two matchings depends only on the partners of that agent in each matching, so given a matching M we let $u_{\alpha_i}(M)$ be short for $u_{\alpha_i}(M(\alpha_i))$. Formally, we represent an instance of 3DR-AS as a pair (N, V) , where V is the collection of all agents' valuation functions.

For any instance (N, V) of 3DR-AS and any matching M in (N, V) , for a set $S \subseteq N$ of agents we define the *utilitarian welfare* of S as $u_S(M) = \sum_{\alpha_i \in S} u_{\alpha_i}(M)$. Let $u(M)$ be short for $u_N(M)$.

We also define three possible restrictions of agents' preferences. We say that valuations are *binary*, if $v_{\alpha_i}(\alpha_j) \in \{0, 1\}$ for any $\alpha_i, \alpha_j \in N$, *ternary* if $v_{\alpha_i}(\alpha_j) \in \{0, 1, 2\}$ for any $\alpha_i, \alpha_j \in N$, and *symmetric* if $v_{\alpha_i}(\alpha_j) = v_{\alpha_j}(\alpha_i)$ for any $\alpha_i, \alpha_j \in N$.

It is possible to consider an instance of 3DR-AS as a weighted directed graph, in which the set of vertices is the set of agents and the weight of any arc from one agent to another is the corresponding valuation. In particular, an instance of 3DR-AS with binary and symmetric preferences corresponds directly to an undirected graph, which we shall refer to as the *underlying graph*.

By the definition of stability, if M and M' are matchings in some instance of 3DR-AS, M is stable, and $u_{\alpha_i}(M') \geq u_{\alpha_i}(M)$ for any agent α_i then it must follow that M' is also stable. We prove a related statement in Proposition 6.1.

Proposition 6.1. *Given an instance (N, V) of 3DR-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 $u_{\alpha_i}(M') < u_{\alpha_i}(M)$.*

Proof. Suppose some triple $\{\alpha_i, \alpha_j, \alpha_k\}$ blocks M' . By the definition of a blocking triple, it must be that $u_{\alpha_i}(\{\alpha_j, \alpha_k\}) > u_{\alpha_i}(M')$, $u_{\alpha_j}(\{\alpha_i, \alpha_k\}) > u_{\alpha_j}(M')$, and $u_{\alpha_k}(\{\alpha_i, \alpha_j\}) > u_{\alpha_k}(M')$. Suppose for a contradiction that there exists no $\alpha_p \in \{\alpha_i, \alpha_j, \alpha_k\}$ exists where $u_{\alpha_p}(M') < u_{\alpha_p}(M)$ and hence $u_{\alpha_{kr}}(M') \geq u_{\alpha_{kr}}(M)$ for $1 \leq r \leq 3$. It follows that $u_{\alpha_i}(\{\alpha_j, \alpha_k\}) > u_{\alpha_i}(M)$, $u_{\alpha_j}(\{\alpha_i, \alpha_k\}) > u_{\alpha_j}(M)$, and $u_{\alpha_k}(\{\alpha_i, \alpha_j\}) > u_{\alpha_k}(M)$ and thus that $\{\alpha_i, \alpha_j, \alpha_k\}$ also blocks M , which is a contradiction. \square

We also introduce some other notation. We denote by $L = \langle l_1, l_2, \dots, l_{|L|} \rangle$ an ordered list. If L and L' are lists then we denote by $L \cdot L'$ the concatenation of L' to the end of L . We also use standard set notation with lists, such as $e \in L$.

6.3 Symmetric binary preferences

6.3.1 Finding a stable matching

6.3.1.1 Preliminaries

In this section we show that every such instance of 3DR-AS contains a stable matching, which can be found in $O(|N|^3)$ time. We give a step-by-step constructive proof of this result between Sections 6.3.1.2–6.3.1.5.

For technical purposes, in this section (Section 6.3.1) only, we make two relaxations to the model described in Section 6.2. In fact, we prove a slightly more general result than required. Firstly, we suppose an instance (N, V) of 3DR-AS may contain a number of agents that is not necessarily divisible by three. Secondly, we define a *matching* only as a set of triples, which does not necessarily partition N . For any matching M and any agent α_i , if no triple in M contains α_i then we say that α_i is *unmatched* in M and write $M(\alpha_i) = \emptyset$. All other definitions and notation remain the same. Thus, in this section we show that any (relaxed) instance of 3DR-AS contains a (relaxed) stable matching. Now, for any (relaxed) instance of 3DR-AS and (relaxed) stable matching M , if the number of agents is divisible by three (and thus the instance meets the original definition) then any unmatched agents may be arbitrarily

matched into triples. It follows by Proposition 6.1 that the resulting (non-relaxed) matching M' is stable in the (non-relaxed) instance, as required.

We also introduce a restricted type of (relaxed) 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 (relaxed) matching M . We say that a matching M in (N, V) is a *P-matching* if $M(\alpha_p) \neq \emptyset$ implies $u_{\alpha_p}(M) > 0$. A *P-matching* thus corresponds to a $\{K_3, P_3\}$ -packing in the underlying graph [82]. Note that any triple in a *P-matching* M must contain at least one agent with utility two. A *stable P-matching* is a *P-matching* that is also stable. Our main result is that any instance of 3DR-AS with binary and symmetric preferences contains a stable *P-matching*.

6.3.1.2 Removing triangles

In an instance (N, V) of 3DR-AS with binary and symmetric preferences, a *triangle* comprises three agents $\alpha_{m_1}, \alpha_{m_2}, \alpha_{m_3}$ such that $v_{\alpha_{m_1}}(\alpha_{m_2}) = v_{\alpha_{m_2}}(\alpha_{m_3}) = v_{\alpha_{m_3}}(\alpha_{m_1}) = 1$. 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 triangles, until it is triangle-free. This operation is our first step towards constructing a stable *P-matching* in (N, V) . We formalise this result in Lemma 6.1.

Lemma 6.1. *Given an instance (N, V) of 3DR-AS with binary and symmetric preferences, we can, in $O(|N|^3)$ time, identify an instance (N', V') of 3DR-AS with binary and symmetric preferences and a *P-matching* M_Δ 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) .*

Proof. Construct M_Δ as a *maximal triangle packing* [83] in the underlying graph, in $O(|N|^3)$ time. Let $N' = N \setminus \bigcup M_\Delta$. Construct V' accordingly. By definition, M_Δ is a *P-matching*, (N', V') is triangle-free, and $|N'| \leq |N|$.

Suppose M is a stable *P-matching* in (N', V') . Consider $M' = M \cup M_\Delta$. By definition, M' is a *P-matching*. Since each triple in M_Δ corresponds to a triangle, any agent in any triple in M_Δ must have utility two in M_Δ and thus does not belong to a triple that blocks M' in (N, V) . If some triple blocks M' in (N, V) then it must contain three agents in N' , and thus it must also block M in (N', V') , which is impossible. It follows that M' is stable in (N, V) . \square

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

In this section (Section 6.3.1.3), we consider instances of 3DR-AS that are triangle-free, and in them define a special type of *P-matching* that is *repairable*.

We present Subroutine *repair*, shown in Algorithms 6.1 and 6.2 which, given a triangle-free instance (N, V) and a repairable *P-matching* M , constructs a new *P-matching* M' that is

in Algorithm 6.1, it identifies some set of agents in the instance with a specific structure. In Phase 2, shown in Algorithm 6.1, it modifies the triples of these agents in order to construct a stable P -matching M' . Phase 1 involves the construction of a list of agents S , which initially comprises $\langle \alpha_{j_1}, \alpha_{j_3}, \alpha_{j_4} \rangle$. At any point in time, the list S has length $3c$ for some $c \geq 1$ where $\{S_{3c-2}, S_{3c-1}, S_{3c}\} \in M$ and $v_{S_p}(S_{p+1}) = 1$ for any p where $1 \leq p < 3c$. It follows that S corresponds to a path in the underlying graph. In each iteration of the main loop, three agents 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 stopping conditions (shown in the first if/else statement). We will show that eventually at least one of these stopping conditions must hold. After the loop terminates, the subroutine enters Phase 2 and constructs M' . The exact construction of M' depends on which stopping condition(s) caused the main loop to terminate. Two of these conditions, and the corresponding constructions of M' , generalise the two example cases (involving α_{z_1} and α_{z_2}). The other four conditions, and the corresponding constructions of M' , relate to alternative cases in which it is possible to construct a stable P -matching M' .

The six stopping conditions correspond to seven possible constructions of M' , which are labelled Construction 1–7. Each of the six stopping conditions corresponds to a single construction except the first condition, which corresponds to two constructions (Construction 1 and Construction 3). Constructions 1 and 3 generalise the first example case (involving α_{z_1}). Construction 2 generalises the second example case (involving α_{z_2}). Constructions 4–7 correspond to alternative cases. Like in the two example cases, in each of Constructions 1–6 no agent identified by the subroutine, including α_i , becomes unmatched in M' . This simplifies the proof that M' is stable in Constructions 1–6. The proof that M' is stable in Construction 7 is more complicated. In Construction 7, the final agent in the list S , labelled S_{3c} , becomes unmatched in M' . To prove that S_{3c} does not then become part of a triple that blocks M' we must invoke on the fact that no stopping condition relating to previous constructions held in any previous iteration of the main loop. In this way, the six stopping conditions and seven corresponding constructions of M' are somewhat hierarchical. For another example, the proof that M' is stable in Construction 4 relies on the fact that in no previous iteration did the stopping condition relating to Constructions 1 and 3 hold. A similar reliance exists among the proofs for the other constructions. This hierarchy helps demonstrate why the six stopping conditions and seven constructions of M' are both necessary and sufficient.

In order to prove the correctness and time complexity of Subroutine *repair* we use a number of lemmas. The following lemma, Lemma 6.2, shows that the while loop in Subroutine *repair* must terminate in $O(|N|)$ time.

Lemma 6.2. *The while loop in Subroutine *repair* terminates after at most $\lfloor (|N| - 2) / 3 \rfloor$ iterations.*

Proof. By the pseudocode, any three agents $\{\alpha_{w_1}, \alpha_{w_2}, \alpha_{w_3}\}$ added to S in a single iteration

form a triple in M . Just before the addition of $\langle \alpha_{w_1}, \alpha_{w_2}, \alpha_{w_3} \rangle$ to S , it must be that $\alpha_{w_1} \notin S$. It follows that $\alpha_{w_2}, \alpha_{w_3} \notin S$, so in general S contains any agent in N at most once. Since $\alpha_i, \alpha_{j_2} \notin S$ it follows that $|S| \leq |N| - 2$ and thus the while loop terminates after at most $\lfloor (|N| - 2) / 3 \rfloor$ iterations. \square

In Construction 3, the subroutine identifies some agent α_{z_4} in $N \setminus \{\alpha_i\}$ such that $v_{S_{3c-1}}(\alpha_{z_4}) = 1$ and $u_{\alpha_{z_4}}(M) = 0$. Proposition 6.2 shows that such an agent must exist.

Proposition 6.2. *In Construction 3 of Subroutine repair, some agent α_{z_4} in $N \setminus \{\alpha_i, \alpha_{j_2}\}$ exists where $v_{S_{3c-2}}(\alpha_{z_4}) = 1$ and $u_{\alpha_{z_4}}(M) = 0$.*

Proof. Refer to Figure 6.2. We first claim that $c > 1$. Suppose for a contradiction that $c = 1$. By the pseudocode, in Construction 3 it must be that $\alpha_{z_1} = \alpha_{j_2}$ and $v_{\alpha_{z_1}}(S_{3c-1}) = 1$. Since $c = 1$ it must be that $S_{3c-1} = \alpha_{j_3}$ so the triple $\{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$ forms a triangle in (N, V) , which contradicts the assumption that (N, V) is triangle-free.

Algorithm 6.1 Subroutine repair (Phase 1)

Input: a triangle-free instance (N, V) of 3DR-AS with binary and symmetric preferences and a repairable P -matching M in (N, V) (see Section 6.3.1.3) with some such α_i

Output: a 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  $v_{\alpha_{z_1}}(S_{3c-1}) = 1$  and  $u_{\alpha_{z_1}}(M) = 0$ , otherwise  $\perp$ 
   $\alpha_{z_2} \leftarrow$  some  $\alpha_{z_2} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$  where  $v_{\alpha_{z_2}}(S_{3c}) = 1$  and  $u_{\alpha_{z_2}}(M) = 0$ , otherwise  $\perp$ 
   $\alpha_{y_1} \leftarrow$  some  $\alpha_{y_1} \in N$  where  $v_{S_{3c}}(\alpha_i) = v_{\alpha_{y_1}}(\alpha_i) = 1$  and  $u_{\alpha_{y_1}}(M) = 0$ , otherwise  $\perp$ 
   $\alpha_{y_2} \leftarrow$  some  $\alpha_{y_2} \in N$  where  $v_{S_{3c}}(\alpha_{j_2}) = v_{\alpha_{y_2}}(\alpha_{j_2}) = 1$  and  $u_{\alpha_{y_2}}(M) = 0$ , otherwise  $\perp$ 
   $b \leftarrow$  some  $1 \leq b < c$  where  $v_{S_{3b}}(\alpha_{j_2}) = v_{S_{3c}}(S_{3b}) = 1$  else 0
   $\alpha_{w_1} \leftarrow$  some  $\alpha_{w_1} \in N$  where  $v_{S_{3c}}(\alpha_{w_1}) = 1$ ,  $u_{\alpha_{w_1}}(M) = 1$  and  $\alpha_{w_1} \notin S$  and there
    exists some  $\alpha_{z_3} \in N \setminus \{\alpha_i\}$  where  $v_{\alpha_{w_1}}(\alpha_{z_3}) = 1$  and  $u_{\alpha_{z_3}}(M) = 0$ , otherwise  $\perp$ 
  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 in Phase 2)

Algorithm 6.2 Subroutine repair (Phase 2)*(continued from Phase 1)***if** $\alpha_{z_1} \neq \perp$ and $\alpha_{z_1} \neq \alpha_{j_2}$ **then** \triangleright Construction 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** \triangleright Construction 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** \triangleright Construction 3 (note that α_{z_4} must exist by Proposition 6.2)

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

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

else if $\alpha_{y_1} \neq \perp$ **then** \triangleright Construction 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** \triangleright Construction 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** \triangleright Construction 6 (note that α_{z_5} must exist by Proposition 6.3)

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

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

else \triangleright Construction 7 (note that $\alpha_{w_1} = \perp$)

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

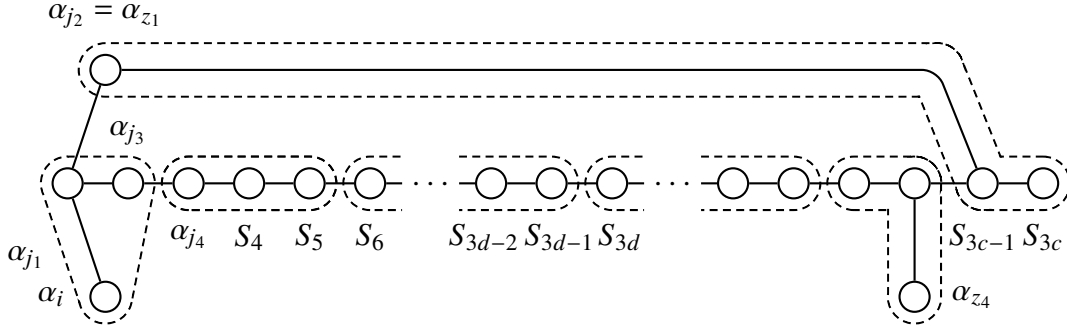
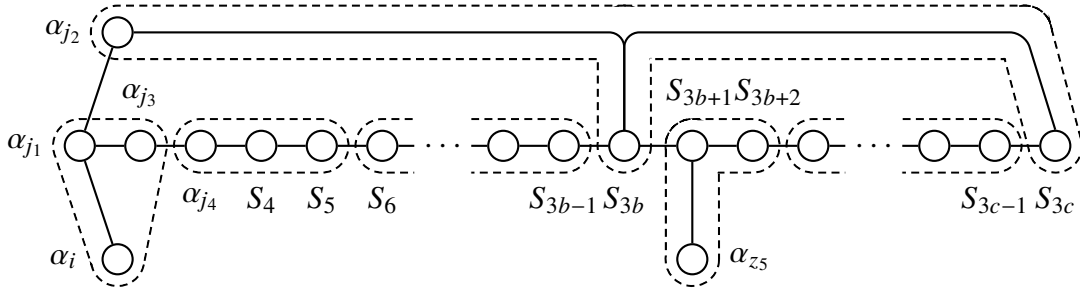
end if

$$M' \leftarrow M_S \cup \{r \in M : r \cap S = \emptyset\}$$

return M'

Since $c > 1$ it follows that $c' = c - 1$ is the value of c in the second last iteration of the while loop. Consider the second-to-last iteration of the while loop. In this iteration, the subroutine identified some $\alpha_{w_1} = S_{3c-2}$ where $v_{S_{3c'}}(\alpha_{w_1}) = 1$, $\alpha_{w_1} \notin S$ and there existed some $\alpha_{z_3} \in N \setminus \{\alpha_i\}$ where $v_{\alpha_{w_1}}(\alpha_{z_3}) = 1$ and $u_{\alpha_{z_3}}(M) = 0$. We shall identify the agent labelled α_{z_3} in this iteration as α_{z_4} . It follows that $\alpha_{z_4} \neq \alpha_i$, $v_{S_{3c-2}}(\alpha_{z_4}) = 1$, and $u_{\alpha_{z_4}}(M) = 0$.

We claim that in $\alpha_{z_4} \neq \alpha_{j_2}$ since otherwise the triple $\{\alpha_{z_4}, S_{3c-1}, S_{3c-2}\}$ forms a triangle in (N, V) , which contradicts the fact that (N, V) is triangle-free. It follows that $\alpha_{z_4} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$, which completes the proof. \square

Figure 6.2: The structure of M' in Construction 3Figure 6.3: The structure of M' in Construction 6

Likewise in Construction 6, the subroutine identifies some agent α_{z_5} in $N \setminus \{\alpha_i, \alpha_{j_2}\}$ exists where $v_{S_{3b+1}}(\alpha_{z_5}) = 1$ and $u_{\alpha_{z_5}}(M) = 0$. Proposition 6.3 shows that such an agent must exist.

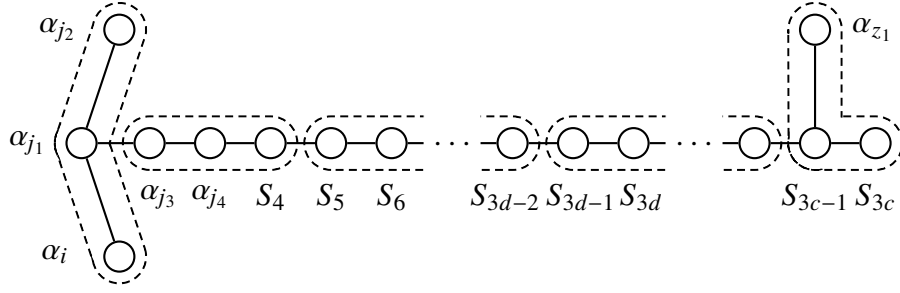
Proposition 6.3. *In Construction 6 of Subroutine repair, some agent α_{z_5} in $N \setminus \{\alpha_i, \alpha_{j_2}\}$ exists where $v_{S_{3b+1}}(\alpha_{z_5}) = 1$ and $u_{\alpha_{z_5}}(M) = 0$.*

Proof. Refer to Figure 6.3. Consider the final value of b in the subroutine. By the definition of b and the pseudocode relating to Construction 6 it must be that $b < c$.

Consider the b^{th} iteration of the while loop. Since $b < c$, it must be that this iteration was not the final iteration. It follows that at the end of this iteration the subroutine identified some agent $\alpha_{w_1} = S_{3b+1}$ and then appended $\langle S_{3b+1}, S_{3b+2}, S_{3b+3} \rangle$ to the end of S . It also follows that, in this iteration, it also identified some agent α_{w_1} where there exists some $\alpha_{z_3} \in N \setminus \{\alpha_i\}$ such that $v_{\alpha_{w_1}}(\alpha_{z_3}) = 1$ and $u_{\alpha_{z_3}}(M) = 0$. We shall identify the agent labelled α_{z_3} in this iteration as α_{z_5} . It follows that $\alpha_{z_5} \neq \alpha_i$, $v_{S_{3b+1}}(\alpha_{z_5}) = 1$, and $u_{\alpha_{z_5}}(M) = 0$.

We claim that in $\alpha_{z_5} \neq \alpha_{j_2}$. By the definition of b , it must be that $v_{S_{3b}}(\alpha_{j_2}) = 1$. Thus, if $\alpha_{z_5} = \alpha_{j_2}$ then the triple $\{\alpha_{z_5}, S_{3b+1}, S_{3b}\}$ forms a triangle in (N, V) , which contradicts the fact that (N, V) is triangle-free. It follows that $\alpha_{z_5} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$, which completes the proof. \square

Lemma 6.3. *Subroutine repair returns a P -matching.*

Figure 6.4: The structure of M' in Construction 1

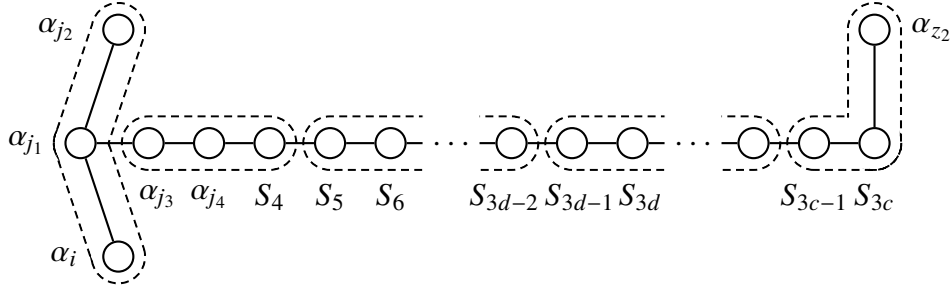
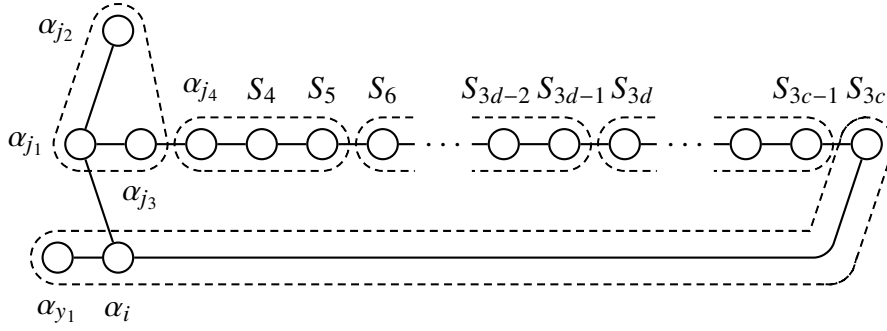
Proof. By the construction of M' in Constructions 1–7 of M' , shown in Figures 6.4–6.8. \square

In Lemmas 6.4, 6.5 and 6.6 we show that in no construction of M' does there exist any agent α_g where $u_{\alpha_g}(M') < u_{\alpha_g}(M)$ and α_g belongs to a triple that blocks M' . This fact will help us show that M' must be stable.

Lemma 6.4. *In Constructions 1 and 3 of Subroutine repair, there exists no agent α_g where $u_{\alpha_g}(M') < u_{\alpha_g}(M)$ and α_g belongs to a triple that blocks M' .*

Proof. Refer to Figures 6.2 and 6.4. Suppose for a contradiction that there exists some such $\alpha_g \in N$. By the construction of M' in Constructions 1 and 3, $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N \setminus S$. It follows that $\alpha_g \in S$ and thus by the construction of M' in Constructions 1 and 3 that $u_{\alpha_g}(M') \geq 1$. Since $u_{\alpha_g}(M') < u_{\alpha_g}(M)$ by assumption it must be that $u_{\alpha_g}(M) = 2$. The only such agents in S are labelled S_{3d-1} for some d where $1 \leq d \leq c$, so it must be that $\alpha_g = S_{3d-1}$ for some such d .

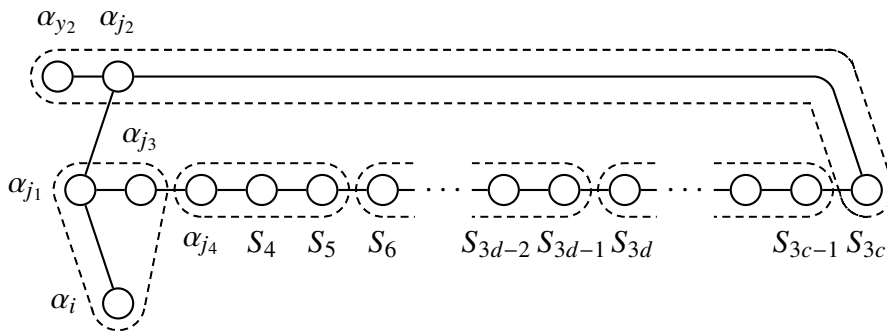
First consider S_{3c-1} . Since $u_{S_{3c-1}}(M') = 2$ it follows that S_{3c-1} does not belong to a triple that blocks M' and hence $\alpha_g \neq S_{3c-1}$. It remains that $\alpha_g = S_{3d-1}$ where $1 \leq d < c$. By assumption, it must be that some triple $\{S_{3d-1}, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' , where $\alpha_{k_1}, \alpha_{k_2} \in N$. Since $u_{S_{3d-1}}(M') = 1$ it follows that $u_{S_{3d-1}}(\{\alpha_{k_1}, \alpha_{k_2}\}) = 2$ and thus that $v_{S_{3d-1}}(\alpha_{k_1}) = v_{S_{3d-1}}(\alpha_{k_2}) = 1$. Consider α_{k_1} and α_{k_2} . Since (N, V) is triangle-free, it must be that $v_{\alpha_{k_1}}(\alpha_{k_2}) = 0$ and thus that $u_{\alpha_{k_1}}(\{S_{3d-1}, \alpha_{k_2}\}) = u_{\alpha_{k_2}}(\{S_{3d-1}, \alpha_{k_1}\}) = 1$. Since the triple $\{S_{3d-1}, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' it follows that $u_{\alpha_{k_1}}(M') = u_{\alpha_{k_2}}(M') = 0$. By the construction of M' , there exists no $\alpha_p \in N$ where $u_{\alpha_p}(M') = 0$ and $u_{\alpha_p}(M') < u_{\alpha_p}(M)$. It follows that $u_{\alpha_{k_1}}(M) = u_{\alpha_{k_2}}(M) = 0$. Recall the d^{th} iteration of the while loop. We have shown that two agents $\alpha_{k_1}, \alpha_{k_2}$ exist in that iteration such that $v_{S_{3d-1}}(\alpha_{k_1}) = v_{S_{3d-1}}(\alpha_{k_2}) = 1$ and $u_{\alpha_{k_1}}(M) = u_{\alpha_{k_2}}(M) = 0$. It follows that, in that iteration, there existed some $\alpha_{z_1} \in N \setminus \{\alpha_i\}$ where $v_{\alpha_{z_1}}(S_{3d-1}) = 1$ and $u_{\alpha_{z_1}}(M) = 0$, since either $\alpha_{z_1} = \alpha_{k_1}$ or $\alpha_i = \alpha_{k_1}$ and $\alpha_{z_1} = \alpha_{k_2}$. In that iteration, since $\alpha_{z_1} \neq \perp$ the break condition held and the while loop terminated. This is a contradiction since $d < c$. \square

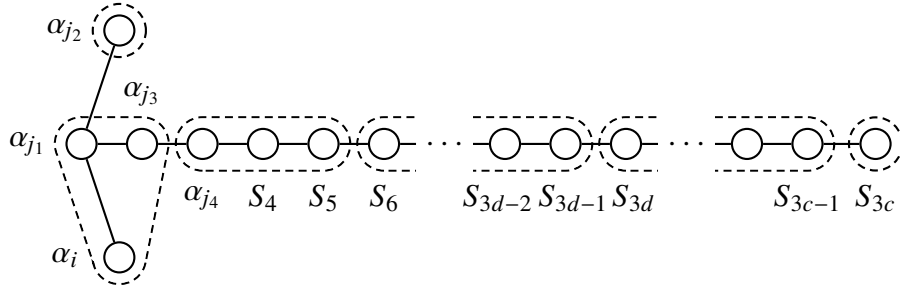
Figure 6.5: The structure of M' in Construction 2Figure 6.6: The structure of M' in Construction 4

Lemma 6.5. *In Constructions 2, 4, 5, and 6 of Subroutine repair, there exists no agent α_g where $u_{\alpha_g}(M') < u_{\alpha_g}(M)$ and α_g belongs to a triple that blocks M' .*

Proof. Refer to Figures 6.3, 6.5, 6.6, and 6.7. Suppose for a contradiction that there exists some such $\alpha_g \in N$. By the construction of M' in Constructions 2, 4, 5, and 6, $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N \setminus S$. It follows that $\alpha_g \in S$ and hence $u_{\alpha_g}(M') \geq 1$. Since $u_{\alpha_g}(M') < u_{\alpha_g}(M)$ it must be that $u_{\alpha_g}(M) = 2$. The only such agents in S are labelled S_{3d-1} for some d where $1 \leq d \leq c$, so it must be that $\alpha_g = S_{3d-1}$ for some such d .

Consider S_{3d-1} for $1 \leq d \leq c$. Note that in Constructions 2, 4, 5, and 6 it must be that $u_{S_{3d-1}}(M) = 2$ and $u_{S_{3d-1}}(M') = 1$. By assumption, it must be that some triple $\{S_{3d-1}$,

Figure 6.7: The structure of M' in Construction 5

Figure 6.8: The structure of M' in Construction 7

$\alpha_{k_1}, \alpha_{k_2}$ blocks M' , where $\alpha_{k_1}, \alpha_{k_2} \in N$. Since $u_{S_{3d-1}}(M') = 1$ it follows that $u_{S_{3d-1}}(\{\alpha_{k_1}, \alpha_{k_2}\}) = 2$. Consider α_{k_1} and α_{k_2} . Since (N, V) is triangle-free, it must be that $v_{\alpha_{k_1}}(\alpha_{k_2}) = 0$ and thus that $u_{\alpha_{k_1}}(\{S_{3d-1}, \alpha_{k_2}\}) = u_{\alpha_{k_2}}(\{S_{3d-1}, \alpha_{k_1}\}) = 1$. It follows that $u_{\alpha_{k_1}}(M') = u_{\alpha_{k_2}}(M') = 0$. By the construction of M' , there exists no $\alpha_p \in N$ where $u_{\alpha_p}(M') = 0$ and $u_{\alpha_p}(M) < u_{\alpha_p}(M)$. It follows that $u_{\alpha_{k_1}}(M) = u_{\alpha_{k_2}}(M) = 0$. Recall the d^{th} iteration of the while loop. We have shown that two agents $\alpha_{k_1}, \alpha_{k_2}$ exist where $v_{S_{3d-1}}(\alpha_{k_1}) = v_{S_{3d-1}}(\alpha_{k_2}) = 1$ and $u_{\alpha_{k_1}}(M) = u_{\alpha_{k_2}}(M) = 0$. It follows that, in that iteration, there existed some $\alpha_{z_1} \in N \setminus \{\alpha_i\}$ where $v_{\alpha_{z_1}}(S_{3d-1}) = 1$ and $u_{\alpha_{z_1}}(M) = 0$, since either $\alpha_{z_1} = \alpha_{k_1}$ or $\alpha_i = \alpha_{k_1}$ and $\alpha_{z_1} = \alpha_{k_2}$. In that iteration, since $\alpha_{z_1} \neq \perp$ the break condition must have held, the while loop terminated, and the condition for either Construction 1 or Construction 3 was true. This is a contradiction since by assumption the subroutine constructed M' according to one of Constructions 2, 4, 5, or 6. \square

Lemma 6.6. *In Construction 7 of Subroutine repair, there exists no agent α_g where $u_{\alpha_g}(M') < u_{\alpha_g}(M)$ and α_g belongs to a triple that blocks M' .*

Proof. Refer to Figure 6.8. Suppose for a contradiction that there exists some such α_g .

First, consider any $\alpha_p \in N$ where $\alpha_p \notin S \cup \{\alpha_{j_2}, \alpha_i\}$. By the construction of M' , it can be seen that $M(\alpha_p) = M'(\alpha_p)$ so $u_{\alpha_p}(M) = u_{\alpha_p}(M')$ and thus $\alpha_g \notin S \cup \{\alpha_{j_2}, \alpha_i\}$.

Next, consider α_i and α_{j_2} . Since $u_{\alpha_i}(M) = 0 < 1 = u_{\alpha_i}(M')$ it follows that $\alpha_g \neq \alpha_i$. Similarly, since $u_{\alpha_{j_2}}(M) = u_{\alpha_{j_2}}(M') = 0$ it follows that $\alpha_g \neq \alpha_{j_2}$.

It remains that $\alpha_g \in S$.

Consider any S_{3d-2} where $1 \leq d \leq c$. By construction of M' it follows that $u_{S_{3d-2}}(M') = 2$ so $\alpha_p \neq S_{3d-2}$ for any d where $1 \leq d \leq c$.

Next, consider any S_{3d-1} where $1 \leq d \leq c$. Suppose for a contradiction that $\alpha_g = S_{3d-1}$ where $1 \leq d \leq c$ and thus that some triple $\{S_{3d-1}, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' where $\alpha_{k_1}, \alpha_{k_2} \in N$. Since $u_{S_{3d-1}}(M') = 1$ it follows that $u_{S_{3d-1}}(\{\alpha_{k_1}, \alpha_{k_2}\}) = 2$. Consider α_{k_1} and α_{k_2} . Since (N, V) is triangle-free, it must be that $v_{\alpha_{k_1}}(\alpha_{k_2}) = 0$ and thus that $u_{\alpha_{k_1}}(\{S_{3d-1}, \alpha_{k_2}\}) = u_{\alpha_{k_2}}(\{S_{3d-1}, \alpha_{k_1}\}) = 1$. It follows that $u_{\alpha_{k_1}}(M') = u_{\alpha_{k_2}}(M') = 0$. By construction of M' it can be seen

that S_{3c} is the only agent α_p in N where $u_{\alpha_p}(M') = 0$ and $u_{\alpha_p}(M') < u_{\alpha_p}(M)$. Since the triple $\{S_{3d-1}, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' and $u_{\alpha_{k_1}}(\{S_{3d-1}, \alpha_{k_2}\}) = u_{\alpha_{k_2}}(\{S_{3d-1}, \alpha_{k_1}\}) = 1$ it follows that either $u_{\alpha_{k_1}}(M) = 0$, $u_{\alpha_{k_2}}(M) = 0$, or both. Assume without loss of generality that $u_{\alpha_{k_1}}(M) = 0$. Since $u_{\alpha_{k_1}}(M') = 0$ it follows that $\alpha_{k_1} \neq \alpha_i$. Recall the d^{th} iteration of the while loop. Since $v_{S_{3d-1}}(\alpha_{k_1}) = 1$, $u_{\alpha_{k_1}}(M) = 0$, and $\alpha_{k_1} \neq \alpha_i$, it follows that, in that iteration, there existed some $\alpha_{z_1} \in N \setminus \{\alpha_i\}$, namely α_{k_1} , where $v_{\alpha_{z_1}}(S_{3d-1}) = 1$ and $u_{\alpha_{z_1}}(M) = 0$. In that iteration, since $\alpha_{z_1} \neq \perp$ the break condition must have held, the while loop terminated, and either the condition for Construction 1 was true or the condition for Construction 3 was true. This is a contradiction since by assumption the subroutine constructed M' according to Construction 7. It follows that $\alpha_g \neq S_{3d-1}$ for any d where $1 \leq d \leq c$.

Next, consider any S_{3d} where $1 \leq d < c$. By construction of M' it follows that $u_{S_{3d}}(M') = u_{S_{3d}}(M) = 1$ so $\alpha_g \neq S_{3d}$ for any such d .

The only possibility is thus that $\alpha_g = S_{3c}$. By the definition of α_g , there exists some triple $\{S_{3c}, \alpha_{k_1}, \alpha_{k_2}\}$ that blocks M' , where $\alpha_{k_1}, \alpha_{k_2} \in N$. Since $u_{S_{3c}}(M') = 0$ it must be that either $v_{S_{3c}}(\alpha_{k_1}) = 1$, $v_{S_{3c}}(\alpha_{k_2}) = 1$, or both.

Firstly, suppose that both $v_{S_{3c}}(\alpha_{k_1}) = 1$ and $v_{S_{3c}}(\alpha_{k_2}) = 1$ so $u_{S_{3c}}(\{\alpha_{k_1}, \alpha_{k_2}\}) = 2$. Since (N, V) is triangle-free, it must be that $v_{\alpha_{k_1}}(\alpha_{k_2}) = 0$ and thus that $u_{\alpha_{k_1}}(\{S_{3c}, \alpha_{k_2}\}) = u_{\alpha_{k_2}}(\{S_{3c}, \alpha_{k_1}\}) = 1$. Since $\{S_{3c}, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' it must be that $u_{\alpha_{k_1}}(M') = u_{\alpha_{k_2}}(M') = 0$. By the construction of M' it can be seen that S_{3c} is the only agent α_p in N where $u_{\alpha_p}(M') = 0$ and $u_{\alpha_p}(M') < u_{\alpha_p}(M)$. It follows that $u_{\alpha_{k_1}}(M) = u_{\alpha_{k_2}}(M) = 0$. Note that since $u_{\alpha_i}(M') = 1$ it follows that $\alpha_{k_1} \neq \alpha_i$ and $\alpha_{k_2} \neq \alpha_i$. It follows that either $\alpha_{k_1} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$, $\alpha_{k_2} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$, or both. Without loss of generality assume that $\alpha_{k_1} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$. In summary, after the termination of the while loop there existed some $\alpha_{z_2} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$, namely α_{k_1} , where $v_{\alpha_{z_2}}(S_{3c}) = 1$ and $u_{\alpha_{z_2}}(M) = 0$. Since $\alpha_{z_2} \neq \perp$ the condition of Construction 2 holds, which is a contradiction since, by assumption, the subroutine entered Construction 7.

Secondly, suppose either $v_{S_{3c}}(\alpha_{k_1}) = 1$ or $v_{S_{3c}}(\alpha_{k_2}) = 1$ but not both. Assume without loss of generality that $v_{S_{3c}}(\alpha_{k_1}) = 1$ and $v_{S_{3c}}(\alpha_{k_2}) = 0$. It follows that $u_{\alpha_{k_2}}(\{S_{3c}, \alpha_{k_1}\}) = 1$ and hence $u_{\alpha_{k_2}}(M') = 0$. Since S_{3c} is the only agent α_p in N where $u_{\alpha_p}(M') = 0$ and $u_{\alpha_p}(M') < u_{\alpha_p}(M)$, it follows that $u_{\alpha_{k_2}}(M) = 0$. It must be that $v_{\alpha_{k_1}}(\alpha_{k_2}) = 1$ since $u_{\alpha_{k_2}}(\{S_{3c}, \alpha_{k_1}\}) = 1$ and $v_{S_{3c}}(\alpha_{k_2}) = 0$. In summary, since $v_{S_{3c}}(\alpha_{k_1}) = 1$ and $v_{\alpha_{k_1}}(\alpha_{k_2}) = 1$ it follows that $u_{\alpha_{k_1}}(\{S_{3c}, \alpha_{k_2}\}) = 2$. It follows that either $u_{\alpha_{k_1}}(M') = 1$ or $u_{\alpha_{k_1}}(M') = 0$. Suppose firstly that $u_{\alpha_{k_1}}(M') = 0$. Since S_{3c} is the only agent α_p in N where $u_{\alpha_p}(M') = 0$ and $u_{\alpha_p}(M') < u_{\alpha_p}(M)$ it follows that $u_{\alpha_{k_1}}(M) = 0$. There are now two possibilities. Firstly, that $\alpha_{k_1} = \alpha_{j_2}$. Secondly, that $\alpha_{k_1} \neq \alpha_{j_2}$. In the first possibility, since $\alpha_{k_1} = \alpha_{j_2}$ then after the termination of the while loop there exists some $\alpha_{y_2} \in N$, namely α_{k_2} , where $v_{\alpha_{y_2}}(\alpha_{j_2}) = v_{\alpha_{y_2}}(\alpha_{j_2}) = 1$ and $u_{\alpha_{y_2}}(M) = 0$. In the algorithm, since $\alpha_{y_2} \neq \perp$ the condition of Construction 5 holds, which is a contradiction. In the second possibility, recall that $\alpha_{k_1} \neq \alpha_{j_2}$.

Since $u_{\alpha_i}(M') = 1$ it follows that $\alpha_i \neq \alpha_{k_1}$ and hence there exists some $\alpha_{z_2} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$, namely α_{k_1} , where $v_{\alpha_{z_2}}(S_{3c}) = 1$ and $u_{\alpha_{z_2}}(M) = 0$. It follows that after the termination of the while loop $\alpha_{z_2} \neq \perp$ and thus the condition of Construction 2 holds, which is a contradiction. It remains that $u_{\alpha_{k_1}}(M') = 1$. To recap, we have shown that $v_{S_{3c}}(\alpha_{k_1}) = v_{\alpha_{k_1}}(\alpha_{k_2}) = 1$, $u_{\alpha_{k_2}}(M') = u_{\alpha_{k_2}}(M) = 0$, and $u_{\alpha_{k_1}}(M') = 1$. This situation is illustrated in Figure 6.9.

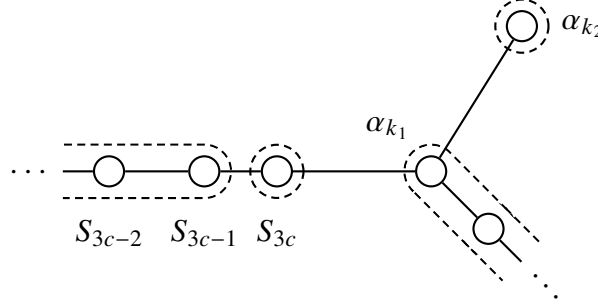


Figure 6.9: A hypothetical blocking triple in M' in Construction 7. In Lemma 6.6 we suppose for a contradiction that some triple $\{S_{3c}, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' where $\alpha_{k_1}, \alpha_{k_2} \in N$. We then show that $v_{S_{3c}}(\alpha_{k_1}) = v_{\alpha_{k_1}}(\alpha_{k_2}) = 1$, $u_{\alpha_{k_2}}(M') = u_{\alpha_{k_2}}(M) = 0$, and $u_{\alpha_{k_1}}(M') = 1$. We then show that this is a contradiction, and conclude that no such $\alpha_{k_1}, \alpha_{k_2}$ exist. This shows that S_{3c} does not belong to a triple that blocks M' .

By the condition of Construction 7, after the termination of the while loop it must have been that $\alpha_{w_1} = \perp$. By the pseudocode, it follows that in the last iteration there existed no $\alpha_{w_1} \in N$ where $v_{S_{3c}}(\alpha_{w_1}) = 1$, $u_{\alpha_{w_1}}(M) = 1$, $\alpha_{w_1} \notin S$, and that there existed some $\alpha_{z_3} \in N \setminus \{\alpha_i\}$ where $v_{\alpha_{z_3}}(\alpha_{w_1}) = 1$ and $u_{\alpha_{z_3}}(M) = 0$. If $u_{\alpha_{k_1}}(M) = 1$ and $\alpha_{k_1} \notin S$ then in the last iteration there existed some such α_{w_1} and α_{z_3} , namely α_{k_1} and α_{k_2} , which is a contradiction. It follows that either $u_{\alpha_{k_1}}(M) \neq 1$, $\alpha_{k_1} \in S$, or both.

Firstly suppose that $u_{\alpha_{k_1}}(M) \neq 1$. Recall that $u_{\alpha_{k_1}}(M') = 1$. By the construction of M' in Construction 7, $u_{\alpha_p}(M') = u_{\alpha_p}(M)$ for any $\alpha_p \in N \setminus (S \cup \{\alpha_i\})$. It follows that $\alpha_{k_1} \in S \cup \{\alpha_i\}$. By assumption, $\alpha_{k_1} \notin S$ so it must be that $\alpha_{k_1} = \alpha_i$. In this case, in the last iteration of the while loop there existed some $\alpha_{y_1} \in N$, namely α_{k_2} , where $v_{S_{3c}}(\alpha_i) = v_{\alpha_i}(\alpha_{y_1}) = 1$ and $u_{\alpha_{y_1}}(M) = 0$. It follows that, in the last iteration, the subroutine enters Construction 4, which is a contradiction.

It remains that $\alpha_{k_1} \in S$. Recall that $u_{\alpha_{k_1}}(M') = 1$. Since $u_{S_{3d-2}}(M') = 2$ for any d where $1 \leq d \leq c$ it follows that $\alpha_{k_1} \neq S_{3d-2}$ for any such d . It follows that either $\alpha_{k_1} = S_{3d-1}$ or $\alpha_{k_1} = S_{3d}$ for some d where $1 \leq d \leq c$.

Suppose that $\alpha_{k_1} = S_{3d-1}$ for some d where $1 \leq d \leq c$. Recall the d^{th} iteration of the while loop. We have shown that in that iteration, there existed some $\alpha_{z_1} \in N \setminus \{\alpha_i\}$, namely α_{k_2} , where $v_{\alpha_{z_1}}(S_{3d-1}) = 1$ and $u_{\alpha_{z_1}}(M) = 0$. It follows that after the d^{th} iteration of the while loop, $\alpha_{z_1} \neq \perp$ and thus that $d = c$ and the subroutine entered either Construction 1 or

Construction 3, which is a contradiction.

It now remains that $\alpha_{k_1} = S_{3d}$ for some d where $1 \leq d \leq c$. Recall that $S_{3c} \neq \alpha_{k_1}$ so $d < c$. Recall the d^{th} iteration of the while loop. Since $v_{S_{3d}}(\alpha_{k_2}) = 1$, $u_{\alpha_{k_2}}(M) = 0$, and $\alpha_{k_2} \neq \alpha_i$, and $u_{\alpha_i}(M') = 1$, it follows that in that iteration there existed some $\alpha_{z_2} \in N \setminus \{\alpha_i\}$, namely α_{k_2} , where $v_{\alpha_{z_2}}(S_{3d}) = 1$ and $u_{\alpha_{z_2}}(M) = 0$. There are two possibilities. The first is that $\alpha_{k_2} \neq \alpha_{j_2}$. The second is that $\alpha_{k_2} = \alpha_{j_2}$. Suppose first that $\alpha_{k_2} \neq \alpha_{j_2}$. In this case there existed some $\alpha_{z_2} \in N \setminus \{\alpha_i, \alpha_{j_2}\}$, namely α_{k_2} , where $v_{\alpha_{z_2}}(S_{3d}) = 1$ and $u_{\alpha_{z_2}}(M) = 0$. It follows that, in that iteration, $\alpha_{z_2} \neq \perp$ so the break condition held and the while loop terminated after that iteration. This is a contradiction since $d < c$. It remains that that $\alpha_{k_2} = \alpha_{j_2}$. It follows that, in that iteration, there existed some index b , namely d , where $1 \leq b < c$ and $v_{S_{3b}}(\alpha_{j_2}) = v_{S_{3c}}(S_{3b}) = 1$. It follows that, after the final iteration of the while loop, the condition for Construction 6 was true, which is a contradiction. \square

It is now relatively straightforward to show that M' must be stable.

Lemma 6.7. *Subroutine repair returns a stable P -matching M' .*

Proof. By Lemma 6.2 the subroutine must eventually terminate. By Lemma 6.3 the subroutine returns a P -matching.

Suppose M' is a P -matching returned by the algorithm. By Lemmas 6.4, 6.5, and 6.6, in Constructions 1–7, there exists no $\alpha_g \in N$ where $u_{\alpha_g}(M') < u_{\alpha_g}(M)$ and α_g belongs to a triple that blocks M' .

Suppose for a contradiction that M' is not stable and some triple $\{\alpha_{k_1}, \alpha_{k_2}, \alpha_{k_3}\}$ blocks M' . It follows that $u_{\alpha_{k_r}}(M') \geq u_{\alpha_{k_r}}(M)$ for $1 \leq r \leq 3$, otherwise some such α_g exists. By Proposition 6.1, it follows that $\{\alpha_{k_1}, \alpha_{k_2}, \alpha_{k_3}\}$ also blocks M . By the definition of repairable, any triple that blocks M must contain α_i so assume without loss of generality that $\alpha_{k_1} = \alpha_i$.

In Construction 4, $u_{\alpha_i}(M') = 2$ and thus α_i does not belong to a triple that blocks M' , which is a contradiction. It follows that M' is stable in Construction 4.

In Constructions 1, 2, 3, 5, 6, and 7, it must be that $u_{\alpha_i}(M') = 1$. It follows that $u_{\alpha_i}(\{\alpha_{k_2}, \alpha_{k_3}\}) = 2$ so $v_{\alpha_i}(\alpha_{k_2}) = v_{\alpha_i}(\alpha_{k_3}) = 1$. Since (N, V) is triangle-free, it must be that $v_{\alpha_{k_2}}(\alpha_{k_3}) = 0$ and thus that $u_{\alpha_{k_2}}(\{\alpha_i, \alpha_{k_3}\}) = u_{\alpha_{k_3}}(\{\alpha_i, \alpha_{k_2}\}) = 1$. Since $\{\alpha_i, \alpha_{k_2}, \alpha_{k_3}\}$ blocks M , It follows that $u_{\alpha_{k_2}}(M) = u_{\alpha_{k_3}}(M) = 0$ and thus that $\{\alpha_i, \alpha_{k_2}, \alpha_{k_3}\}$ also blocks M . It follows that M is not repairable, since there exists some triple $\{\alpha_i, \alpha_{k_2}, \alpha_{k_3}\}$ that blocks M where $\alpha_{k_2}, \alpha_{k_3} \in N$ and $u_{\alpha_{k_2}}(M) = 0, u_{\alpha_{k_3}}(M) = 0$. This is a contradiction, so it follows that M' is stable in Constructions 1, 2, 3, 5, 6, and 7. \square

Lemma 6.8. *Subroutine repair has running time $O(|N|^2)$.*

Proof. The pseudocode of Subroutine `repair`, shown in Algorithms 6.1 and 6.2, describes the subroutine at a high level. To analyse the worst-case time complexity we describe a suitable system of data structures, which we set up in a preprocessing step.

Suppose that (N, V) is stored such that, for a given $\alpha_p \in N$, the subroutine can iterate through the set $\{\alpha_q \in N : v_{\alpha_p}(\alpha_q) = 1\}$ in $O(|N|)$ time. Suppose that M is stored such that the subroutine can iterate through each triple in $O(|N|)$ time. For example, (N, V) could be stored as a graph using adjacency lists and M could be stored as a linked list. It follows that, given three agents $\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3} \in N$ the subroutine can compute $u_{\alpha_{h_1}}(\{\alpha_{h_2}, \alpha_{h_3}\})$, $u_{\alpha_{h_2}}(\{\alpha_{h_1}, \alpha_{h_3}\})$, and $u_{\alpha_{h_3}}(\{\alpha_{h_1}, \alpha_{h_2}\})$ in $O(|N|)$ time.

The preprocessing step involves constructing two lookup tables. Each lookup table contains exactly $|N|$ entries and is indexed by some $\alpha_p \in N$. Each entry in each table contains some integer less than or equal to $|N|$. It follows that finding an entry given its index requires constant time. Each entry in L_1 will contain either zero, one, or two. For each agent $\alpha_p \in N$, the subroutine constructs L_1 so that the p^{th} entry contains $u_{\alpha_p}(M)$. By assumption, the subroutine can compute $u_{\alpha_p}(M)$ for any $\alpha_p \in N$ in $O(|N|)$ time. It follows that L_1 can be constructed in $O(|N|^2)$ time by iterating through M and computing $u_{\alpha_{h_1}}(M)$, $u_{\alpha_{h_2}}(M)$, and $u_{\alpha_{h_3}}(M)$ for each triple $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\} \in M$. Since $|M| = O(|N|)$ this step takes $O(|N|^2)$ time. It follows that we can use L_1 to look up $u_{\alpha_p}(M)$ for any $\alpha_p \in N$ in constant time. Each entry in L_2 contains either the label of some agent or \perp . Construct L_2 such that for any $\alpha_p \in N$, the p^{th} entry either contains some $\alpha_q \in N \setminus \{\alpha_i\}$ where $v_{\alpha_p}(\alpha_q) = 1$ and $u_{\alpha_q}(M) = 0$ if it exists and otherwise \perp . The subroutine will use L_2 primarily in the body of the loop to identify α_{w_1} , if it exists, using S_{3c} . The lookup table L_2 can be constructed in $O(|N|^2)$ time, as follows. For each $\alpha_p \in N$, look up $u_{\alpha_p}(M)$ in L_1 . If $u_{\alpha_p}(M) = 0$ then consider each $\alpha_q \in N$ where $v_{\alpha_p}(\alpha_q) = 1$ and $\alpha_q \neq \alpha_i$. If the q^{th} entry of L_2 is currently \perp then set that entry to α_p .

The list S can be stored using a linked list or any data structure in which a new element can be appended to the end of S in constant time and iterating through S takes $O(|N|)$ time. The list S will be supplemented with a lookup table L_S . For any $\alpha_p \in N$, the table L_S can be used to test membership in S and look up the position of any agent in S in constant time. This is possible because the only modification that the subroutine makes to S is appending a single agent to the end of S in each iteration. As noted in Lemma 6.2, any agent is added to S at most than once. Like the tables L_1 and L_2 , the table L_S contains exactly $|N|$ entries and is indexed by each $\alpha_p \in N$. Each entry in L_S contains some integer position less than or equal to $|S|$. Before the subroutine appends an element $\alpha_p \in N$ to the end of S , it can maintain L_S in constant time by setting the p^{th} entry to $|S|$.

The first step in the subroutine involves identifying agents $\alpha_{j_1}, \alpha_{j_2}$ where $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$ blocks M in (N, V) and $u_{\alpha_{j_1}}(M) = 1$ as follows. Given any $\alpha_{j_1}, \alpha_{j_2} \in N$ where $u_{\alpha_{j_1}}(M) = 1$,

$u_{\alpha_{j_2}}(M) = 0$ and $v_{\alpha_i}(\alpha_{j_1}) = v_{\alpha_i}(\alpha_{j_2}) = 1$, the triple $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$ blocks M in (N, V) . It follows that some $\alpha_{j_1}, \alpha_{j_2} \in N$ can be found in $O(|N|)$ time, as follows. Consider each agent α_p for which $v_{\alpha_i}(\alpha_p) = 1$, and look up $u_{\alpha_p}(M)$ in L_1 . If $u_{\alpha_p}(M) = 1$ then look up the p^{th} entry of L_2 . By the construction of L_2 , if this entry is not equal to \perp then it contains some $\alpha_q \in N \setminus \{\alpha_i\}$ where $v_{\alpha_p}(\alpha_q) = 1$ and $u_{\alpha_q}(M) = 0$. In this case let $\alpha_{j_1} = \alpha_p$ and $\alpha_{j_2} = \alpha_q$. Since M is not stable in (N, V) , by the condition of M there must exist some such $\alpha_{j_1}, \alpha_{j_2}$.

The second step in the subroutine involves identifying agents $\alpha_{j_3}, \alpha_{j_4}$ where $\alpha_{j_3}, \alpha_{j_4} \in M(\alpha_{j_1}) \setminus \{\alpha_{j_1}\}$ and $u_{\alpha_{j_3}}(M) = 2$. This can be done in $O(|N|)$ time, as follows. Consider each triple in M until $M(\alpha_{j_1})$ is found. This takes $O(|N|)$ time. Use L_1 to identify α_{j_3} and α_{j_4} .

The initialisation of $S, c, \alpha_{z_1}, \alpha_{z_2}, \alpha_{y_1}, \alpha_{y_2}$ and α_{w_1} in the subroutine takes constant time.

Consider the while loop. By Lemma 6.2, there are at most $\lfloor (|N| - 2) / 3 \rfloor = O(|N|)$ iterations. Setting up the lookup tables allows us to ensure that each iteration takes $O(|N|)$ time. It follows that the loop terminates in $O(|N|^2)$ time.

To identify α_{z_1} as described, first identify S_{3c-1} , in constant time. Consider each $\alpha_p \in N$ for which $v_{S_{3c-1}}(\alpha_p) = 1$. This takes $O(|N|)$ time. For each such α_p , if $\alpha_p = \alpha_i$ then continue. If $\alpha_p \neq \alpha_i$ then look up $u_{\alpha_p}(M)$ in L_1 . If $u_{\alpha_p}(M) = 0$ then set $\alpha_{z_1} = \alpha_p$. If no such α_p is found then no such α_{z_1} exists so set $\alpha_{z_1} = \perp$.

Similarly, to identify some α_{z_2} as described, first identify S_{3c} . Consider each $\alpha_{l_1} \in N$ for which $v_{S_{3c}}(\alpha_{l_1}) = 1$. This takes $O(|N|)$ time. For each such α_{l_1} , if $\alpha_{l_1} = \alpha_i$ or $\alpha_{l_1} = \alpha_{j_2}$ then continue. If not, look up $u_{\alpha_{l_1}}(M)$ in L_1 . If $u_{\alpha_{l_1}}(M) = 0$ then set $\alpha_{z_2} = \alpha_{l_1}$. If no such α_p is found then no such α_{z_2} exists so set $\alpha_{z_2} = \perp$.

To identify α_{y_1} as described, test if $v_{S_{3c}}(\alpha_i) = 1$. This takes $O(|N|)$ time. If $v_{S_{3c}}(\alpha_i) = 0$ then no such α_{y_1} exists. If $v_{S_{3c}}(\alpha_i) = 1$ then consider each $\alpha_p \in N$ for which $v_{\alpha_i}(\alpha_p) = 1$. Note that $\alpha_p \neq \alpha_{j_2}$ since otherwise $v_{\alpha_{j_2}}(\alpha_i) = 1$, from which it follows that $\{\alpha_i, \alpha_{j_1}, \alpha_{j_2}\}$ is a triangle in (N, V) . Look up $u_{\alpha_p}(M)$ in L_1 . If $u_{\alpha_p}(M) = 0$ then set $\alpha_{y_1} = \alpha_p$. If no such α_p where $u_{\alpha_{l_1}}(M) = 0$ is found then no such α_{y_1} exists so set $\alpha_{y_1} = \perp$. The identification of α_{y_2} , if it exists, can be performed similarly in $O(|N|)$ time.

To compute $1 \leq b < c$ as described, if there exists some such S_{3b} where $v_{S_{3b}}(\alpha_{j_2}) = v_{S_{3c}}(S_{3b}) = 1$, consider each $\alpha_p \in N$ for which $v_{S_{3c}}(\alpha_p) = 1$. This takes $O(|N|)$ time. For each such α_p , determine its position b' in S if it belongs to S . If α_p belongs to S and b' is divisible by three and less than c then set $b = b'$. Otherwise, it must be that no such S_{3b} exists so set $b = 0$.

To identify some α_{w_1} as described, first identify S_{3c} in constant time. Consider each $\alpha_p \in N$ for which $v_{S_{3c}}(\alpha_p) = 1$. This takes $O(|N|)$ time. For each such α_p , test if α_p belongs to S using L_S . If so, then continue. If not, then look up the p^{th} entry in L_2 . If this entry is \perp then continue. If not, then suppose this entry is α_q . By the construction of L_2 , it follows that

$\alpha_q \in N \setminus \{\alpha_i\}$, $v_{\alpha_p}(\alpha_q) = 1$ and $u_{\alpha_q}(M) = 0$. Accordingly, set α_{w_1} to α_p since the subroutine has identified $\alpha_{z_3} = \alpha_q \in N \setminus \{\alpha_i\}$ for which $v_{\alpha_{w_1}}(\alpha_{z_3}) = 1$ and $u_{\alpha_{z_3}}(M) = 0$.

Evaluating the break condition in the loop can be performed in constant time. If the break condition is true then α_{w_1} exists. The identification of α_{w_2} and α_{w_3} can be accomplished in $O(|N|)$ time, using the same process as for α_{j_3} and α_{j_4} . Adding three elements to S requires constant time.

Now consider the final if/else statement and the seven possible constructions of M' . In each of the seven cases, M' contains each triple in $\{r \in M : S \cap r = \emptyset\}$. This set can be constructed in $O(|N|)$ time by considering each triple in M and the three corresponding entries in L_S . In Constructions 3 and 6, the agents α_{z_4} and α_{z_5} can each be identified in $O(|N|)$ time, using a similar procedure as described for α_{z_1} in the loop body. The remaining triples in M' can be constructed after one scan of S in $O(|N|)$ time. \square

Lemma 6.9. *Subroutine repair returns a stable P -matching in $O(|N|^2)$ time.*

Proof. By Lemmas 6.7 and 6.8. \square

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

In the previous section, we presented Subroutine repair, which given a triangle-free instance and a repairable P -matching in that instance, can construct a P -matching in that instance that is stable. In this section we present Subroutine findStableInTriangleFree, which calls Subroutine repair. Given a triangle-free instance, Subroutine findStableInTriangleFree can find a P -matching in that instance that is stable.

Subroutine findStableInTriangleFree, shown in Algorithm 6.3, is recursive. The subroutine first constructs a smaller instance (N', V') from (N, V) by removing an arbitrary agent α_i . It then uses a recursive call to construct a P -matching M that is stable in the smaller instance (N', V') . By Proposition 6.1, any triple that blocks M in the larger instance (N, V) must either contain α_i or also block M in the smaller instance (N', V') . There are three possible cases involving types of triple that block M in (N', V') . In two out of three cases, the subroutine constructs M' in a straightforward way by adding to M a new triple that contains α_i and two agents that are unmatched in M . In the third case, M must be repairable. It follows by Lemma 6.9 that Subroutine repair can be used to construct a P -matching that is stable in (N, V) .

In the following lemma we verify the correctness of Subroutine findStableInTriangleFree.

Lemma 6.10. *Given a triangle-free instance (N, V) , Subroutine findStableInTriangleFree returns a stable P -matching in (N, V) .*

Algorithm 6.3 Subroutine findStableInTriangleFree

Input: an instance (N, V) of 3DR-AS with binary and symmetric preferences that is triangle-free

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

```

if  $|N| = 2$  then
  return  $\emptyset$ 
end if

 $\alpha_i \leftarrow$  an arbitrary agent in  $N$ 
 $(N', V') \leftarrow$  remove  $\alpha_i$  from  $(N, V)$ 
 $M \leftarrow$  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  $v_{\alpha_i}(\alpha_{l_1}) = v_{\alpha_i}(\alpha_{l_2}) = 1$  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
   $v_{\alpha_i}(\alpha_{l_3}) = v_{\alpha_i}(\alpha_{l_4}) = 1$  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  $v_{\alpha_i}(\alpha_{l_5}) = v_{\alpha_i}(\alpha_{l_6}) = 1$  then
   $\triangleright$   $M$  is repairable in  $(N, V)$  (see Section 6.3.1.3). 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

```

Proof. By strong induction on $|N|$. In the base case, suppose $|N| \leq 2$. It follows by the pseudocode that the subroutine returns \emptyset , which is a stable P -matching in (N, V) .

We now show the inductive step. Consider the execution of the subroutine given some an arbitrary instance (N, V) . By the inductive hypothesis it follows that the subroutine returns a stable P -matching M in the smaller instance (N', V') , since $|N'| < |N|$.

Consider the first branch of the if/else statement in Subroutine findStableInTriangleFree. By construction, $u_{\alpha_i}(M') = 2$ and $u_{\alpha_{l_1}}(M') = u_{\alpha_{l_2}}(M') = 1$. Since M is a P -matching, it follows that the subroutine returns some P -matching M' . Since $u_{\alpha_i}(M') = 2$, no triple that contains α_i blocks M' in (N, V) . By construction, $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N$, so it follows that any triple that blocks M' in (N, V) also blocks M in (N', V') . It follows that M' is stable in (N, V) .

Consider the second branch of the if/else statement. By construction, $u_{\alpha_{l_3}}(M') = 2$ and $u_{\alpha_i}(M') = u_{\alpha_{l_4}}(M') = 1$. Since M is a P -matching, it follows that the subroutine returns some P -matching M' . Suppose for a contradiction that some triple blocks M' in (N, V) . By construction, $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N$ so any triple that blocks M' in (N, V) must contain α_i , for otherwise that triple blocks M in (N', V') , which is a contradiction. Suppose then that some triple $\{\alpha_i, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' in (N, V) , where $\alpha_{k_1}, \alpha_{k_2} \in N'$. By construction, $u_{\alpha_i}(M') = 1$ so it must be that $u_{\alpha_i}(\{\alpha_{k_1}, \alpha_{k_2}\}) = 2$ and thus that $v_{\alpha_i}(\alpha_{k_1}) = v_{\alpha_i}(\alpha_{k_2}) = 1$.

Since (N, V) is triangle-free, it follows that $v_{\alpha_{k_1}}(\alpha_{k_2}) = 0$ and thus that $u_{\alpha_{k_1}}(\{\alpha_i, \alpha_{k_2}\}) = u_{\alpha_{k_2}}(\{\alpha_i, \alpha_{k_1}\}) = 1$. It then follows that $u_{\alpha_{k_1}}(M') = u_{\alpha_{k_2}}(M') = 0$. By the construction of M' it must be that $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N$ so $u_{\alpha_{k_1}}(M) = u_{\alpha_{k_2}}(M) = 0$. This contradicts the condition of the first branch of the if/else statement, since there exist two agents α_{l_1} and α_{l_2} , namely α_{k_1} , and α_{k_2} , where $u_{\alpha_{l_1}}(M) = u_{\alpha_{l_2}}(M) = 0$ and $v_{\alpha_i}(\alpha_{l_1}) = v_{\alpha_i}(\alpha_{l_2}) = 1$.

Consider the third branch of the if/else statement. It must be that the conditional expressions in the first and second branches of the if/else statement do not hold. It follows from this that any triple that blocks M in (N', V') comprises $\{\alpha_i, \alpha_{l_5}, \alpha_{l_6}\}$ where $\alpha_{l_5}, \alpha_{l_6} \in N$, $u_{\alpha_{l_5}}(M) = 1$, $u_{\alpha_{l_6}}(M) = 0$, and $v_{\alpha_i}(\alpha_{l_5}) = v_{\alpha_i}(\alpha_{l_6}) = 1$. Note that $u_{\alpha_i}(M) = 0$ and thus M is repairable (defined in Section 6.3.1.3). By Lemma 6.9, Subroutine `repair` returns some P -matching M' that is stable in (N, V) .

Consider the fourth branch of the if/else statement. It must be that the conditional expressions in the first, second, and third branches of the if/else statement do not hold. By construction, $u_{\alpha_p}(M') = u_{\alpha_p}(M)$ for any $\alpha_p \in N$. It follows that any triple that blocks M' in (N, V) must contain α_i , for otherwise that triple also blocks M in (N', V') , which is a contradiction. Suppose for a contradiction that some triple $\{\alpha_i, \alpha_{k_1}, \alpha_{k_2}\}$ blocks M' in (N, V) .

Suppose firstly that $u_{\alpha_i}(\{\alpha_{k_1}, \alpha_{k_2}\}) = 2$. Since (N, V) is triangle-free, it follows that $u_{\alpha_{k_1}}(\{\alpha_i, \alpha_{k_2}\}) = u_{\alpha_{k_2}}(\{\alpha_i, \alpha_{k_1}\}) = 1$. It follows that $u_{\alpha_{k_1}}(M') = u_{\alpha_{k_2}}(M') = 0$. Since $u_{\alpha_p}(M') \geq u_{\alpha_p}(M)$ for any $\alpha_p \in N$, it must be that $u_{\alpha_{k_1}}(M) = u_{\alpha_{k_2}}(M) = 0$. This contradicts the condition of the first branch of the if/else statement.

Suppose secondly that $u_{\alpha_i}(\{\alpha_{k_1}, \alpha_{k_2}\}) = 1$. It must be that either $u_{\alpha_{k_1}}(\{\alpha_i, \alpha_{k_2}\}) = 2$ or $u_{\alpha_{k_2}}(\{\alpha_i, \alpha_{k_1}\}) = 2$. Suppose without loss of generality that $u_{\alpha_{k_1}}(\{\alpha_i, \alpha_{k_2}\}) = 2$. It follows that $v_{\alpha_{k_1}}(\alpha_i) = v_{\alpha_{k_1}}(\alpha_{k_2}) = 1$. There are two possibilities: either $u_{\alpha_{k_1}}(M) = 1$ or $u_{\alpha_{k_1}}(M) = 0$. The first possibility implies that the conditional expression of the second if/else branch holds, which is a contradiction. The second possibility implies that the conditional expression of the third if/else branch holds, also a contradiction. \square

We now consider the worst-case time complexity of Subroutine `findStableInTriangleFree`.

Lemma 6.11. *Subroutine `findStableInTriangleFree` has running time $O(|N|^3)$.*

Proof. Since Subroutine `findStableInTriangleFree` is recursive, and the recursive call involves an instance (N', V') where $|N'| = |N| - 1$, it suffices to show that worst-case time complexity of the subroutine excluding the recursive call is $O(|N|^2)$.

Suppose that the input (N, V) is given such that, for a given $\alpha_p \in N$, the subroutine can iterate through the set $\{\alpha_q \in N : v_{\alpha_p}(\alpha_q) = 1\}$ in $O(|N|)$ time. For example, (N, V) could be stored as a graph using adjacency lists. It follows that, given three agents $\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3} \in N$ the subroutine can compute $u_{\alpha_{h_1}}(\{\alpha_{h_2}, \alpha_{h_3}\})$, $u_{\alpha_{h_2}}(\{\alpha_{h_1}, \alpha_{h_3}\})$ and $u_{\alpha_{h_3}}(\{\alpha_{h_1}, \alpha_{h_2}\})$ in $O(|N|)$

time. The subroutine will return a P -matching M' stored as a linked list or similar data structure that allows a triple to be appended to the end of list in constant time.

The constructed instance (N', V') can be stored using adjacency lists or an equivalent data structure. A straightforward procedure to identify α_i and construct (N', V') takes $O(|N|)$ time.

After this call, the subroutine can construct a supplementary lookup table L_1 , with exactly $|N| - 1$ entries indexed by each $\alpha_p \in N'$. Each entry will contain either zero, one, or two. For each agent $\alpha_p \in N$, the subroutine constructs L_1 so that the p^{th} entry contains $u_{\alpha_p}(M)$. By assumption, the subroutine can compute $u_{\alpha_p}(M)$ for any $\alpha_p \in N$ in $O(|N|)$ time. It follows that L_1 can be constructed in $O(|N|^2)$ time by iterating through M and computing $u_{\alpha_{h_1}}(M)$, $u_{\alpha_{h_2}}(M)$, and $u_{\alpha_{h_3}}(M)$ for each triple $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\} \in M$. Since $|M| = O(|N|)$ this step takes $O(|N|^2)$ time. It follows that we can use L_1 to look up $u_{\alpha_p}(M)$ for any $\alpha_p \in N$ in constant time.

The construction of L_1 allows the subroutine to identify some α_{l_1} and $\alpha_{l_2} \in N$ where $u_{\alpha_{l_1}}(M) = u_{\alpha_{l_2}}(M) = 0$ and $v_{\alpha_i}(\alpha_{l_1}) = v_{\alpha_i}(\alpha_{l_2}) = 1$, if two such agents exist, in $O(|N|^2)$ time. One way to do this is to consider each pair $\{\alpha_{l_1}, \alpha_{l_2}\} \in \binom{N}{2}$ and look up $u_{\alpha_{l_1}}(M)$ and $u_{\alpha_{l_2}}(M)$ in L_1 . Since M is stored using a linked list or similar data structure, if such $\alpha_{l_1}, \alpha_{l_2} \in N$ exist then M' can be constructed by adding the triple $\{\alpha_i, \alpha_{l_1}, \alpha_{l_2}\}$ to M , in constant time. Similarly, the identification of $\alpha_{l_3}, \alpha_{l_4} \in N$ where $u_{\alpha_{l_3}}(M) = u_{\alpha_{l_4}}(M) = 0$ and $v_{\alpha_i}(\alpha_{l_3}) = v_{\alpha_i}(\alpha_{l_4}) = 1$ can be performed in $O(|N|^2)$ time and the corresponding construction of M' in constant time. In the third branch of the if/else statement, the identification of $\alpha_{l_5}, \alpha_{l_6} \in N$ where $u_{\alpha_{l_5}}(M) = 1, u_{\alpha_{l_6}}(M) = 0$ and $v_{\alpha_i}(\alpha_{l_5}) = v_{\alpha_i}(\alpha_{l_6}) = 1$ can be similarly performed in $O(|N|^2)$ time. By Lemma 6.8, the call to Subroutine repair also takes $O(|N|^2)$ time. \square

6.3.1.5 Finding a stable P -matching in an arbitrary instance

In the previous section, we presented Subroutine findStableInTriangleFree, which given a triangle-free instance can construct a P -matching in that instance that is stable. In this section we present Algorithm findStable, which given an arbitrary instance of 3DR-AS can construct a P -matching that is stable in that instance.

Algorithm findStable involves two steps, as follows. In the first step, it constructs a P -matching M_Δ in (N, V) and a triangle-free instance (N', V') where $|N'| \leq |N|$ and if M is a stable P -matching in (N', V') then $M_\Delta \cup M$ is a stable P -matching in (N, V) . By Lemma 6.1, this can be done in $O(|N|^3)$ time. In the second step, it calls Subroutine findStableInTriangleFree on (N', V') to construct a P -matching M' that is stable in (N', V') . It then returns $M' = M \cup M_\Delta$.

Theorem 6.1. *Given an instance (N, V) of 3DR-AS with binary and symmetric preferences, a stable P -matching, and hence a stable matching, must exist and can be found in $O(|N|^3)$ time.*

Proof. By Lemma 6.1, the first step of the algorithm takes $O(|N|^3)$ time and (N', V') is a triangle-free instance where $|N'| \leq |N|$ and if M is a stable P -matching in (N', V') then $M_\Delta \cup M$ is a stable P -matching in (N, V) .

By Lemma 6.10, the P -matching M returned by the call to Subroutine `findStableInTriangleFree` is stable in (N', V') . By Lemma 6.11, this call, which constitutes the second step of Algorithm `findStable`, takes $O(|N|^3)$ time. It follows that $M' = M_\Delta \cup M$ is a stable P -matching in (N, V) and the worst-case time complexity of Algorithm `findStable` is $O(|N|^3)$. \square

6.3.2 Maximising utilitarian welfare

We have shown that in an instance of 3DR-AS with binary and symmetric preferences, a stable matching must exist and can be found in polynomial time. In this section we consider a related optimisation problem, in which the goal is to find a stable matching with maximum utilitarian welfare given an instance of 3DR-AS with binary and symmetric preferences. We first show that this problem is NP-hard and then extend Algorithm `findStable` to devise a 2-approximation algorithm.

We formalise this optimisation problem as the *3DR-AS Stable Maximum Utilitarian Welfare problem* (3DR-AS-SMUW). It is straightforward to show that 3DR-AS-SMUW is NP-hard, as follows.

Theorem 6.2. *3DR-AS-SMUW is NP-hard.*

Proof. A direct reduction exists from PIT to the problem of deciding if a given instance of 3DR-AS-SMUW contains a stable matching M with utilitarian welfare greater than or equal to a given bound, as follows. Suppose $G = (W, E)$ is an arbitrary undirected graph. First, for each vertex w_i in W construct one agent α_i in N . Next, for any two agents $\alpha_i, \alpha_j \in N$, let $v_{\alpha_i}(\alpha_j) = 1$ if $\{w_i, w_j\} \in E$ and 0 otherwise. It is straightforward to show that (N, V) contains a stable matching with utilitarian welfare $2|W|$ if and only if G contains a partition into triangles. \square

Note that the reduction from PIT to 3DR-AS-SMUW also shows that the problem of finding a (not-necessarily stable) matching with maximum utilitarian welfare in a given instance of 3DR-AS is also NP-hard, even when preferences are binary and symmetric.

We now present an approximation algorithm for 3DR-AS-SMUW, which we call Algorithm `findStableUW`, shown in Algorithm 6.4. We first provide some intuition regarding its design and then prove that it is correct and analyse its approximation ratio.

At a high level, Algorithm `findStableUW` involves two phases. In the first phase, it calls Algorithm `findStable` to construct a stable P -matching M_1 . In the second phase, it orders

Algorithm 6.4 Algorithm findStableUW**Input:** an instance (N, V) of 3DR-AS with binary and symmetric preferences**Output:** a stable matching M in (N, V)

```

 $M_1 \leftarrow \text{findStable}((N, V))$ 
 $U \leftarrow \text{agents in } N \text{ unmatched in } M_1$ 
 $Q \leftarrow \text{maximal2DMatching}((N, V), U)$ 
if  $|Q| \geq |U|/3$  then
     $X \leftarrow \text{any } |U|/3 \text{ elements of } Q$ 
else
    ▷ note that  $|U \setminus \bigcup Q| > 2(|U|/3 - |Q|)$  since  $|U \setminus \bigcup Q| = |U| - 2|Q|$ 
     $W \leftarrow |U|/3 - |Q|$  pairs of agents chosen from the set of agents  $U \setminus \bigcup Q$ 
     $X \leftarrow Q \cup W$ 
end if
 $Y \leftarrow U \setminus \bigcup X$ 
    ▷ Suppose  $X = \{X_1, X_2, \dots, X_{|U|/3}\}$  and  $Y = \{y_1, y_2, \dots, y_{|U|/3}\}$ . Note that  $X$  is a set
      of pairs of agents and  $Y$  is a set of individual agents.
     $M_2 \leftarrow \{X_i \cup \{y_i\} : 1 \leq i \leq |U|/3\}$ 
return  $M_1 \cup M_2$ 

```

the unmatched agents U in M_1 into triples such that utilitarian welfare of the agents in U is maximised. In order to do this, it constructs a maximal matching in the subgraph induced by U and then orders the agents in U to triples such that the number of triples that contain an edge in the maximal matching is maximised.

It is straightforward to show that Algorithm findStableUW returns a matching M in polynomial time. We now analyse its approximation ratio.

Suppose (N, V) is an arbitrary instance of 3DR-AS with binary and symmetric preferences, M is a matching returned by Algorithm findStableUW given (N, V) and M^* is a stable matching with maximum utilitarian welfare in (N, V) .

At a high level, the analysis involves placing a lower bound on the welfare in M of the agents in each triple apportioned by the triples in M^* . To do this, let $T(y)$ be the triples in M with utilitarian welfare y and $T^*(y)$ be the triples in M^* with utilitarian welfare y , for some $y \geq 0$. In fact, since preferences are binary and symmetric it must be that the utilitarian welfare of any triple in (N, V) must be either 0, 2, 4, or 6. Thus, by definition

$$M = T(6) \cup T(4) \cup T(2) \cup T(0) \tag{6.1}$$

and

$$M^* = T^*(6) \cup T^*(4) \cup T^*(2) \cup T^*(0) . \tag{6.2}$$

It follows that

$$u(M) = 6|T(6)| + 4|T(4)| + 2|T(2)| \quad \text{by Equation 6.1} \quad (6.3)$$

and

$$u(M^*) = 6|T^*(6)| + 4|T^*(4)| + 2|T^*(2)| \quad \text{by Equation 6.2.} \quad (6.4)$$

We first place a lower bound on $|T(6)|$ in terms of $|T^*(6)|$.

Lemma 6.12. $|T(6)| \geq |T^*(6)|/3$.

Proof. By the pseudocode of Algorithm findStable and the definition of Subroutine eliminateTriangles, $T(6)$ contains a maximal triangle packing in the underlying graph (N, E) . Since by definition $T^*(6)$ is also a maximal triangle packing in the same graph, it must be that $|T(6)| \geq |T^*(6)|/3$. \square

We now show that if no triple in M has utilitarian welfare 0 then $2u(M) \geq u(M^*)$.

Lemma 6.13. *If $T(0) = \emptyset$ then $2u(M) \geq u(M^*)$.*

Proof. First consider M^* . Now

$$\begin{aligned}
 2u(M) &= 12|T(6)| + 8|T(4)| + 4|T(2)| && \text{by Equation 6.3} \\
 &\geq 12|T(6)| + 4(|T(4)| + |T(2)|) \\
 &= 12|T(6)| + 4(|M| - |T(6)|) && \text{by Equation 6.1, since } T(0) = \emptyset \\
 &= 12|T(6)| + 4(n - |T(6)|) && \text{by definition, } |M| = n \\
 &= 8|T(6)| + 4n \\
 &\geq \frac{8|T^*(6)|}{3} + 4n && \text{by Lemma 6.12} \\
 &\geq 2|T^*(6)| + 4n \\
 &= 6|T^*(6)| - 4|T^*(6)| + 4n \\
 &= 6|T^*(6)| + 4(n - |T^*(6)|) \\
 &= 6|T^*(6)| + 4(|M^*| - |T^*(6)|) \\
 &= 6|T^*(6)| + 4(|T^*(4)| + |T^*(2)| + |T^*(0)|) && \text{by Equation 6.2} \\
 &= 6|T^*(6)| + 4|T^*(4)| + 4|T^*(2)| && \text{since } u(T^*(0)) = 0 \\
 &\geq 6|T^*(6)| + 4|T^*(4)| + 2|T^*(2)| \\
 &= u(M^*) && \text{by Equation 6.4.}
 \end{aligned}$$

\square

We now consider the case when there exists at least one triple in $T(0)$. The existence of such a triple allows us to deduce that $|Q| < |U|/3$ and thus that any two agents that form an edge in the maximal matching Q must be assigned to the same triple in M .

Lemma 6.14. *If $|T(0)| > 0$ then $|Q| < |U|/3$.*

Proof. We prove the contrapositive. Suppose $|Q| \geq |U|/3$. By the pseudocode of Algorithm findStableUW, $X \subseteq Q$ is a set of pairs where $v_{x_i}(x_j) = 1$ for each pair $\{x_i, x_j\} \in X$. It follows that each triple in M_2 contains two agents x_i, x_j for which $v_{x_i}(x_j) = 1$. It follows that $u_t(M) \geq 2$ for any triple $t \in M_2$. Since M_1 is a P -matching, by definition $u_t(M) \geq 2$ for any $t \in M_1$ so it must be that $|T(0)| = \emptyset$. \square

Lemma 6.15. *If $|T(0)| > 0$ then $u_{\alpha_p}(M) \geq 1$ for any $\alpha_p \in \bigcup Q$.*

Proof. Suppose $|T^*(0)| > 0$. Consider an arbitrary $\alpha_p \in \bigcup Q$. It follows that some $\alpha_q \in N$ exists where $\{\alpha_p, \alpha_q\} \in Q$ and hence $v_{\alpha_p}(\alpha_q) = 1$, by the definition of Q .

By Lemma 6.14, $|Q| < |U|/3$. It follows that $\{\alpha_p, \alpha_q\} \in X$. It follows that there exists some i where $1 \leq i \leq |U|/3$ such that $X_i = \{\alpha_p, \alpha_q\}$ and hence, by construction of M_2 , the triple $X_i \cup \{y_i\}$ belongs to M_2 . It follows that $\alpha_q \in M_2(\alpha_p)$ and hence $u_{\alpha_p}(M) \geq 1$. \square

Lemma 6.16. *If $|T(0)| > 0$ then for any $\alpha_r, \alpha_s \in N$ where $v_{\alpha_r}(\alpha_s) = 1$ it must be that $u_{\{\alpha_r, \alpha_s\}}(M) \geq 1$.*

Proof. Suppose for a contradiction that $|T(0)| > 0$ and that there exists some $\alpha_r, \alpha_s \in N$ where $v_{\alpha_r}(\alpha_s) = 1$ and $u_{\{\alpha_r, \alpha_s\}}(M) = 0$. It follows that $u_{\alpha_r}(M) = u_{\alpha_s}(M) = 0$. It follows by Lemma 6.15 that $\alpha_r \notin \bigcup Q$ and $\alpha_s \notin \bigcup Q$. It follows that $Q' = Q \cup \{\alpha_r, \alpha_s\}$ is a disjoint set of pairs of agents in U where $v_{\alpha_p}(\alpha_q) = 1$ for each pair $\{\alpha_p, \alpha_q\} \in Q'$. Since $|Q'| > |Q|$, this contradicts the maximality of Q (which is computed in Subroutine maximal2DMatching). \square

Lemma 6.17. *If $|T(0)| > 0$ then $u_t(M) \geq 3$ for any $t \in T^*(6)$.*

Proof. Suppose $|T(0)| > 0$. Consider an arbitrary $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\} \in T^*(6)$. By definition, $v_{\alpha_{h_1}}(\alpha_{h_2}) = v_{\alpha_{h_2}}(\alpha_{h_3}) = v_{\alpha_{h_3}}(\alpha_{h_1}) = 1$. Since M is a stable matching, the triple $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}$ does not block M . It follows that at least one of the following holds: $u_{\alpha_{h_1}}(M) = 2$, $u_{\alpha_{h_2}}(M) = 2$, or $u_{\alpha_{h_3}}(M) = 2$. Suppose without loss of generality that $u_{\alpha_{h_1}}(M) = 2$. By Lemma 6.16, it must be that $u_{\{\alpha_{h_2}, \alpha_{h_3}\}}(M) \geq 1$. In total, $u_{\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}}(M) \geq 3$. \square

Lemma 6.18. *If $|T(0)| > 0$ then $u_t(M) \geq 2$ for any $t \in T^*(4)$.*

Proof. Suppose $|T(0)| > 0$. Consider an arbitrary $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\} \in T^*(4)$ where $v_{\alpha_{h_1}}(\alpha_{h_2}) = v_{\alpha_{h_2}}(\alpha_{h_3}) = 1$ and $v_{\alpha_{h_1}}(\alpha_{h_3}) = 0$. Suppose for a contradiction that $u_{\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}}(M) < 2$.

If $u_{\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}}(M) = 0$, then $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}$ blocks M in (N, V) . It must be that $u_{\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}}(M) = 1$. By Lemma 6.16, it must be that $u_{\{\alpha_{h_1}, \alpha_{h_2}\}}(M) \geq 1$ and also that $u_{\{\alpha_{h_2}, \alpha_{h_3}\}}(M) \geq 1$. It follows that $u_{\alpha_{h_1}}(M) = u_{\alpha_{h_3}}(M) = 0$ and $u_{\alpha_{h_2}}(M) = 1$. In this case, $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}$ blocks M in (N, V) , which is a contradiction. It follows that $u_{\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}}(M) \geq 2$. \square

Lemma 6.19. *If $|T(0)| > 0$ then $u_t(M) \geq 1$ for any $t \in T^*(2)$.*

Proof. Suppose $|T(0)| > 0$. Consider an arbitrary $\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\} \in T^*(2)$ where $v_{\alpha_{h_1}}(\alpha_{h_2}) = 1$ and $v_{\alpha_{h_1}}(\alpha_{h_3}) = v_{\alpha_{h_2}}(\alpha_{h_3}) = 0$. By Lemma 6.16, it must be that $u_{\{\alpha_{h_1}, \alpha_{h_2}\}}(M) \geq 1$ and hence $u_{\{\alpha_{h_1}, \alpha_{h_2}, \alpha_{h_3}\}}(M) \geq 1$. \square

We can now combine Lemmas 6.17, 6.18, and 6.19 to prove the approximation ratio of Algorithm findStableUW.

Lemma 6.20. *The approximation ratio of Algorithm findStableUW is 2.*

Proof. If $|T(0)| = 0$ then by Lemma 6.13 it must be that $2u(M) \geq u(M^*)$. It remains to consider the case in which $|T(0)| > 0$. In this case,

$$\begin{aligned}
 2u(M) &= 2 \sum_{t \in M^*} u_t(M) \\
 &= 2 \left(\sum_{t \in T^*(6)} u_t(M) + \sum_{t \in T^*(4)} u_t(M) + \sum_{t \in T^*(2)} u_t(M) + \sum_{t \in T^*(0)} u_t(M) \right) \text{ by the definition of } T^* \\
 &\geq 2 \sum_{t \in T^*(6)} u_t(M) + 2 \sum_{t \in T^*(4)} u_t(M) + 2 \sum_{t \in T^*(2)} u_t(M) \\
 &\geq 6|T^*(6)| + 4|T^*(4)| + 2|T^*(2)| \quad \text{by Lemmas 6.17–6.19} \\
 &= u(M^*) .
 \end{aligned}$$

\square

Theorem 6.3. *There exists a polynomial-time 2-approximation algorithm for 3DR-AS-SMUW.*

Proof. It is straightforward to show that Algorithm findStableUW returns a matching in polynomial time. In Lemma 6.20 we show that the approximation ratio of this algorithm is 2. \square

Recall that in Section 6.3.1.5, we showed that a stable matching exists in an instance of 3DR-AS even if the number of agents is not divisible by three and the definition of a matching allows for at most two agents to be unmatched. We remark that it is straightforward to adapt

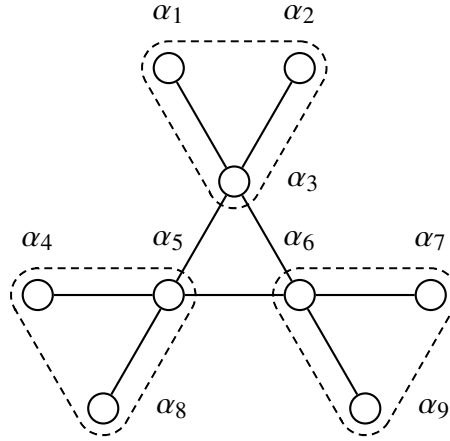


Figure 6.10: An instance of 3DR-AS-SMUW in which $u(M^*) = 2u(M)$. The dashed enclosure depicts M^* .

the proof in this section to show that Algorithm `findStableUW` has the same approximation ratio even if the number of agents is not divisible by three and the definition of a matching allows for agents to be unmatched.

It is straightforward to show that the analysis of Algorithm `findStableUW` is tight, which we do as follows. Consider the instance of 3DR-AS shown in Figure 6.10, which has binary and symmetric preferences. Algorithm `findStableUW` is bound to return $M = \{\{\alpha_3, \alpha_5, \alpha_6\}\}$ while $M^* = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_4, \alpha_5, \alpha_8\}, \{\alpha_6, \alpha_7, \alpha_9\}\}$. Since $u(M) = 6$ and $u(M^*) = 12$ it follows that $u(M^*) = 2u(M)$ and thus that our analysis of Algorithm `findStableUW` is tight. Interestingly, this particular instance also shows that any approximation algorithm with a better performance ratio than 2 must not always begin, like Algorithm `findStableUW` does, by selecting a maximal set of triangles.

6.4 Binary preferences

In this section we show that deciding if a given instance of 3DR-AS contains a stable matching is NP-complete, even when preferences are binary (and not necessarily symmetric). The reduction is from *Partition Into Triangles* (PIT, Problem 5.1).

The reduction, illustrated in Figure 6.11, is as follows. Unless otherwise specified assume that $v_{\alpha_i}(\alpha_j) = 0$ for any $\alpha_i, \alpha_j \in N$. For each i where $1 \leq i \leq 3q$ construct three agents labelled a_{2i} , a_{2i-1} , and b_i . Let $v_{a_{2i}}(a_{2i-1}) = v_{a_{2i}}(b_i) = 1$, $v_{a_{2i-1}}(a_{2i}) = v_{a_{2i-1}}(b_i) = 1$, and $v_{b_i}(a_{2i}) = v_{b_i}(a_{2i-1}) = 1$. For each $w_i, w_j \in W$ let $v_{b_i}(b_j) = 1$ if $\{w_i, w_j\} \in E$ and 0 otherwise. Next, for each r where $1 \leq r \leq 6q$ construct a set of five agents $P_r = \{p_r^1, p_r^2, \dots, p_r^5\}$, which we refer to as the r^{th} *pentagadget*. To simplify the description of the valuations in each pentagadget, in this section we write $i \oplus y$ to denote $((i + y - 1) \bmod 5) + 1$. For

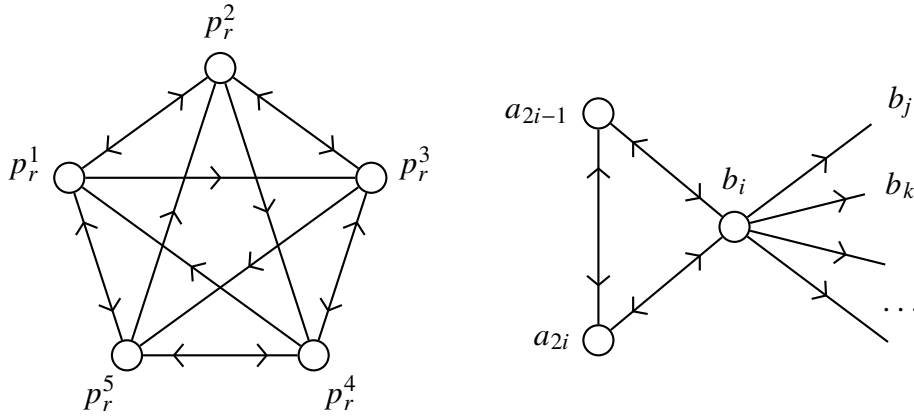


Figure 6.11: The reduction from PIT to the problem of deciding if an instance of 3DR-AS with binary preferences contains a stable matching. Each vertex represents an agent. An arc is present from agent α_i to agent α_j if $v_{\alpha_i}(\alpha_j) = 1$. Depicted is some pentagadget P_r and some agents b_i , a_{2i} , and a_{2i-1} where $1 \leq i \leq 3q$ and $N(w_i) = \{w_j, w_k, \dots\}$.

each i where $1 \leq i \leq 5$ let $v_{p_r^i}(p_r^{i \oplus 1}) = v_{p_r^{i \oplus 1}}(p_r^i) = 1$ and $v_{p_r^i}(p_r^{i \oplus 2}) = 1$. This completes the construction of (N, V) . Note that $|N| = 39q$.

It is straightforward to show that this reduction can be performed in polynomial time. To prove that the reduction is correct we show that the 3DR-AS instance (N, V) contains a stable matching if and only if the PIT instance G contains a partition into triangles.

We first show that if the PIT instance G contains a partition into triangles then the 3DR-AS instance (N, V) contains a stable matching.

Lemma 6.21. *If G contains a partition into triangles then (N, V) contains a stable matching.*

Proof. Suppose G contains a partition into triangles $X = \{X_1, X_2, \dots, X_q\}$. We shall construct a matching M that is stable in (N, V) . For each triangle $X_p = \{w_i, w_j, w_k\} \in W$, add $\{b_i, b_j, b_k\}$ to M . For each r where $1 \leq r \leq 6q$, add $\{p_r^1, p_r^2, p_r^3\}$ to M . This leaves agents a_{2i} and a_{2i-1} for each $1 \leq i \leq 3q$ and agents p_r^4 and p_r^5 for each $0 \leq r \leq 6q$. For each $1 \leq i \leq 3q$, add to M the triples $\{a_{2i}, p_{2i}^4, p_{2i}^5\}, \{a_{2i-1}, p_{2i-1}^4, p_{2i-1}^5\}$.

Since $u_{b_i}(M) = 2$ for each $1 \leq i \leq 3q$ it follows that b_i does not belong to a triple that blocks M .

Suppose for a contradiction that some agent a_{2i} where $1 \leq i \leq 3q$ belongs to a triple t that blocks M . We have shown that b_i does not belong to a triple that blocks M , so it must be that $a_{2i-1} \in t$, otherwise $u_{a_{2i}}(t) = 0$, which is impossible. Suppose then that $t = \{a_{2i}, a_{2i-1}, \alpha_j\}$ where $\alpha_j \in N$ and $\alpha_j \neq b_i$. Considering the design of the instance, for any such α_j it must be that $u_{\alpha_j}(\{a_{2i}, a_{2i-1}\}) = u_{\alpha_j}(t) = 0$, which is a contradiction. A symmetric argument shows that no a_{2i-1} where $1 \leq i \leq 3q$ belongs to a triple that blocks M .

The remaining possibility is that some triple $\{p_r^{s_1}, p_r^{s_2}, p_r^{s_3}\}$ blocks M where $1 \leq r \leq 6q$. By the construction of M , $u_{p_r^1}(M) = u_{p_r^2}(M) = 2$, so neither p_r^1 nor p_r^2 blocks M . It follows that $\{s_1, s_2, s_3\} = \{3, 4, 5\}$, which is a contradiction since $u_{p_r^5}(M) = 1 = u_{p_r^5}(\{p_r^3, p_r^4\})$. \square

We now show, using a sequence of lemmas, that if the 3DR-AS instance (N, V) contains a stable matching then G contains a partition into triangles.

We also introduce some new notation. For any set $S \subseteq N$ let $\sigma(S, N)$ be the number of triples in N that each contain at least one agent in S .

Lemma 6.22. *If (N, V) contains a stable matching M then $\sigma(P_r, M) = 2$ for any r where $1 \leq r \leq 6q$.*

Proof. By definition, $2 \leq \sigma(P_r, M) \leq 5$. Suppose for a contradiction that $\sigma(P_r) \geq 4$. It must be that at least three triples in M contain exactly one agent in P_r . Label these three triples as $M(p_r^{s_1})$, $M(p_r^{s_2})$, and $M(p_r^{s_3})$. It follows that $u_{p_r^{s_1}}(M) = u_{p_r^{s_2}}(M) = u_{p_r^{s_3}}(M) = 0$. By the design of the reduction it must be that $u_{p_r^{s_1}}(\{p_r^{s_2}, p_r^{s_3}\}) \geq 1$, $u_{p_r^{s_2}}(\{p_r^{s_1}, p_r^{s_3}\}) \geq 1$, and $u_{p_r^{s_3}}(\{p_r^{s_1}, p_r^{s_2}\}) \geq 1$. Now $\{p_r^{s_1}, p_r^{s_2}, p_r^{s_3}\}$ blocks M , which is a contradiction.

Suppose then, for a contradiction, that $\sigma(P_r, M) = 3$. There are two possibilities. In the first, two triples in M each contain exactly two agents in P_r and one triple in M contains exactly one agent in P_r . In the second, two triples in M each contain exactly one agent in P_r and one triple in M contains exactly three agents in P_r .

Suppose firstly that two triples in M each contain exactly two agents in P_r and one triple in M contains exactly one agent in P_r . Assume without loss of generality that p_r^1 is the agent in the latter triple. It follows that $u_{p_r^1}(M) = 0$. By assumption, $M(p_r^4)$ and $M(p_r^5)$ each contain exactly two agents in P_r so it follows that $u_{p_r^4}(M) \leq 1$ and $u_{p_r^5}(M) \leq 1$. Now $\{p_r^1, p_r^4, p_r^5\}$ blocks M since $u_{p_r^4}(\{p_r^5, p_r^1\}) = u_{p_r^5}(\{p_r^1, p_r^4\}) = 2$ and $u_{p_r^1}(\{p_r^4, p_r^5\}) = 1$, which is a contradiction.

It remains that two triples in M each contain exactly one agent in P_r and one triple in M contains exactly three agents in P_r . Suppose $p_r^{s_1}$ and $p_r^{s_2}$ are the two agents in the former two triples. Excluding symmetries, there are two possible cases. In the first case, $s_1 = 1$ and $s_2 = 2$. It follows that $\{p_r^3, p_r^4, p_r^5\} \in M$. Now $\{p_r^5, p_r^1, p_r^2\}$ blocks M since $u_{p_r^5}(\{p_r^1, p_r^2\}) = u_{p_r^1}(\{p_r^2, p_r^5\}) = 2$ and $u_{p_r^2}(\{p_r^1, p_r^5\}) = 1$, which is a contradiction. In the second case, $s_1 = 1$ and $s_2 = 3$. It follows that $\{p_r^2, p_r^4, p_r^5\} \in M$. Now $\{p_r^1, p_r^2, p_r^3\}$ blocks M since $u_{p_r^1}(\{p_r^2, p_r^3\}) = u_{p_r^2}(\{p_r^1, p_r^3\}) = 2$ and $u_{p_r^3}(\{p_r^1, p_r^2\}) = 1$, which is also a contradiction. \square

Lemma 6.23. *If (N, V) contains a stable matching M then $u_{a_k}(M) = 0$ for each $1 \leq k \leq 6q$.*

Proof. Suppose M is a stable matching in (N, V) . Consider an arbitrary pentagadget index r_1 where $1 \leq r_1 \leq 6q$. By Lemma 6.22, it must be that $\sigma(P_{r_1}, M) = 2$. It follows that one

triple in M contains exactly three agents in P_{r_1} and another triple in M contains exactly two agents in P_r as well as some third agent α_h . It follows that $u_{\alpha_h}(M) = 0$. We now show that $\alpha_h = a_k$ where $1 \leq k \leq 6q$.

By the design of the reduction, it must be that either $\alpha_h \in P_{r_2}$ where $1 \leq r_2 \leq 6q$, $\alpha_h = b_j$ where $1 \leq j \leq 3q$, or $\alpha_h = a_k$ where $1 \leq k \leq 6q$.

Suppose firstly that $\alpha_h \in P_r$ where $1 \leq r_2 \leq 6q$. Label $\alpha_h = p_{r_2}^s$ where $1 \leq s \leq 5$. By the definition of $\alpha_h = p_{r_2}^s$, it must be that $r_1 \neq r_2$. Since $M(p_{r_2}^s)$ contains $p_{r_2}^s$ and two agents in P_{r_1} , by Lemma 6.22 the four agents in $P_{r_2} \setminus \{p_{r_2}^s\}$ must belong to exactly one triple in M , which is clearly a contradiction.

Suppose then that $\alpha_h = b_j$ where $1 \leq j \leq 3q$. Consider a_{2j} and a_{2j-1} . Since $a_{2j} \notin M(b_j)$ and $a_{2j-1} \notin M(b_j)$ it must be that $u_{a_{2j}}(M) \leq 1$ and $u_{a_{2j-1}}(M) \leq 1$. Since $u_{\alpha_h}(M) = u_{b_j}(M) = 0$ it follows that $\{b_j, a_{2j}, a_{2j-1}\}$ blocks M , since $u_{b_j}(\{a_{2j}, a_{2j-1}\}) = u_{a_{2j}}(\{b_j, a_{2j-1}\}) = u_{a_{2j-1}}(\{b_j, a_{2j}\}) = 2$, which is a contradiction.

It remains that $\alpha_h = a_k$ where $1 \leq k \leq 6q$. Since the choice of r_1 where $1 \leq r_1 \leq 6q$ was arbitrary, there are exactly $6q$ choices of α_h . It follows that $u_{a_k}(M) = 0$ for every $1 \leq k \leq 6q$. \square

Lemma 6.24. *If (N, V) contains a stable matching M then $u_{b_i}(M) = 2$ for any i where $1 \leq i \leq 3q$.*

Proof. Suppose for a contradiction that there exists some $1 \leq i \leq 3q$ where $u_{b_i}(M) < 2$. Lemma 6.23 shows that $u_{2i}(M) = u_{a_{2i-1}}(M) = 0$. Considering the valuation functions of a_{2i} , a_{2i-1} , and b_i , we can see that $u_{b_i}(\{a_{2i}, a_{2i-1}\}) = u_{a_{2i}}(\{b_i, a_{2i-1}\}) = u_{a_{2i-1}}(\{b_i, a_{2i}\}) = 2$. Now $\{b_i, a_{2i}, a_{2i-1}\}$ blocks M , which is a contradiction. \square

Lemma 6.25. *If (N, V) contains a stable matching M then for any b_i where $1 \leq i \leq 3q$, the triple $M(b_i)$ comprises $\{b_i, b_j, b_k\}$ where $1 \leq j, k \leq 3q$ and $\{w_i, w_j\}, \{w_i, w_k\} \in E$.*

Proof. Lemma 6.24 shows that $u_{b_i}(M) = 2$. Suppose $M(b_i) = \{b_i, \alpha_k, \alpha_l\}$ for some $\alpha_k, \alpha_l \in N$. Since $u_{b_i}(M) = 2$, it must be that $v_{b_i}(\alpha_k) = 1$ and hence either $\alpha_k = a_{2i}$, $\alpha_k = a_{2i-1}$, or $\alpha_k = b_j$ where $1 \leq j \leq 3q$ where $\{w_i, w_j\} \in E$. Suppose first that $\alpha_k = a_{2i}$. Since $b_i \in M(a_{2i})$ it follows that $u_{a_{2i}}(M) \geq 1$ which contradicts Lemma 6.23. A similar argument shows that $\alpha_k \neq a_{2i-1}$. It remains that $\alpha_k = b_j$ where $1 \leq j \leq 3q$ such that $\{w_i, w_j\} \in E$. The same argument shows that $\alpha_l = b_k$ where $1 \leq k \leq 3q$ where $\{w_i, w_k\} \in E$. We have shown that $M(b_i) = \{b_i, b_j, b_k\}$ for some j, k where $1 \leq j, k \leq 3q$ and $\{w_i, w_j\}, \{w_i, w_k\} \in E$. \square

Lemma 6.26. *If (N, V) contains a stable matching then G contains a partition into triangles.*

Proof. Lemma 6.25 shows that for an arbitrary b_i where $1 \leq i \leq 3q$, $M(b_i)$ comprises $\{b_i, b_j, b_k\}$ where $1 \leq j, k \leq 3q$, $\{w_i, w_j\} \in E$, and $\{w_i, w_k\} \in E$. It follows that there are exactly q

triples in M each containing three agents $\{b_i, b_j, b_k\}$, where the three corresponding vertices w_i, w_j, w_k are pairwise adjacent in G . From these triples of pairwise adjacent vertices, a partition into triangles X can be easily constructed. \square

We have now shown that the 3DR-AS instance (N, V) contains a stable matching if and only if the PIT instance G contains a partition into triangles. This shows that the reduction is correct.

Theorem 6.4. *Deciding if a given instance of 3DR-AS contains a stable matching is NP-complete, even when preferences are binary.*

Proof. It is straightforward to show that this decision problem belongs to NP. We have presented a polynomial-time reduction from Partition Into Triangles (PIT, Problem 5.1), which is NP-complete [43]. Given an arbitrary instance G of PIT, the reduction constructs an instance (N, V) of 3DR-AS with binary preferences. Lemmas 6.21 and 6.26 show that (N, V) contains a stable matching if and only if G contains a partition into triangles and thus that this decision problem is NP-hard. \square

6.5 Symmetric ternary preferences

We saw in Section 6.4 that an instance (N, V) of 3DR-AS may not contain a stable matching, and the associated decision problem is NP-complete, even when valuations are binary. It follows that the decision problem for ternary valuations, i.e. $v_{\alpha_i}(\alpha_j) \in \{0, 1, 2\}$ for each $\alpha_i, \alpha_j \in N$ is also NP-complete. In contrast we saw in Section 6.3 that when valuations are binary and symmetric a stable matching always exists and can be found in polynomial time. It is natural to ask if this polynomial-time solvability also holds in the more general case of ternary and symmetric valuations. We answer this question in the negative (assuming $P = NP$), and show that deciding if a given instance of 3DR-AS contains a stable matching is NP-complete, even when valuations are ternary and symmetric. We remark that a result of Deineko and Woeginger [63] for *Geometric 3D-SR* (see Chapter 2) implies a weaker version of our result, namely that if valuations are symmetric (but not necessarily ternary) then deciding if a given instance of 3DR-AS contains a stable matching is NP-complete, even when valuations are ternary and symmetric.

We present a polynomial-time reduction from Partition into Triangles (PIT, Problem 5.1), which is similar to the reduction that we presented in Section 6.4 for the analogous decision problem involving preferences that are binary but not necessarily symmetric. The main difference is in the design of the gadgets. Instead of “pentagadgets” we introduce a number of “octogadgets”. Nevertheless, the purpose of the octogadgets is the same as the pentagadgets in the previous reduction and the proof follows the same structure as before.

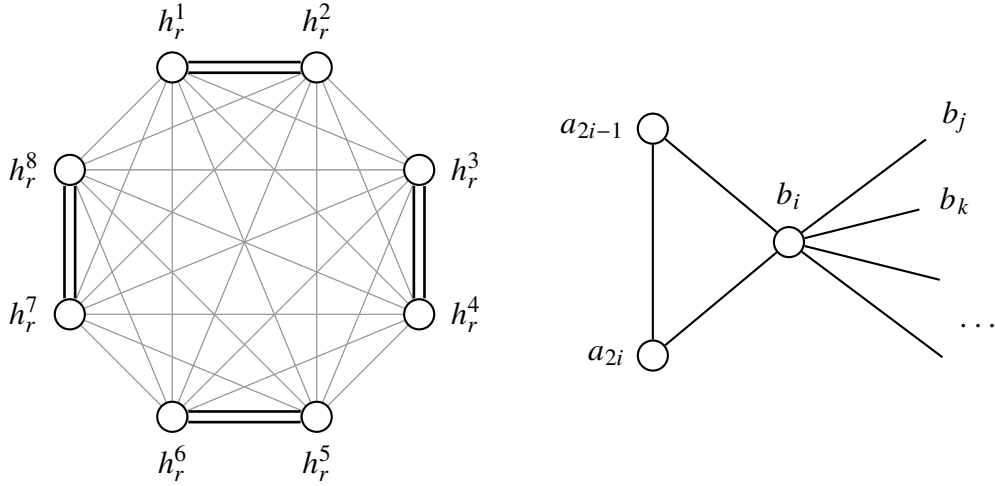


Figure 6.12: The reduction from PIT to the problem of deciding if an instance of 3DR-AS with ternary preferences contains a stable matching. Each vertex represents an agent. A single edge is present from agent α_i to agent α_j if $v_{\alpha_i}(\alpha_j) = 1$. A double edge is present from α_i to α_j if $v_{\alpha_i}(\alpha_j) = 2$. Depicted is some octogadget H_r and some agents b_i , a_{2i} , and a_{2i-1} where $1 \leq i \leq 6q$ and $N(w_i) = \{w_j, w_k, \dots\}$.

The reduction, illustrated in Figure 6.12, is as follows. Since valuations are symmetric in (N, V) , we shall usually specify valuations in one direction only. For example, instead of writing “let $v_{\alpha_i}(\alpha_j) = v_{\alpha_j}(\alpha_i) = 1$ ” we write “let $v_{\alpha_i}(\alpha_j) = 1$ ”. Unless otherwise specified assume that $v_{\alpha_i}(\alpha_j) = 0$ for any $\alpha_i, \alpha_j \in N$. To simplify the description of the valuations in the reduction, in this section we write $i \oplus y$ to denote $((i + y - 1) \bmod 8) + 1$.

For each i where $1 \leq i \leq 3q$ construct three agents labelled a_{2i} , a_{2i-1} , and b_i . Let $v_{a_{2i}}(a_{2i-1}) = v_{b_i}(a_{2i}) = v_{b_i}(a_{2i-1}) = 1$. For each $w_i, w_j \in W$ let $v_{b_i}(b_j) = 1$ if $\{w_i, w_j\} \in E$ and 0 otherwise. Next, for each r where $1 \leq r \leq 6q$ construct a set of eight agents $H_r = \{h_r^1, h_r^2, \dots, h_r^8\}$, which we refer to as the h^{th} octogadget. For each i and j where $1 \leq i, j \leq 8$ let $v_{h_r^i}(h_r^j) = 2$ if both i is odd and $j = i \oplus 1$ otherwise 1. This completes the construction of (N, V) . Note that $|N| = 57q$.

It is straightforward to show that the reduction runs in polynomial time. To prove that the reduction is correct we show that the 3DR-AS instance (N, V) contains a stable matching if and only if the PIT instance G contains a partition into triangles.

We first show that if the PIT instance G contains a partition into triangles then the 3DR-AS instance (N, V) contains a stable matching.

Lemma 6.27. *If G contains a partition into triangles then (N, V) contains a stable matching.*

Proof. Suppose G contains a partition into triangles $X = \{X_1, X_2, \dots, X_q\}$. We shall construct a matching M that is stable in (N, V) . For each triangle $X_p = \{w_i, w_j, w_k\} \in W$, add $\{b_i, b_j, b_k\}$ to M . For each index r where $1 \leq r \leq 6q$ add the triples $\{h_r^1, h_r^2, h_r^3\}$,

$\{h_r^4, h_r^5, p_r^6\}, \{h_r^7, h_r^8, a_r\}$ to M . Note that $u_{h_r^p}(M) \geq 2$ for each $1 \leq p \leq 8$ by the design of the octogadget H_r .

Since $u_{b_i}(M) = 2$ for each $1 \leq i \leq 3q$ it follows that b_i does not belong to a triple that blocks M .

Suppose for a contradiction that some agent a_{2i} where $1 \leq i \leq 3q$ belongs to a triple t that blocks M . We have shown that b_i does not belong to a triple that blocks M , so it must be that $a_{2i-1} \in t$, otherwise $u_{a_{2i}}(t) = 0$, which is impossible. Suppose then that $t = \{a_{2i}, a_{2i-1}, \alpha_j\}$ where $\alpha_j \in N$ and $\alpha_j \neq b_i$. Considering the design of the instance, for any such α_j it must be that $u_{\alpha_j}(\{a_{2i}, a_{2i-1}\}) = u_{\alpha_j}(t) = 0$, which is a contradiction. A symmetric argument shows that no a_{2i-1} where $1 \leq i \leq 3q$ belongs to a triple that blocks M .

The only remaining possibility is that some triple $\{h_r^{s_1}, h_r^{s_2}, h_r^{s_3}\}$ blocks M where $1 \leq r \leq 6q$ and $1 \leq s_1, s_2, s_3 \leq 8$. Suppose for a contradiction that some such triple exists. We noted earlier in this proof that $u_{h_r^p}(M) \geq 2$ for each $1 \leq p \leq 11$ so it must be that $u_{h_r^{s_1}}(\{h_r^{s_2}, h_r^{s_3}\}) \geq 3$, $u_{h_r^{s_2}}(\{h_r^{s_1}, h_r^{s_3}\}) \geq 3$, and $u_{h_r^{s_3}}(\{h_r^{s_1}, h_r^{s_2}\}) \geq 3$. Considering the valuations of the agents in H_r we can see that no such $h_r^{s_1}, h_r^{s_2}, h_r^{s_3}$ exist, which is a contradiction. \square

We now show, using a sequence of lemmas, that if the 3DR-AS instance (N, V) contains a stable matching then G contains a partition into triangles.

Lemma 6.28. *If (N, V) contains a stable matching M then $u_{h_r^s}(M) \geq 1$ for any r and s where $1 \leq r \leq 6q$ and $1 \leq s \leq 11$.*

Proof. Suppose for a contradiction that $u_{h_r^{s_1}}(M) = 0$ for some $1 \leq r \leq 6q$ and $1 \leq s_1 \leq 8$. Note that it must be that $M(h_r^{s_1})$ contains exactly one agent, namely $h_r^{s_1}$, in H_r .

We claim that no triple in M contains exactly two agents in H_r . Suppose for a contradiction that some such triple $\{h_r^{s_2}, h_r^{s_3}, \alpha_i\}$ exists where $1 \leq s_2, s_3 \leq 8$ and $\alpha_i \notin H_r$. By the construction of H_r , it must be that $u_{h_r^{s_1}}(\{h_r^{s_2}, h_r^{s_3}\}) \geq 2$. Since $\alpha_i \notin H_r$ it must also be that $v_{h_r^{s_2}}(h_r^{s_1}) > v_{h_r^{s_2}}(\alpha_i)$ and $v_{h_r^{s_3}}(h_r^{s_1}) > v_{h_r^{s_3}}(\alpha_i)$ so $u_{h_r^{s_2}}(\{h_r^{s_1}, h_r^{s_3}\}) > u_{h_r^{s_2}}(M)$ and $u_{h_r^{s_3}}(\{h_r^{s_1}, h_r^{s_2}\}) > u_{h_r^{s_3}}(M)$. It follows that the triple $\{h_r^{s_1}, h_r^{s_2}, h_r^{s_3}\}$ blocks M , which is a contradiction.

We now claim that at most two triples in M contain exactly one agent in H_r . Suppose three or more triples in M each contain exactly one agent in H_r . It follows that these three agents in H_r each have utility zero in M . By the construction of H_r , these three agents block M .

We have now established that no triple in M contains exactly two agents in H_r and at most two triples in M contain exactly one agent in H_r . Since $|H_r| = 8$ the only possibility is that two triples in M each contain exactly three agents in H_r and two triples in M each contain exactly one agent in H_r .

By the design of the octogadget H_r there are four disjoint pairs of agents $\{h_r^{p_1}, h_r^{p_2}\}$ for which $v_{h_r^{p_1}}(h_r^{p_2}) = 2$. Since there are exactly two triples that contain three agents in H_r there exists at

least two disjoint pairs of agents $\{h_r^{p_1}, h_r^{p_2}\}$ and $\{h_r^{p_3}, h_r^{p_4}\}$ where $v_{h_r^{p_1}}(h_r^{p_2}) = v_{h_r^{p_3}}(h_r^{p_4}) = 2$ and $M(h_r^{p_1}) \neq M(h_r^{p_2})$ and $M(h_r^{p_3}) \neq M(h_r^{p_4})$. For example, if $\{h_r^3, h_r^4, h_r^5\} \in M$ and $\{h_r^6, h_r^7, h_r^8\} \in M$ then $\{\{h_r^{p_1}, h_r^{p_2}\}, \{h_r^{p_3}, h_r^{p_4}\}\} = \{\{h_r^1, h_r^2\}, \{h_r^5, h_r^6\}\}$. Suppose without loss of generality that $\{h_r^{p_1}, h_r^{p_2}\}$ does not contain $h_r^{s_1}$. Since $h_r^{p_2} \notin M(h_r^{p_1})$, by the design of H_r it must be that $u_{h_r^{p_1}} \leq 2$. Similarly, since $h_r^{p_1} \notin M(h_r^{p_2})$ it must be that $u_{h_r^{p_2}} \leq 2$. It follows that the triple $\{h_r^{s_1}, h_r^{p_1}, h_r^{p_2}\}$ blocks M , since $u_{h_r^{s_1}}(M) = 0 < u_{h_r^{s_1}}(\{h_r^{p_1}, h_r^{p_2}\}) = 2$, $u_{h_r^{p_1}}(M) \leq 2 < u_{h_r^{p_1}}(\{h_r^{s_1}, h_r^{p_2}\}) = 3$, and $u_{h_r^{p_2}}(M) \leq 2 < u_{h_r^{p_2}}(\{h_r^{s_1}, h_r^{p_1}\}) = 3$. This contradicts the supposition that M is stable. \square

Lemma 6.29. *If (N, V) contains a stable matching M then $u_{a_k}(M) = 0$ for each $1 \leq k \leq 6q$.*

Proof. Consider some arbitrary octogadget H_{r_1} where $1 \leq r_1 \leq 6q$. Since $|H_{r_1}| = 8$ there exists at least one triple in M that contains some agent $h_{r_1}^{s_1}$ where $1 \leq s_1 \leq 8$ and some agent $\alpha_i \notin H_{r_1}$. By Lemma 6.28, $u_{h_{r_1}^{s_1}}(M) \geq 1$, and it follows that $M(h_{r_1}^{s_1}) = \{h_{r_1}^{s_1}, h_{r_1}^{s_2}, \alpha_i\}$ where $1 \leq s_2 \leq 8$ and $\alpha_i \in N$. It must be that $\alpha_i \notin H_{r_2}$ for any r_2 where $1 \leq r_2 \leq 6q$, for otherwise $u_{\alpha_i}(M) = 0$, which contradicts Lemma 6.28. It remains that either $\alpha_i = b_j$ where $1 \leq j \leq 3q$ or $\alpha_i = a_k$ where $1 \leq k \leq 6q$. Note that by the design of the instance it follows that $u_{\alpha_i}(M) = 0$.

Suppose for a contradiction that $\alpha_i = b_j$ where $1 \leq j \leq 3q$. It must also be that $u_{a_{2j}}(M) \leq 1$ and $u_{a_{2j-1}}(M) \leq 1$, since $a_{2j} \notin M(b_j)$ and $a_{2j-1} \notin M(b_j)$. We can now see that $\{b_j, a_{2j}, a_{2j-1}\}$ blocks M since $u_{b_j}(M) = 0 < u_{b_j}(\{a_{2j}, a_{2j-1}\}) = 2$, $u_{a_{2j}}(M) \leq 1 < 2 = u_{a_{2j}}(\{b_j, a_{2j-1}\})$ and $u_{a_{2j-1}}(M) \leq 1 < 2 = u_{a_{2j-1}}(\{b_j, a_{2j}\})$. This contradicts the supposition that M is stable. It remains that $\alpha_i = a_k$ where $1 \leq k \leq 6q$. Recall that $u_{\alpha_i}(M) = 0$. Since there are $6q$ octogadgets and the choice of r_1 was arbitrary the only possibility is that $u_{a_k}(M) = 0$ for every $1 \leq k \leq 6q$. \square

Lemma 6.30. *If (N, V) contains a stable matching M then $u_{b_i}(M) = 2$ for any i where $1 \leq i \leq 3q$.*

Proof. Suppose for a contradiction that there exists some $1 \leq i \leq 3q$ where $u_{b_i}(M) < 2$. Lemma 6.29 shows that $u_{2i}(M) = u_{a_{2i-1}}(M) = 0$. Considering the valuation functions of a_{2i} , a_{2i-1} , and b_i , we can see that $u_{b_i}(\{a_{2i}, a_{2i-1}\}) = u_{a_{2i}}(\{b_i, a_{2i-1}\}) = u_{a_{2i-1}}(\{b_i, a_{2i}\}) = 2$. The triple $\{b_i, a_{2i}, a_{2i-1}\}$ therefore blocks M , which is a contradiction. \square

Lemma 6.31. *If (N, V) contains a stable matching M then for any b_i where $1 \leq i \leq 3q$, the triple $M(b_i)$ comprises $\{b_i, b_j, b_k\}$ for some $1 \leq j, k \leq 3q$ where $\{w_i, w_j\}, \{w_i, w_k\} \in E$.*

Proof. Lemma 6.30 shows that $u_{b_i}(M) = 2$. Suppose $M(b_i) = \{b_i, \alpha_k, \alpha_l\}$ for some $\alpha_k, \alpha_l \in N$. Since $u_{b_i}(M) = 2$, it must be that $v_{b_i}(\alpha_k) = 1$ and hence either $\alpha_k = a_{2i}$, $\alpha_k = a_{2i-1}$, or $\alpha_k = b_j$ where $1 \leq j \leq 3q$ where $\{w_i, w_j\} \in E$. Suppose first that $\alpha_k = a_{2i}$. Since

$b_i \in M(a_{2i})$ it follows that $u_{a_{2i}}(M) \geq 1$ which contradicts Lemma 6.29. A similar argument shows that $\alpha_k \neq a_{2i-1}$. It remains that $\alpha_k = b_j$ where $1 \leq j \leq 3q$ such that $\{w_i, w_j\} \in E$. The same argument shows that $\alpha_l = b_k$ where $1 \leq k \leq 3q$ and $\{w_i, w_k\} \in E$. We have shown that $M(b_i) = \{b_i, b_j, b_k\}$ where $1 \leq j, k \leq 3q$ and $\{w_i, w_j\}, \{w_i, w_k\} \in E$. \square

Lemma 6.32. *If (N, V) contains a stable matching then G contains a partition into triangles.*

Proof. Lemma 6.31 shows that for an arbitrary b_i where $1 \leq i \leq 3q$, $M(b_i)$ comprises $\{b_i, b_j, b_k\}$ where $1 \leq j, k \leq 3q$, $\{w_i, w_j\} \in E$, and $\{w_i, w_k\} \in E$. It follows that there are exactly q triples in M each containing three agents $\{b_i, b_j, b_k\}$, where the three corresponding vertices w_i, w_j, w_k are pairwise adjacent in G . From these triples of pairwise adjacent vertices, a partition into triangles X can be easily constructed. \square

We have now shown that the 3DR-AS instance (N, V) contains a stable matching if and only if the PIT instance G contains a partition into triangles. This shows that the reduction is correct.

Theorem 6.5. *Deciding if a given instance of 3DR-AS contains a stable matching is NP-complete, even when preferences are ternary and symmetric.*

Proof. It is straightforward to show that this decision problem belongs to NP. We have presented a polynomial-time reduction from Partition Into Triangles (PIT), which is NP-complete [43]. Given an arbitrary instance G of PIT, the reduction constructs an instance (N, V) of 3DR-AS with ternary and symmetric preferences. Lemmas 6.27 and 6.32 show that (N, V) contains a stable matching if and only if G contains a partition into triangles and thus that this decision problem is NP-hard. \square

6.6 Summary and open problems

In this chapter we formalised a model of 3DR involving additively separable preferences, which we called 3DR-AS. We considered in instances of 3DR-AS the existence of, and complexity of finding, matchings that are stable, under three possible restrictions of the agents' valuations.

We first showed that any instance of 3DR-AS with binary and symmetric preferences contains a stable matching, and presented a polynomial-time algorithm that can construct a stable matching in such an instance. We then considered the problem of finding a stable matching with maximum utilitarian welfare, given an instance with binary and symmetric preferences. We called this problem the *3DR-AS Stable Maximum Utilitarian Welfare problem* (3DR-AS-SMUW). We proved that 3DR-AS-SMUW is NP-hard, but showed that the algorithm for

constructing a stable matching in this setting can be modified to yield a 2-approximation algorithm. We then complemented the previous tractability results with two hardness results. The first is that a stable matching need not exist in general, and the associated decision problem is NP-complete even when preferences are binary and not necessarily symmetric. The second is that the same decision problem is NP-complete even when preferences are ternary and symmetric.

Later, in Chapter 7, we shall combine some of the results from this chapter with other results for 3DR-AS and present a comprehensive analysis of the existence and complexity of feasible matchings in 3DR-AS under restricted preferences, for four related solution concepts. This classification is shown in Table 7.1 in Chapter 7.

We now present some open problems specifically involving stability in 3DR-AS. More general problems, involving solution concepts other than stability and other models of fixed-size coalitions, are discussed in Chapter 9.

An immediate open problem is whether our results for binary preferences hold in a more general setting in which $v_{\alpha_i}(\alpha_j) \in \{a, b\}$ for any non-negative integers a and b where $a < b$, and whether our results for ternary preferences hold when $v_{\alpha_i}(\alpha_j) \in \{a, b, c\}$ for any non-negative integers a, b , and c where $a < b < c$. We conjecture that both statements are true, and that both the polynomial-time algorithm and NP-hardness reductions can be modified accordingly.

As we noted in Section 6.3.2, any approximation algorithm for 3DR-AS-SMUW with approximation ratio strictly less than 2 must not always begin, like Algorithm `findStableUW` does, by selecting a maximal set of triangles. While it would be very interesting to derive such an algorithm, it would also be informative to derive an inapproximability result for 3DR-AS-SMUW. We conjecture that 3DR-AS-SMUW is APX-hard.

An open direction of work involves the *price of anarchy* and *price of stability* of 3DR-AS [73]. In the setting of 3DR-AS, we could define the price of anarchy (stability) as the worst- (best-)case ratio between the utilitarian welfare of an arbitrary stable matching and the maximum possible utilitarian welfare over all matchings. Our example instance, shown in Figure 6.10, thus shows that the price of anarchy is at least 2, even when preferences are binary and symmetric.

It would be very interesting to identify other restrictions of 3DR-AS in which a stable matching can be found in polynomial time. For example, a crucial part of the reduction shown in Section 6.4 was the pentagadget, which corresponds to a unique type of regular tournament graph [84]. It would be quite remarkable if, for example, every instance of 3DR-AS that does not contain a pentagadget contains a stable matching. As a starting point, we conjecture that any instance corresponding to a directed acyclic graph must contain a stable matching.

Continuing the connection to graph theory (discussed in Section 6.1), it might also be possible to consider the problem of finding a stable matching in a given instance of 3DR-AS from the perspective of the parameterised complexity with respect to a graph parameter. For example, in the case of binary and symmetric preferences, one could consider the tree-width [85] of the underlying graph.

As we noted for 3DR-B and 3DR-W (in Chapters 4 and 5), it would be very interesting to estimate the probability that a random instance of 3DR-AS contains a stable matching, or to estimate the same probability in a random instance of 3DR-AS with binary or ternary preferences. It might be possible to use probabilistic techniques from graph theory, such as the Erdős-Rényi model of a random graph. Alternatively an empirical approach might be informative, for example by formulating the problem as an integer program [86].

Chapter 7

Three-Dimensional Envy-Free Roommates with Additively Separable Preferences

7.1 Introduction

In the previous chapter we considered the existence of stable matchings in a model of Three-Dimensional Roommates with additively separable preferences, called 3DR-AS. We showed that in general, a stable matching may not exist in a given instance of 3DR-AS, and the associated decision problem is NP-complete. We also showed that if the agents' preferences are sufficiently restricted then a stable matching must always exist, which can be found in polynomial time. In this chapter we consider three alternative solution concepts, one of which is a strictly weaker concept than stability. For each concept, we study the existence of such matchings under different restrictions on the agents' preferences, and the computational complexity of the associated existence and construction problems (see Chapter 2). This culminates in a full complexity classification, which includes for each of the three solution concepts a dichotomy between polynomial-time solvability and NP-hardness.

The three solution concepts that we study all relate to matchings in which there exist no two agents α_i and α_j where α_i would prefer to swap with α_j . In such a case we say that α_i has *envy* for α_j . If a matching M contains no agent with envy for another agent then we say that it is *envy-free*. The other two solution concepts involve the other agents in the triple of α_j , $M(\alpha_j)$. Informally, we say that α_i has *justified envy* (abbreviated j-envy) for α_j if α_i has envy for α_j and each of the other agents in $M(\alpha_j)$ prefer α_i to α_j . We say that α_i has *weakly justified envy* (wj-envy) for α_j if α_i has envy for α_j and each of the other agents in $M(\alpha_j)$ either prefers α_i to α_j or is indifferent. Formally, an agent α_i has envy for another agent α_j in

a matching M if $u_{\alpha_i}(M(\alpha_j) \setminus \{\alpha_j\}) > u_{\alpha_i}(M)$. If α_i has envy for α_j and $v_{\alpha_k}(\alpha_i) > v_{\alpha_k}(\alpha_j)$ for each $\alpha_k \in M(\alpha_j) \setminus \{\alpha_j\}$ then we say that α_i has j-envy for α_j . If $v_{\alpha_k}(\alpha_i) \geq v_{\alpha_k}(\alpha_j)$ for each $\alpha_k \in M(\alpha_j) \setminus \{\alpha_j\}$ then we say that α_i has wj-envy for α_j . The definitions of j-envy-free and wj-envy-free matchings are analogous.

These three solution concepts are hierarchical: by definition, any matching that is envy-free is also wj-envy-free, and any matching that is wj-envy-free must also be j-envy-free. In fact, they are also related to other solution concepts proposed in the literature. It is relatively straightforward to show that any envy-free matching is also *strictly swap stable* [73]. It is also straightforward to show that any stable matching is also j-envy-free. In fact, j-envy-freeness is a strictly weaker concept than stability. With the results of Chapter 6 in mind, this observation partly motivates our study of j-envy-freeness, since for example it would be interesting if a strictly weaker restriction on the agents' preferences than binary and symmetric valuations (which guarantees the existence of a stable matching) is sufficient to guarantee the existence of a j-envy-free matching.

A strong motivation also exists for combining these three solution concepts with a setting of fixed-size coalitions (and of size three). As we discussed in Chapter 2, in a model in which coalitions have arbitrary size, such as a hedonic game, it can be trivial to construct partitions that are envy-free by, for example, placing all agents into the grand coalition [16, 37].

We begin in Section 7.2 by showing that an arbitrary instance of 3DR-AS may not contain an envy-free matching, even when preferences are binary and symmetric and the maximum degree of the underlying graph is 2. We describe a polynomial-time algorithm for this case that can either construct an envy-free matching or report that no such matching exists (Theorem 7.1). We then contrast this result by showing that the corresponding existence problem is NP-complete even when the maximum degree of the underlying graph is 3 (Theorem 7.2).

In Section 7.3, we identify a similar dichotomy for wj-envy-freeness. We first show that a wj-envy-free matching may not exist, even when preferences are binary and symmetric and the maximum degree of the underlying graph is 2. Notably, the set of instances of 3DR-AS with maximum degree 2 that do not contain a wj-envy-free matching is a strict subset of the set of instances with maximum degree 2 that do not contain an envy-free matching. We describe a slightly more complex polynomial-time algorithm for this case, compared to the corresponding algorithm in Section 7.2, that can either construct a wj-envy-free matching or report that no such matching exists (Theorem 7.3). As for envy-freeness, we show that the corresponding existence problem is also NP-complete even when the maximum degree of the underlying graph is 3 (Theorem 7.4).

Next, in Section 7.4 we consider j-envy-freeness. We first remark that any stable matching is also j-envy-free. We then show that if preferences are binary but not necessarily symmetric,

a j -envy-free matching must exist and can be found in polynomial time (Theorem 7.5). Significantly, this tractability result holds in a larger set of instances of 3DR-AS than the corresponding result for stability, which we saw in Chapter 6. Nevertheless, we complement this result with two hardness results. The first is that a given instance of 3DR-AS may not contain a j -envy-free matching even when preferences are ternary but not symmetric, and the associated existence problem is NP-complete (Theorem 7.6). The second is that a given instance of 3DR-AS may not contain a j -envy-free matching even when preferences are non-binary and symmetric, and the associated existence problem is NP-complete (Theorem 7.7). Finally, in Section 7.5, we recap on our contribution and discuss some directions for future work.

7.2 Envy-freeness

7.2.1 Symmetric binary preferences with maximum degree two

Our first result is a necessary and sufficient condition for the existence of an envy-free matching in an instance (N, V) of 3DR-AS with binary and symmetric preferences and maximum degree 2. Since preferences are binary and symmetric, in the following proof we refer to the *underlying graph* (N, E) of the instance (N, V) . The underlying graph (N, E) is constructed such that $\{\alpha_i, \alpha_j\} \in E$ if and only if $v_{\alpha_i}(\alpha_j) = 1$, for any two agents $\alpha_i, \alpha_j \in N$.

Lemma 7.1. *Consider an instance of 3DR-AS with binary and symmetric preferences and maximum degree 2. Let P be the set of isolated agents, Q be the set of connected components of size $3k_1 - 2$ for any $k_1 > 1$, and \mathcal{R} be the set of connected components of size $3k_2 - 1$ for any $k_2 \geq 1$. An envy-free matching exists if and only if $2|Q| + |\mathcal{R}| \leq |P|$.*

Proof. Consider an arbitrary instance of 3DR-AS with binary and symmetric preferences, represented by its underlying graph (N, E) . Let \mathcal{S} be the set of connected components of size $3k_3$ for any $k_3 \geq 1$.

To show the first direction, suppose $2|Q| + |\mathcal{R}| \leq |P|$. We shall construct a matching M and demonstrate that it is envy-free. First, observe that if any agent has utility 1 or more then that agent is not envious, since the maximum degree of the underlying graph is 2.

Construct M as follows. First, consider each $S = (s_1, s_2, \dots, s_{3k_3})$ in \mathcal{S} . For each i where $1 \leq i \leq k_3$, add $\{s_{3i-2}, s_{3i-1}, s_{3i}\}$ to M . It follows that each agent in S has utility at least 1 and thus is not envious. Now consider each $R = (r_1, r_2, \dots, r_{3k_2-1})$ in \mathcal{R} . For each i where $1 \leq i \leq k_2 - 1$, add $\{r_{3i-2}, r_{3i-1}, r_{3i}\}$ to M . Next, add $\{r_{3k_2-2}, r_{3k_2-1}, p_{2|Q|+i}\}$ to M (recall that $|P| \geq 2|Q| + |\mathcal{R}|$). It follows that each agent in R is not envious. Now consider each

$Q = (q_1, q_2, \dots, q_{3k_1-2})$ in \mathcal{Q} . For each i where $1 \leq i \leq k_1 - 2$, add $\{q_{3i-2}, q_{3i-1}, q_{3i}\}$ to M . Next, add to M the triples $\{q_{3k_1-5}, q_{3k_1-4}, p_i\}$ and $\{q_{3k_1-3}, q_{3k_1-2}, p_{2i}\}$. It follows that each agent in Q has utility at least 1 and thus is not envious.

Finally, arbitrarily add the remaining agents in P to triples in M . Since these agents are isolated they are not envious.

To show the second direction, suppose for a contradiction that (N, E) has an envy-free matching M and $2|Q| + |\mathcal{R}| > |P|$. Since the degree of any agent in $\bigcup Q$ is at least 1, it must be that the utility of each agent in $\bigcup Q$ is at least 1. Similarly, the utility of each agent in $\bigcup \mathcal{R}$ must also be at least 1. It follows that any agent in M that has utility 0 belongs to P .

Now consider some $Q \in \mathcal{Q}$. By definition, $|Q| = 3k_1 - 2$ for some $k_1 > 1$. It follows that there exists two triples in M that each contain exactly two agents in Q and some agent with utility 0 in M . Similarly, for each $R \in \mathcal{R}$ there must exist at least one triple in M that contains exactly two agents in R and some agent with utility 0 in M . It follows that there are at least $2|Q| + |\mathcal{R}|$ agents with utility 0 in M . The only possibility is that there exists $2|Q| + |\mathcal{R}|$ agents in P , which is a contradiction. \square

Theorem 7.1. *Consider an instance of 3DR-AS with binary and symmetric preferences and maximum degree 2. There exists an $O(|N|)$ -time algorithm that can either find an envy-free matching or report that no such matching exists.*

Proof. Lemma 7.1 gives a necessary and sufficient condition for the existence of an envy-free matching in (N, E) , based on the number of connected components of different sizes. Define P , Q , and \mathcal{R} as before in Lemma 7.1. We outline a linear-time algorithm based on the constructive proof in Lemma 7.1. The algorithm outputs either \perp , if no envy-free matching exists, or a labelling τ of each agent α_i with some index $1 \leq \tau(\alpha_i) \leq n$ that represents the index of $M(\alpha_i)$ in some arbitrary ordering of the triples in an envy-free matching M .

The algorithm has three phases. In the first phase, the algorithm constructs a stack P that contains all isolated agents in (N, E) . It also constructs a stack T , which contains exactly one agent per connected component of size two or more, such that if a connected component is a path then T contains one of its endpoints. The construction of T can thus be performed in linear time.

The algorithm now enters the second phase, which is as follows. The algorithm maintains a counter r to track the label of the agent last labelled. Initially, $r = 1$. The algorithm pops an unmarked agent m_i from the stack T and marks $\tau(m_i) = 1$. It sets a counter c to 1, which will track the size of the connected component that contains m_i . It then identifies successive adjacent agents and labels each one with r , incrementing r by one every third agent, following the path or cycle in the underlying graph. The successive agents are therefore marked 1, 1, 1, 2, 2, 2, 3, 3, 3, ... The counter c is updated to ensure that c is the size of this connected

component. Eventually, either some agent with degree 1 or some previously labelled agent is discovered. In this case, there are three possibilities.

The first possibility is that $c = 3k_3$ for some $k_3 \geq 1$. In this case the algorithm pops some yet unlabelled m_i from the stack T and repeats the above process.

The second possibility is that $c = 3k_2 - 1$ for some $k_2 \geq 1$. In this case the algorithm pops some isolated agent p_i from the stack P and labels $\tau(p_i) = r$. If the stack P is empty then it must be that $2|Q| + |R| > |P|$ and thus the algorithm returns \perp . The algorithm then pops some yet unlabelled m_i from the stack T and repeats the above process.

The third possibility is that $c = 3k_1 - 2$ for some $k_1 > 1$. In this case it must be that exactly one agent α_j has been marked with r . The algorithm identifies the last agent α_l that was labelled with $r - 1$, which must be adjacent to α_j . It relabels $\tau(\alpha_l) = r$. It follows that exactly two adjacent agents are labelled with $r - 1$ and exactly two adjacent agents are labelled with r . The algorithm then pops two isolated agents p_g, p_h from the stack P and labels $\tau(p_g) = r$ and $\tau(p_h) = r - 1$. If $|P| < 2$ then it must be that $2|Q| + |R| > |P|$ and thus the algorithm returns \perp . The algorithm then pops some yet-unlabelled m_i from the stack T and repeats the above process.

In the third phase, since the algorithm has not yet returned \perp , it must be that each agent with degree 1 or more has been labelled and therefore assigned to some triple in M . The algorithm arbitrarily assigns the remaining agents in P to triples in M by popping successive agents p_i from the stack P and labelling $\tau(p_i) = r$, incrementing r every third agent. \square

7.2.2 Symmetric binary preferences with maximum degree three

We now consider instances of 3DR-AS with binary and symmetric preferences in which the maximum degree of the underlying graph is 3. In contrast with Theorem 7.1, we show that deciding if a given instance of 3DR-AS contains an envy-free matching is NP-complete, even when preferences are binary and symmetric and the maximum degree is 3.

We present a polynomial-time reduction from a variant of *Exact Satisfiability* (XSAT) [43]. An instance of XSAT is a boolean formula in conjunctive normal form (CNF). We write $C = \{c_1, c_2, \dots, c_m\}$ to represent such a formula as the set of its clauses. We represent each clause $c_r \in C$ as a set of literals. Each literal is either an occurrence of a single variable or its negation. Given a formula, we write X to mean the set of variables contained in the formula. A *truth assignment* $f : X \mapsto \{\text{true}, \text{false}\}$ is an assignment of values to the set of variables. We say that an *exact model* is a truth assignment to the variables such that each clause contains exactly one true literal (and therefore exactly two false literals). Given an instance C , if an exact model exists then we say that C is *exactly satisfiable*. Deciding if an instance C of XSAT is exactly satisfiable is NP-complete [87] even when each clause

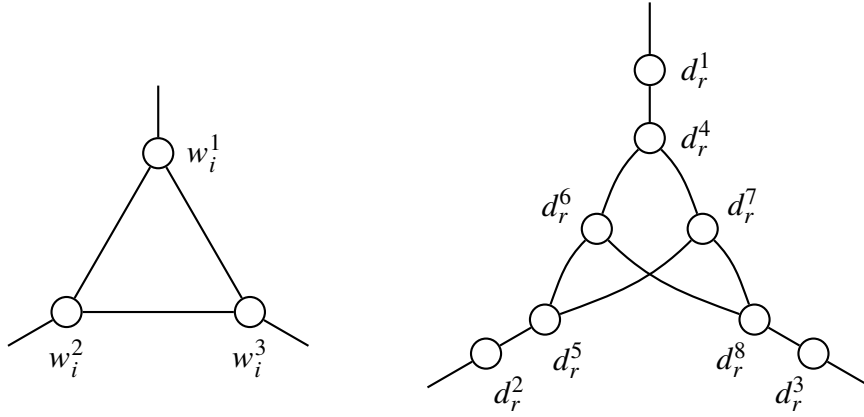


Figure 7.1: The reduction from $\text{X3SAT}_+^=3$ to the problem of deciding if a given instance of 3DR-AS contains an envy-free matching. A variable gadget W_i and clause gadget D_r are represented as undirected graphs.

contains exactly three literals. Porschen et al. [88, Lemma 5] show that this problem remains NP-complete even for the restricted case every literal is positive and each variable occurs in exactly three clauses, which they denote by $3\text{-CNF}_+^3\text{-XSAT}$. We shall refer to this variant as $\text{X3SAT}_+^=3$ (Problem 7.1). Note that in this problem $|X| = m$.

Problem 7.1. $\text{X3SAT}_+^=3$

Input: a boolean formula in conjunctive normal form, represented as a set of clauses $C = \{c_1, c_2, \dots, c_m\}$, in which every literal is positive and each variable occurs in exactly three clauses

Question: Is C exactly satisfiable?

The reduction, illustrated in Figure 7.1, is as follows. Suppose C is an arbitrary instance of $\text{X3SAT}_+^=3$. We shall construct an instance (N, E) of 3DR-AS.

For each variable $x_i \in X$, there are three corresponding literals in three different clauses. For each such x_i , arbitrarily label each of these literals as the first, second, and third occurrences of x_i . For each variable $x_i \in X$ construct a set of three agents $W_i = \{w_i^1, w_i^2, w_i^3\}$, which we refer to as the i^{th} variable gadget. Add the edges $\{w_i^1, w_i^2\}$, $\{w_i^2, w_i^3\}$, and $\{w_i^3, w_i^1\}$ to E . Next, for each clause $c_r \in C$ construct a set of eight agents $D_r = \{d_r^1, d_r^2, \dots, d_r^8\}$, which we refer to as the r^{th} clause gadget. Add the edges $\{d_r^1, d_r^4\}$, $\{d_r^2, d_r^5\}$, $\{d_r^3, d_r^8\}$, $\{d_r^4, d_r^6\}$, $\{d_r^5, d_r^6\}$, $\{d_r^6, d_r^7\}$, $\{d_r^7, d_r^8\}$, and $\{d_r^8, d_r^7\}$. Next, we shall connect the variable and clause gadgets. Consider each clause $c_r = \{x_i, x_j, x_k\}$. If c_r contains the first occurrence of x_i then add the edge $\{d_r^1, w_i^1\}$. Alternatively, if c_r contains the second occurrence of x_i then add the edge $\{d_r^1, w_i^2\}$. Alternatively, if c_r contains the third occurrence of x_i then add the edge $\{d_r^1, w_i^3\}$. Similarly, add an edge between d_r^3 and an agent in W_j depending on the index

of the occurrence of x_j in the clause c_r . Finally, add an edge between d_r^4 and an agent in W_k depending on the index of the occurrence of x_k in the clause c_r . We say that the clause gadget D_r is adjacent to the variable gadgets W_i , W_j , and W_k , and vice-versa.

This completes the construction of (N, E) . Note that each agent in a variable gadget has degree 3, the agents $d_r^4, d_r^5, d_r^6, d_r^7$ and d_r^8 for each $1 \leq r \leq m$ have degree 3, and the agents d_r^1, d_r^2, d_r^3 for each $1 \leq r \leq m$ have degree 2. It follows that the maximum degree of (N, E) is 3.

It is straightforward to show that this reduction can be performed in polynomial time. To prove that the reduction is correct we show that the 3DR-AS instance (N, E) contains an envy-free matching if and only if the X3SAT_+^3 instance C is exactly satisfiable.

We first prove some preliminary results. Recall that for any set of agents $S \subseteq N$, $\sigma(S, N)$ denotes the number of triples in N that each contain at least one agent in S .

Lemma 7.2. *Suppose M is a matching in (N, E) . If $u_{\alpha_i}(M) = 1$ and $\sigma(N(\alpha_i), M) = \deg(\alpha_i)$ then α_i is not envious in M .*

Proof. Suppose, to the contrary, that α_i envies some $\alpha_j \in N$. Then $u_{\alpha_i}(M(\alpha_j) \setminus \{\alpha_j\}) = 2$. It follows that two agents in $N(\alpha_i)$ belong to the same triple, $M(\alpha_j)$, so $\sigma(N(\alpha_i), M) < \deg(\alpha_i)$, which is a contradiction. \square

Lemma 7.3. *Suppose M is a matching in (N, E) . If $u_{\alpha_i}(M) = 0$ then α_i is envious in M .*

Proof. Note that by construction of (N, E) , each agent has degree at least one. Suppose then, for a contradiction, that there exists some $\alpha_i \in N$ where $u_{\alpha_i}(M) = 0$. There must exist some α_j where $\{\alpha_i, \alpha_j\} \in E$, so it follows that α_i envies both agents in $M(\alpha_j)$, which is a contradiction. \square

We now show that if the X3SAT_+^3 instance C is exactly satisfiable then the 3DR-AS instance (N, E) contains an envy-free matching.

Lemma 7.4. *If C is exactly satisfiable then (N, E) contains an envy-free matching.*

Proof. Suppose f is an exact model of C . We shall construct a matching M that is envy-free. For each x_i in X where $f(x_i)$ is false, add $\{w_i^1, w_i^2, w_i^3\}$ to M . Next, consider each clause $c_r = \{x_i, x_j, x_k\}$ and the corresponding clause gadget D_r , labelling i, j, k such that W_i contains an agent adjacent to d_r^1 , W_j contains an agent adjacent to d_r^2 , and W_k contains an agent adjacent to d_r^3 . There are three cases: $f(x_i)$ is true while both $f(x_j)$ and $f(x_k)$ are false, $f(x_j)$ is true while both $f(x_i)$ and $f(x_k)$ are false, and $f(x_k)$ is true while both $f(x_i)$ and $f(x_j)$ are false. In the first case, suppose c_r contains the u^{th} occurrence of x_i . Add the triples $\{w_i^u, d_r^1, d_r^4\}$, $\{d_r^2, d_r^5, d_r^7\}$, and $\{d_r^3, d_r^6, d_r^8\}$. The constructions in the second and third

cases are symmetric: in the second case, suppose c_r contains the u^{th} occurrence of x_j . Add the triples $\{w_j^u, d_r^2, d_r^5\}$, $\{d_r^1, d_r^4, d_r^6\}$, and $\{d_r^3, d_r^7, d_r^8\}$. In the third case, suppose c_r contains the u^{th} occurrence of x_k . Add the triples $\{w_k^u, d_r^3, d_r^8\}$, $\{d_r^1, d_r^4, d_r^6\}$, and $\{d_r^2, d_r^5, d_r^7\}$.

Now for any variable gadget W_i either $\sigma(W_i, M) = 1$ or $\sigma(W_i, M) = 3$. For each clause gadget D_r , there exist two triples in M that each contain three agents in D_r and one triple in M that contains two agents in D_r and one agent in some variable gadget. There are therefore three kinds of triple in M : those triples that contain three agents belonging to the same variable gadget; those triples that contain one agent in some variable gadget and two agents in a clause gadget; and those triples that contain three agents in the same clause gadget. We shall show that no triple of each kind contains an envious agent.

First, consider some triple $t \in M$ where $t = W_i$ for some variable gadget W_i . Since each agent in t has utility 2, no agent in t is envious.

Second, consider some triple $t \in M$ that contains one agent w_i^a in some variable gadget W_i and two agents in some clause gadget D_r . By the construction of M , either the triple contains $\{w_i^u, d_r^1, d_r^4\}$, $\{w_i^a, d_r^2, d_r^5\}$, or $\{w_i^a, d_r^3, d_r^8\}$. Suppose the triple contains $\{w_i^a, d_r^1, d_r^4\}$. Since $u_{d_r^1}(M) = 2$ clearly d_r^1 is not envious. By construction, $\sigma(W_i, M) = 3$ so since $u_{w_i^a}(M) \geq 1$ it follows by Lemma 7.2 that $w_i^u = a$ is also not envious. Similarly, since $M(d_r^6) \neq M(d_r^7)$ it follows that $\sigma(N(d_r^4), M) = 3$ so, by Lemma 7.2, d_r^4 is also not envious. The proof in the two remaining cases, in which the triple comprises $\{w_i^a, d_r^2, d_r^5\}$ or $\{w_i^a, d_r^3, d_r^8\}$, is symmetric.

Third, consider some triple $t \in M$ where $t \subset D_r$ for some clause gadget D_r . There are four cases: either $t = \{d_r^1, d_r^4, d_r^6\}$, $t = \{d_r^2, d_r^5, d_r^7\}$, $t = \{d_r^3, d_r^6, d_r^8\}$, or $t = \{d_r^3, d_r^7, d_r^8\}$. Suppose $t = \{d_r^1, d_r^4, d_r^6\}$. Since $\sigma(N(d_r^1), M) = 2$ and $u_{d_r^1}(M) = 1$ by Lemma 7.2 d_r^1 is not envious. Since $M(d_r^5) \neq M(d_r^8)$, it must be that $\sigma(N(d_r^6), M) = 3$ so since $u_{d_r^6}(M) = 1$, by Lemma 7.2 d_r^6 is not envious. Since $u_{d_r^4}(M) = 2$, d_r^4 is also not envious. The proof in the three remaining cases, in which $t = \{d_r^2, d_r^5, d_r^7\}$, $t = \{d_r^3, d_r^6, d_r^8\}$, or $t = \{d_r^3, d_r^7, d_r^8\}$, is symmetric. It follows that no agent in t is envious. \square

We now show that if the 3DR-AS instance (N, E) contains an envy-free matching then the $\text{X3SAT}_+^{\equiv 3}$ instance C is exactly satisfiable.

Lemma 7.5. *If (N, E) contains an envy-free matching M then for any variable gadget W_i , either $\sigma(W_i, M) = 1$ or $\sigma(W_i, M) = 3$.*

Proof. Since $|W_i| = 3$, clearly $1 \leq \sigma(W_i, M) \leq 3$. Suppose for a contradiction that $\sigma(W_i, M) = 2$. It must be that some triple in M contains exactly two agents in W_i and some third agent, which we label α_j . Label the remaining agent in the variable gadget w_i^a . Since the maximum degree of the instance is three, it must be that $u_{w_i^a}(M) \leq 1$. It follows that w_i^a envies α_j since $u_{w_i^a}(M(\alpha_j) \setminus \{\alpha_j\}) = 2$. \square

Lemma 7.6. *If (N, E) contains an envy-free matching then C is exactly satisfiable.*

Proof. Suppose M is an envy-free matching in (N, E) . By Lemma 7.5, for any variable gadget W_i either $\sigma(W_i, M) = 1$ or $\sigma(W_i, M) = 3$. Construct a truth assignment f in C by setting $f(x_i)$ to be true if $\sigma(W_i, M) = 3$ and false otherwise. Each variable x_i corresponds to exactly one variable gadget W_i so it follows that f is a valid truth assignment. By the construction of (N, E) , each clause c_r corresponds to exactly one clause gadget D_r . Each clause gadget is adjacent to three variable gadgets that correspond to the three variables in that clause. To show that f is an exact model of C , it is sufficient to show that for each clause gadget D_r there exists exactly one variable gadget W_i such that D_r is adjacent to W_i and $\sigma(W_i, M) = 3$.

Consider an arbitrary clause gadget D_r and the corresponding clause $c_r = \{x_i, x_j, x_k\}$, labelling i, j, k such that d_r^1 is adjacent to some agent in W_i , d_r^2 is adjacent to some agent in W_j and d_r^3 is adjacent to some agent in W_k . We shall show that a contradiction exists if either D_r is adjacent to two variable gadgets W_i, W_j where $\sigma(W_i, M) = \sigma(W_j, M) = 3$ or D_r is adjacent to three variable gadgets W_i, W_j, W_k where $\sigma(W_i, M) = \sigma(W_j, M) = \sigma(W_k, M) = 1$. The remaining possibility is that D_r is adjacent to exactly one variable gadget W_i where $\sigma(W_i, M) = 3$ and exactly two variable gadgets W_j, W_k where $\sigma(W_j, M) = \sigma(W_k, M) = 1$.

First, suppose that D_r is adjacent to two variable gadgets W_i and W_j where $\sigma(W_i, M) = 3$ and $\sigma(W_j, M) = 3$. Suppose c_r contains the a^{th} occurrence of x_i and the b^{th} occurrence of x_j . Consider $M(w_i^a)$. By Lemma 7.3, no agent has utility 0 in M , so either $\{d_r^1, d_r^4\} \in M(w_i^a)$, $\{d_r^2, d_r^5\} \in M(w_i^a)$, or $\{d_r^3, d_r^8\} \in M(w_i^a)$. Similarly, either $\{d_r^1, d_r^4\} \in M(w_j^b)$, $\{d_r^2, d_r^5\} \in M(w_j^b)$, or $\{d_r^3, d_r^8\} \in M(w_j^b)$. By the symmetry of the clause gadget, assume without loss of generality that $\{d_r^1, d_r^4\} \in M(w_i^a)$ and $\{d_r^2, d_r^5\} \in M(w_j^b)$. Consider d_r^6, d_r^7 and d_r^8 . Since no agent has utility 0, the only possibility is that $\{d_r^6, d_r^7, d_r^8\} \in M$. It then follows that d_r^4 envies d_r^8 , since $u_{d_r^4}(M) = 1$ and $u_{d_r^4}(\{d_r^6, d_r^7\}) = 2$, which is a contradiction.

Second, suppose that D_r is adjacent to three variable gadgets W_i, W_j, W_k where $\sigma(W_i, M) = \sigma(W_j, M) = \sigma(W_k, M) = 1$. Since $|D_r| = 8$, there exists triple in M that contains some agent $d_r^a \in D_r$ as well as some agent $\alpha_p \notin D_r$. It follows that either $u_{d_r^a}(M) = 0$ or $u_{\alpha_p}(M) = 0$, which contradicts Lemma 7.3. \square

We have now shown that the 3DR-AS instance (N, E) contains an envy-free matching if and only if the X3SAT_+^3 instance C is exactly satisfiable. This shows that the reduction is correct.

Theorem 7.2. *Deciding if a given instance of 3DR-AS contains an envy-free matching is NP-complete, even when preferences are binary and symmetric and the underlying graph has maximum degree 3.*

Proof. It is straightforward to show that this decision problem belongs to NP, since for any two agents $\alpha_i, \alpha_j \in N$ we can test if α_i envies α_j in constant time.

We have presented a polynomial-time reduction from $\text{X3SAT}_+^{\leq 3}$, which is NP-complete [88]. Given an arbitrary instance C of $\text{X3SAT}_+^{\leq 3}$, the reduction constructs an instance (N, E) of 3DR-AS with binary and symmetric preferences and maximum degree 3. Lemmas 7.4 and 7.6 show that (N, E) contains an envy-free matching if and only if C is exactly satisfiable and thus that this decision problem is NP-hard. \square

7.3 Weakly justified envy-freeness

7.3.1 Symmetric binary preferences with maximum degree two

We begin by defining a class of instances of 3DR-AS, \mathcal{I}^* , such that membership in the class is a necessary and sufficient condition for the non-existence of a wj-envy-free matching, among instances of 3DR-AS with binary and symmetric preferences and maximum degree 2.

Definition 7.1. \mathcal{I}^*

An instance of 3DR-AS belongs to \mathcal{I}^* if and only if the underlying graph comprises a set of disjoint 4-cycles and a single isolated agent.

We now show, using a sequence of lemmas, that any instance of 3DR-AS that belongs to \mathcal{I}^* does not contain a wj-envy-free matching. Consider an instance of 3DR-AS with binary and symmetric preferences that belongs to \mathcal{I}^* and label the underlying graph (\hat{N}, \hat{E}) . Suppose \hat{M} is an arbitrary matching in (\hat{N}, \hat{E}) .

Lemma 7.7. *For any 4-cycle R in (\hat{N}, \hat{E}) , if some triple in \hat{M} contains three agents in R then \hat{M} is not wj-envy-free.*

Proof. Suppose $R = (r_1, r_2, r_3, r_4)$. By definition, $\{r_1, r_2\}, \{r_2, r_3\}, \{r_3, r_4\}, \{r_4, r_1\} \in E$. Without loss of generality we need only consider the case when $\{r_1, r_2, r_3\} \in \hat{M}$. In this case it must be that r_4 has wj-envy for r_2 since $u_{r_4}(\hat{M}) = 0 < 2 = u_{r_4}(\{r_1, r_3\})$, $v_{r_1}(r_2) = 1 \leq 1 = v_{r_1}(r_4)$, and $v_{r_3}(r_2) = 1 \leq 1 = v_{r_3}(r_4)$. \square

Lemma 7.8. *For any three distinct connected components C_1, C_2, C_3 in (\hat{N}, \hat{E}) , if some triple in \hat{M} contains one agent in each of C_1, C_2 , and C_3 then \hat{M} is not wj-envy-free.*

Proof. Suppose some such triple in \hat{M} exists. By the construction of (\hat{N}, \hat{E}) , it must be that at least two of C_1, C_2 , and C_3 are 4-cycles. Assume without loss of generality that C_1 and C_2 are 4-cycles, so label C_1 as $R_1 = (r_1^1, r_1^2, r_1^3, r_1^4)$ and C_2 as $R_2 = (r_2^1, r_2^2, r_2^3, r_2^4)$. We may further

assume that this triple is $\{r_1^1, r_2^1, c_3^1\}$ where $r_1^1 \in C_1$, $r_2^1 \in C_2$, and $c_3^1 \in C_3$. Note that $\{r_1^1, c_3^1\} \notin E$ and $\{r_2^1, c_3^1\} \notin E$. Now consider R_1 . If $u_{r_1^2}(\hat{M}) = 0$ then r_1^2 has wj-envy for c_3^1 since $u_{r_1^2}(\hat{M}) = 0 < 1 = u_{r_1^2}(\{r_1^1, r_2^1\})$, $v_{r_1^2}(c_3^1) = 0 \leq 1 = v_{r_1^2}(r_1^2)$, and $v_{r_1^2}(c_3^1) = 0 \leq 0 = v_{r_1^2}(r_2^1)$. It follows that $u_{r_1^2}(\hat{M}) \geq 1$ so it must be that $\hat{M}(r_1^2)$ contains r_1^3 . A symmetric argument shows that $u_{r_1^4}(\hat{M}) \geq 1$ and thus that $\hat{M}(r_1^2)$ must also contain r_1^3 . Now $\hat{M}(r_1^2) = \{r_1^2, r_1^3, r_1^4\}$ so by Lemma 7.7 it follows that \hat{M} is not wj-envy-free. \square

Lemma 7.9. *For any triple $t \in \hat{M}$, if \hat{M} is wj-envy-free then the agents in t belong to exactly two connected components in (\hat{N}, \hat{E}) .*

Proof. Since t is a triple, the agents in t belong to either 1, 2, or 3 connected components. If \hat{M} is wj-envy-free, then by Lemmas 7.7 and 7.8 the agents in t do not belong to either 1 or 3 connected components in (\hat{N}, \hat{E}) . \square

Recall that for any set of agents $S \subseteq N$, $\sigma(S, N)$ denotes the number of triples in N that each contain at least one agent in S .

Lemma 7.10. *For any 4-cycle R in (\hat{N}, \hat{E}) , if \hat{M} is wj-envy-free then $\sigma(R, \hat{M}) \in \{2, 4\}$.*

Proof. By definition, $2 \leq \sigma(R, \hat{M}) \leq 4$ for any 4-cycle R in (\hat{N}, \hat{E}) . It suffices to show that if \hat{M} is wj-envy-free then $\sigma(R, \hat{M}) \neq 3$ for any such R . Suppose then, for a contradiction, that \hat{M} is wj-envy-free and there exists some 4-cycle R in (\hat{N}, \hat{E}) where $\sigma(R, \hat{M}) = 3$. Label $R = (r_1, r_2, r_3, r_4)$. Since $\sigma(R, \hat{M}) = 3$ there must exist one triple in \hat{M} that contains exactly two agents in R and two triples in \hat{M} that each contain exactly one agent in R . Label the former triple $\{r_{i_1}, r_{i_2}, \alpha_{j_1}\}$ and the latter two triples $\{r_{i_3}, \alpha_{j_2}, \alpha_{j_3}\}$ and $\{r_{i_4}, \alpha_{j_4}, \alpha_{j_5}\}$, where $\alpha_{j_1}, \alpha_{j_2}, \dots, \alpha_{j_5} \in N \setminus R$. Since R is a 4-cycle it must be that $\{r_{i_1}, \alpha_{j_1}\} \notin E$ and $\{r_{i_2}, \alpha_{j_1}\} \notin E$. Similarly, $\{r_{i_3}, \alpha_{j_2}\} \notin E$ and $\{r_{i_3}, \alpha_{j_3}\} \notin E$. It must also be that either $\{r_{i_3}, r_{i_1}\} \in E$ or $\{r_{i_3}, r_{i_2}\} \in E$. We can now see that r_{i_3} has wj-envy for α_{j_1} since $u_{r_{i_3}}(\hat{M}) = 0 < 1 \leq u_{r_{i_3}}(\{r_{i_1}, r_{i_2}\})$, $v_{r_{i_3}}(\alpha_{j_1}) = 0 \leq v_{r_{i_3}}(r_{i_3})$, and $v_{r_{i_3}}(\alpha_{j_1}) = 0 \leq v_{r_{i_3}}(r_{i_3})$. \square

Lemma 7.11. *If an instance of 3DR-AS belongs to \mathcal{I}^* then it does not contain a wj-envy-free matching.*

Proof. We considered an arbitrary instance of 3DR-AS that belongs to \mathcal{I}^* in which the preferences are binary and symmetric and its underlying graph (\hat{N}, \hat{E}) . We also supposed \hat{M} is an arbitrary matching of (\hat{N}, \hat{E}) .

Suppose for a contradiction that \hat{M} is wj-envy-free. Construct a graph (C, Γ) where C is the set of connected components in (\hat{N}, \hat{E}) and Γ is constructed as follows. For any triple $t \in \hat{M}$, it must be that the agents in t belong to exactly two connected components in (\hat{N}, \hat{E}) (Lemma 7.9). For each triple $t \in \hat{M}$ identify the two such connected components C_i, C_j in (\hat{N}, \hat{E}) and add the edge $\{C_i, C_j\}$ to Γ . By the design of (\hat{N}, \hat{E}) , there exists exactly one

connected component in (\hat{N}, \hat{E}) that is not a 4-cycle, which contains exactly one agent. Label this component C_1 . Label the remaining connected components in (\hat{N}, \hat{E}) , which are all 4-cycles, as $C_2, C_3, \dots, C_{|C|}$. Since $|C_1| = 1$ it must be that exactly one triple in \hat{M} contains the agent in C_1 so the degree of vertex C_1 in the graph (C, Γ) is 1. Consider the 4-cycles $C_2, C_3, \dots, C_{|C|}$. By Lemma 7.10, it must be that $\sigma(C_i, \hat{M}) \in \{2, 4\}$ for each such 4-cycle C_i (where $2 \leq i \leq |C|$). It follows that the degree of each vertex C_i in C where $2 \leq i \leq |C|$ is either 2 or 4. It follows that the sum of the degrees of all vertices in C is odd, which is impossible. \square

Building on Lemma 7.11, we present Algorithm `wjPathsCycles`, shown in Algorithm 7.5. Given an instance of 3DR-AS with binary and symmetric preferences and maximum degree 2, this algorithm either returns a wj-envy-free matching M or reports that (N, E) belongs to \mathcal{I}^* . With Lemma 7.11, this establishes the fact that \mathcal{I}^* is a necessary and sufficient condition for the non-existence of a wj-envy-free matching in instances of 3DR-AS with with binary and symmetric preferences and maximum degree 2.

In some respects the approach taken by Algorithm `wjPathsCycles` is straightforward. For example, paths or cycles that contain a number of agents divisible by 3 are broken up into triples of three successively adjacent agents. Other paths and cycles, except 4-cycles, are broken up in a similar fashion leaving one or two surplus agents per connected component. More care is required in the assignment of the agents in 4-cycles to triples. The 12 agents in three 4-cycles can be assigned to four triples in a relatively straightforward way that ensures no agent is wj-envied in some resulting matching. The main complexity of Algorithm `wjPathsCycles` stems from the case when the number of 4-cycles is not divisible by 3.

Algorithm `wjPathsCycles` contains calls to five subroutines, which are presented separately in order to simplify the overall presentation. Four of the subroutines take as input some agents in (N, E) and constructs a set of triples containing some or all of the agents in that set. The final subroutine is a helper function used to shorten the pseudocode of the main algorithm.

The first subroutine is Subroutine `nonC4Components`, shown in Algorithm 7.1. This subroutine takes as input a set of connected components C in (N, E) , none of which are 4-cycles. It returns a pair (T, S) where T is a set of triples of agents and S is a set of agents. For each component in C , the corresponding set of triples in T is constructed in a straightforward way by breaking up the component into triples of three successively adjacent agents. This procedure leaves remaining at most two agents from each component, which are then added to S . It follows that the maximum degree of the subgraph induced by S in (N, E) is 1.

Lemma 7.12. *Suppose C is some set of connected components in (N, E) that are not 4-cycles. Suppose (T, S) is returned by a call `nonC4Components(C)` and M is a matching in (N, E) . For any agent $c_i \in \bigcup T$, if $M(c_i) \in T$ then c_i is not wj-envious in M .*

Algorithm 7.1 Subroutine nonC4Components**Input:** a set of connected components C that are not 4-cycles**Output:** a pair (T, S) where T is a set of triples of agents in C and S is a set of agents

```

 $T \leftarrow \emptyset$ 
for each connected component  $C$  in  $C$ , labelling  $C = (c_1, c_2, \dots, c_{|C|})$  do
  for  $i = 1$  to  $\lfloor |C|/3 \rfloor$  do
     $T \leftarrow T \cup \{c_{3i-2}, c_{3i-1}, c_{3i}\}$ 
  end for
end for
 $S \leftarrow \bigcup C \setminus \bigcup T$ 
return  $(T, S)$ 

```

Proof. Suppose $c_i \in \bigcup T$ belongs to some connected component $C \in C$, which must not be a 4-cycle. By the construction of T in Subroutine nonC4Components, it must be that the triple in T that contains c_i either contains c_{i-1} or c_{i+1} . Since $M(c_i) \in T$ by assumption, it follows that $u_{c_i}(M) \geq 1$. If c_i has wj-envy in M then it must be that two agents not in $M(c_i)$ are adjacent to c_i in (N, E) . Since $u_{c_i}(M) \geq 1$ it follows that c_i has degree 3 in (N, E) , which is a contradiction. \square

Lemma 7.13. *Suppose C is a set of connected components in (N, E) that are not 4-cycles. Suppose (T, S) is returned by a call nonC4Components(C) and M is a matching in (N, E) . For any agent $c_j \in \bigcup T$, if $M(c_j) \in T$ then c_j is not wj-envied in M .*

Proof. Suppose for a contradiction that some $c_j \in \bigcup T$ where $M(c_j) \in T$ is wj-envied in M . Consider the pseudocode of Subroutine nonC4Components. Let $i = \lceil j/3 \rceil$. It must be that the triple $t = \{c_{3i-2}, c_{3i-1}, c_{3i}\}$, which contains c_j , was added to T in the i^{th} iteration of the inner for loop, in the particular iteration of the outer for loop in which component C was identified. Note that $\{c_{3i-2}, c_{3i-1}\} \in E$ and $\{c_{3i-1}, c_{3i}\} \in E$, by definition. There are now three possibilities: $j = 3i - 2$, $j = 3i - 1$, and $j = 3i$.

Suppose either $j = 3i - 2$ or $j = 3i$. Since α_k has wj-envy for c_j it must be that $v_{c_{3i-1}}(\alpha_k) \geq v_{c_{3i-1}}(c_j) = 1$. It follows that $v_{c_{3i-1}}(\alpha_k) = v_{c_{3i-1}}(c_{3i-2}) = v_{c_{3i-1}}(c_{3i}) = 1$ and thus that c_{3i-1} has degree 3 in (N, E) , which is a contradiction.

Suppose then that $j = 3i - 1$. Since α_k has wj-envy for c_j it must be that $v_{c_{3i-2}}(\alpha_k) \geq v_{c_{3i-2}}(c_{3i-1}) = 1$ and $v_{c_{3i}}(\alpha_k) \geq v_{c_{3i}}(c_{3i-1}) = 1$. It follows that $v_{c_{3i-2}}(\alpha_k) = v_{c_{3i}}(\alpha_k) = 1$. The only possibility is that C is a 4-cycle comprising $(c_{3i-2}, c_{3i-1}, c_{3i}, \alpha_k)$, which contradicts the statement of the lemma. \square

Lemma 7.14. *Subroutine nonC4Components terminates in $O(|\bigcup C|)$ time and returns a pair (T, S) where $\bigcup C = S \cup \bigcup T$.*

Proof. There are $|C|$ iterations of the outer for loop. In each iteration of the outer loop, some connected component C is identified, and the number of iterations of the inner for loop is $O(|C|)$. It follows that the total number of iterations of the inner for loop is $O(|\bigcup C|)$. In each iteration of the inner loop, a set of three agents is added to a set T , which can be performed in constant time, using an appropriate data structure for T . In each iteration of the outer loop, at most two agents are added to S , which can also be performed in constant time. It follows that the running time of Subroutine `nonC4Components` is $O(|\bigcup C|)$.

By the pseudocode, it is straightforward to show that $\bigcup C = S \cup \bigcup T$. \square

The second subroutine is Subroutine `oneC4TwoSingles`, shown in Algorithm 7.2. This subroutine takes as input three connected components in (N, E) . The first, R , is a 4-cycle in (N, E) . The second and third, w_1 and w_2 , are other agents in (N, E) . It returns two triples in C , each of which contains two agents in R and either w_1 or w_2 .

Algorithm 7.2 Subroutine `oneC4TwoSingles`

Input: a 4-cycle $R = (r_1, r_2, r_3, r_4)$ and two other agents w_1, w_2

Output: a set of two triples

return $\{\{w_1, r_1, r_2\}, \{w_2, r_3, r_4\}\}$

Lemma 7.15. *Consider an arbitrary 4-cycle R and two other arbitrary agents w_1, w_2 in (N, E) . Suppose T is returned by a call `oneC4TwoSingles`(R, w_1, w_2). If M is a matching in (N, E) where $T \subseteq M$ then no agent in $R \cup \{w_1, w_2\}$ is *wj-envied* in M .*

Proof. By the design of Subroutine `oneC4TwoSingles`, it must be that $T = \{\{w_1, r_1, r_2\}, \{w_2, r_3, r_4\}\}$, for some labelling of R where $R = (r_1, r_2, r_3, r_4)$. Suppose for a contradiction that some agent $\alpha_k \in N$ has *wj-envy* for some agent in $R \cup \{w_1, w_2\}$. By symmetry, we need only consider two cases: either α_k has *wj-envy* for r_1 or α_k has *wj-envy* for w_1 .

If α_k has *wj-envy* for r_1 then consider r_2 . Since $r_2 \in M(r_1)$ it must be that $v_{r_2}(\alpha_k) \geq v_{r_2}(r_1) = 1$ and thus that $v_{r_2}(\alpha_k) = 1$. The only possibility is that $\alpha_k = r_3$. This is a contradiction since $u_{r_3}(M) = 1 = u_{r_3}(\{r_1, w_2\})$, so r_3 does not have *wj-envy* for r_1 .

If α_k has *wj-envy* for w_1 then it must be that $u_{\alpha_k}(\{r_1, r_2\}) \geq 1$. The only possibility is that either $\alpha_k = r_3$ or $\alpha_k = r_4$. Since $u_{r_3}(M) = 1 = u_{r_3}(\{r_1, r_2\})$ and $u_{r_4}(M) = 1 = u_{r_4}(\{r_1, r_2\})$ it follows that neither r_3 nor r_4 have *wj-envy* for w_1 , which is a contradiction. \square

Lemma 7.16. *Subroutine `oneC4TwoSingles` terminates in constant time.*

Proof. A suitable choice of data structure for M allows the asymptotic running time of this subroutine to be $O(1)$. \square

The third subroutine is Subroutine `multipleOfThreeC4s`, shown in Algorithm 7.3. It takes as input a set \mathcal{R} of 4-cycles in (N, E) , where the number of 4-cycles is $3q$ for some integer $q \geq 1$. It returns $4q$ triples, each of which contains two agents in one 4-cycle and one agent in a different 4-cycle. The agents in each 4-cycle are assigned to either two or four triples.

Algorithm 7.3 Subroutine `multipleOfThreeC4s`

Input: a set \mathcal{R} of 4-cycles in (N, E) , where $|\mathcal{R}| = 3q$, labelled R_1, R_2, \dots, R_{3q} where $R_i = (r_i^1, r_i^2, r_i^3, r_i^4)$

Output: a set T of $4q$ triples

```

 $T \leftarrow \emptyset$ 
for  $d = 1$  to  $q$  do
     $T \leftarrow T \cup \{\{r_{3d-2}^1, r_{3d-2}^2, r_{3d-1}^1\}, \{r_{3d-2}^3, r_{3d-2}^4, r_{3d-1}^4\},$ 
         $\{r_{3d-1}^2, r_{3d}^1, r_{3d}^2\}, \{r_{3d-1}^3, r_{3d}^3, r_{3d}^4\}\}$ 
end for
return  $T$ 

```

Lemma 7.17. *Consider an arbitrary set \mathcal{R} of 4-cycles in (N, E) where $|\mathcal{R}| = 3q$ for some $q \geq 1$. Suppose T is returned by a call `multipleOfThreeC4s`(\mathcal{R}). If M is a matching in (N, E) where $T \subseteq M$ then no agent in $\bigcup \mathcal{R}$ is wj-envied in M .*

Proof. Suppose to the contrary that some agent in $\bigcup \mathcal{R}$ is wj-envied in M . By the design of Subroutine `multipleOfThreeC4s` it must be that some such agent was labelled r_j^i for some i where $1 \leq i \leq 4$ and j where $1 \leq j \leq 3q$. By the pseudocode of this subroutine, it must be that some triple containing r_j^i was added to T in the d' th iteration of the for loop, where $d' = \lceil j/3 \rceil$. We show that no agent in any of the four triples added to T in this iteration is wj-envied.

In fact, by the symmetric construction of the four triples in T in this d' th iteration of the loop, it suffices to consider only the triple $\{r_{3d'-2}^1, r_{3d'-2}^2, r_{3d'-1}^1\}$.

Suppose first that some agent $\alpha_k \in N$ has wj-envy for $r_{3d'-2}^1$. Since $v_{r_{3d'-2}^2}^{r_{3d'-2}^1}(\alpha_k) = 1$ it must be that $v_{r_{3d'-2}^2}^{r_{3d'-2}^1}(\alpha_k) = 1$. The only possibility is that $\alpha_k = r_{3d'-2}^3$. This is a contradiction since, by construction, $M(r_{3d'-2}^3) = \{r_{3d'-2}^3, r_{3d'-2}^4, r_{3d'-1}^4\}$ so $u_{r_{3d'-2}^3}(M) = 1 = u_{r_{3d'-2}^3}(\{r_{3d'-2}^2, r_{3d'-1}^1\})$ and thus $r_{3d'-2}^3$ does not have wj-envy for $r_{3d'-2}^1$. A symmetric argument shows that if some α_k has wj-envy for $r_{3d'-2}^2$ then it must be that $\alpha_k = r_{3d'-2}^4$, which leads to a contradiction since $u_{r_{3d'-2}^4}(M) = 1$.

Suppose finally that some agent $\alpha_k \in N$ has wj-envy for $r_{3d'-1}^1$. It follows that $u_{\alpha_k}(\{r_{3d'-2}^1, r_{3d'-2}^2\}) \geq 1$ so either $\alpha_k = r_{3d'-2}^3$ or $\alpha_k = r_{3d'-2}^4$. If $\alpha_k = r_{3d'-2}^3$ then by construction $M(r_{3d'-2}^3) = \{r_{3d'-2}^3, r_{3d'-2}^4, r_{3d'-1}^4\}$ and thus $u_{r_{3d'-2}^3}(M) = 1$. Since $u_{r_{3d'-2}^3}(M) = 1 = u_{r_{3d'-2}^3}(\{r_{3d'-2}^1, r_{3d'-2}^2\})$ it follows that $r_{3d'-2}^3$ does not in fact wj-envy $r_{3d'-1}^1$, which is a contradiction. A similar argument applies if $\alpha_k = r_{3d'-2}^4$. \square

Lemma 7.18. *Subroutine `multipleOfThreeC4s` terminates in $O(|\mathcal{R}|)$ time.*

Proof. Consider the for loop in the subroutine. In each iteration, four triples are added to T , which can be performed in constant time. Since there are exactly q iterations it follows that the running time of this subroutine is $O(|\mathcal{R}|)$. \square

The fourth subroutine is Subroutine `configureSurplusAgents`, shown in Algorithm 7.4. It takes as input a set of agents $\hat{S} \subseteq N$ of agents where $|\hat{S}|$ is divisible by three and the subgraph induced by \hat{S} in (N, E) has maximum degree 1. It returns a set T of $|\hat{S}|/3$ triples. In the context of Algorithm `wjPathsCycles`, this subroutine will be called with a subset of the surplus agents in the second element of the tuple returned by Subroutine `nonC4Components`. We remark that Subroutine `configureSurplusAgents` is essentially the same procedure as one used in Algorithm `findStableUW`, shown in Algorithm 6.4 in Chapter 6.

Algorithm 7.4 Subroutine `configureSurplusAgents`

Input: a set $\hat{S} \subseteq N$ of agents where $|\hat{S}|$ is divisible by three and the maximum degree of the subgraph induced by \hat{S} in (N, E) is 1

Output: a set of $|\hat{S}|/3$ triples

```

 $P \leftarrow$  the set of agents with degree 0 in the subgraph induced by  $\hat{S}$  in  $(N, E)$ , labelling
 $P = \{p_1, p_2, \dots, p_{|P|}\}$ 
 $Q \leftarrow$  a set containing each pair of agents  $\{q_i, q_j\} \subset \hat{S}$  where  $\{q_i, q_j\} \in E$ , labelling
 $Q = \{Q_1, Q_2, \dots, Q_{|Q|}\}$ 
 $\mathcal{X} \leftarrow \emptyset$ 
if  $|Q| \geq |\hat{S}|/3$  then
   $\mathcal{X} \leftarrow \{q_1, q_2, \dots, q_{|\hat{S}|/3}\}$ 
else
   $\triangleright$  note that  $|P| > 2(|\hat{S}|/3 - |Q|)$  since by definition  $|P| = |\hat{S}| - 2|Q|$ 
   $\mathcal{W} \leftarrow \{\{p_i, p_{2i}\} : 1 \leq i \leq |\hat{S}|/3 - |Q|\}$ 
   $\mathcal{X} \leftarrow Q \cup \mathcal{W}$ 
end if
 $Y \leftarrow \hat{S} \setminus \bigcup \mathcal{X}$ 
 $\triangleright$  Suppose  $\mathcal{X} = \{X_1, X_2, \dots, X_{|\hat{S}|/3}\}$  and  $Y = \{y_1, y_2, \dots, y_{|\hat{S}|/3}\}$ . Note that  $\mathcal{X}$  is a set
  of pairs of agents and  $Y$  is a set of individual agents.
return  $\{X_i \cup \{y_i\} : 1 \leq i \leq |\hat{S}|/3\}$ 

```

Lemma 7.19. *Consider an arbitrary set $\hat{S} \subseteq N$ where 3 divides $|\hat{S}|$ and the maximum degree of the subgraph induced by \hat{S} in (N, E) is 1. Suppose T is returned by a call `configureSurplusAgents`(\hat{S}). If M is a matching in (N, E) where $T \subseteq M$ then no agent in \hat{S} has *wj-envy* in M for any other agent in \hat{S} .*

Proof. From the pseudocode, it is straightforward to show that Subroutine `configureSurplusAgents` is bound to terminate and must return a set T of disjoint triples where $\bigcup T = \hat{S}$.

Suppose for a contradiction that some agent $\alpha_{j_1} \in \hat{S}$ has wj-envy for an agent $\alpha_{k_1} \in \hat{S}$ where $M(\alpha_{j_1}) = \{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$ and $M(\alpha_{k_1}) = \{\alpha_{k_1}, \alpha_{k_2}, \alpha_{k_3}\}$. It follows that $u_{\alpha_{j_1}}(\{\alpha_{k_2}, \alpha_{k_3}\}) \geq 1$. Without loss of generality assume that $v_{\alpha_{j_1}}(\alpha_{k_2}) = 1$, or equivalently that $\{\alpha_{j_1}, \alpha_{k_2}\} \in E$. By the definition of \mathcal{Q} , it must be that $\{\alpha_{j_1}, \alpha_{k_2}\} \in \mathcal{Q}$.

Note that if $|\mathcal{Q}| < |\hat{S}|/3$ then, by the pseudocode, for every $\{q_a, q_b\} \in \mathcal{Q}$ it must be that $q_a \in M(q_b)$. Since $\{\alpha_{j_1}, \alpha_{k_2}\} \in \mathcal{Q}$ and $\alpha_{j_1} \notin M(\alpha_{k_2})$ it follows that $|\mathcal{Q}| \geq |\hat{S}|/3$.

By the pseudocode, for each triple $r \in T$ there exists some pair $\{q_a, q_b\} \in \mathcal{Q}$ where $\{q_a, q_b\} \subset r$. Since \mathcal{Q} is agent-disjoint (and we established that $\{\alpha_{j_1}, \alpha_{k_2}\} \in \mathcal{Q}$) the only possibility is that $\{\alpha_{k_1}, \alpha_{k_3}\} \in \mathcal{Q}$. By the definition of \mathcal{Q} it must be that $v_{\alpha_{k_1}}(\alpha_{k_3}) = 1$. Since α_{j_1} has wj-envy for α_{k_1} it must be that $v_{\alpha_{k_3}}(\alpha_{j_1}) \geq v_{\alpha_{k_3}}(\alpha_{k_1})$ so it follows that $v_{\alpha_{k_3}}(\alpha_{j_1}) = 1$. Now $\{\alpha_{k_3}, \alpha_{k_1}\} \in E$ and $\{\alpha_{k_3}, \alpha_{j_1}\} \in E$ so α_{k_3} has degree 2 in the subgraph induced by \hat{S} in (N, E) , which is a contradiction. \square

Lemma 7.20. *Subroutine `configureSurplusAgents` terminates in $O(|\hat{S}|)$ time.*

Proof. The sets of pairs \mathcal{X} and \mathcal{W} , the set of agents Y , and the returned set of triples T can all be constructed in $O(|\hat{S}|)$ time. \square

The fifth subroutine is Subroutine `pickLowDegree`. This subroutine takes as input a set S and integer $k \geq 1$, such that the maximum degree of the subgraph induced by S in (N, E) is 1. It returns a set of k agents in S such that the sum of the degrees of the agents returned in the subgraph induced by S in (N, E) is minimised. Since the maximum degree of the subgraph induced by S in (N, E) is 1, this subroutine can be implemented to run in $O(|N|)$ time.

We now present Algorithm `wjPathsCycles`, shown in Algorithm 7.5. The overall strategy of this algorithm is as follows. First Subroutine `nonC4Components` is used to break up connected components that are not 4-cycles into a set of triples T , in which each triple contains three successively adjacent agents, and a set S of surplus agents. The algorithm then constructs a set \mathcal{R} of all 4-cycles in (N, E) . If $|\mathcal{R}|$ is divisible by three, Subroutine `multipleOfThreeC4s` is called and all agents in $\bigcup \mathcal{R}$, i.e. all agents belonging to 4-cycles, are assigned to triples in M . If $|\mathcal{R}|$ is not divisible by three then there are two cases. In the first, the instance is identified as belonging to \mathcal{I}^* . In the second, a set of surplus agents in S are used to assign the agents belonging to either one or two 4-cycles to triples in M . This set is chosen using Subroutine `pickLowDegree`, which (as we shall see later) ensures that no agent in this set will be wj-envious in M . Next, the remaining 4-cycles (the number of which is divisible by three) are then added to triples in M using Subroutine `multipleOfThreeC4s`. The final step (in the case in which the instance does not belong to \mathcal{I}^*) is a call to Subroutine `configureSurplusAgents` and the assignment of all remaining agents in S to triples in M .

Proposition 7.1. *In Algorithm `wjPathsCycles`, $S \cup \bigcup T$ is the set of agents that do not belong to 4-cycles in (N, E) .*

Algorithm 7.5 Algorithm `wjPathsCycles`

Input: an instance (N, E) of 3DR-AS with binary and symmetric preferences and maximum degree 2

Output: either a wj -envy-free matching M or “belongs to \mathcal{I}^* ”

```

 $M \leftarrow \emptyset; \hat{T} \leftarrow \emptyset; \hat{S} \leftarrow \emptyset$ 
 $C \leftarrow$  the set of all connected components in  $(N, E)$  that are not 4-cycles
 $(T, S) \leftarrow \text{nonC4Components}(C)$ 
 $\mathcal{R} \leftarrow$  the set of 4-cycles in  $(N, E)$ , labelling  $\mathcal{R} = \{R_1, R_2, \dots, R_{|\mathcal{R}|}\}$ 
 $l \leftarrow 0$ 
if  $|\mathcal{R}| \bmod 3 = 2$  then
  if  $|S| \geq 4$  then
     $\{w_1, w_2, w_3, w_4\} \leftarrow \text{pickLowDegree}(S, 4)$ 
     $\hat{S} \leftarrow S \setminus \{w_1, w_2, w_3, w_4\}$ 
     $\hat{T} \leftarrow T$ 
  else if  $|T| \geq 1$  then
     $\triangleright$  note that  $|S| = 1$  by Proposition 7.2
     $w_1 \leftarrow$  the agent in  $S$ 
     $\hat{t} \leftarrow$  some triple in  $T$ 
     $\{w_2, w_3, w_4\} \leftarrow \hat{t}$ 
     $\hat{S} \leftarrow \emptyset$ 
     $\hat{T} \leftarrow T \setminus \hat{t}$ 
  else
    return “ $(N, E)$  belongs to  $\mathcal{I}^*$ ”
  end if
   $M \leftarrow M \cup \text{oneC4TwoSingles}(R_1, w_1, w_2) \cup \text{oneC4TwoSingles}(R_2, w_3, w_4)$ 
   $l \leftarrow 2$ 
else if  $|\mathcal{R}| \bmod 3 = 1$  then
   $\triangleright$  note that  $|S| \geq 2$  by Proposition 7.3
   $\{w_1, w_2\} \leftarrow \text{pickLowDegree}(S, 2)$ 
   $\hat{S} \leftarrow S \setminus \{w_1, w_2\}$ 
   $\hat{T} \leftarrow T$ 
   $M \leftarrow M \cup \text{oneC4TwoSingles}(R_1, w_1, w_2)$ 
   $l \leftarrow 1$ 
else
   $\triangleright$  it must be that  $|\mathcal{R}| \bmod 3 = 0$ 
   $\hat{S} \leftarrow S$ 
   $\hat{T} \leftarrow T$ 
end if
   $\triangleright$  it must be that 3 divides  $(|\mathcal{R}| - l)$ 
   $\mathcal{R}' \leftarrow \{R_{l+1}, R_{l+2}, \dots, R_{|\mathcal{R}|}\}$ 
   $M \leftarrow M \cup \text{multipleOfThreeC4s}(\mathcal{R}') \cup \hat{T} \cup \text{configureSurplusAgents}(\hat{S})$ 
return  $M$ 

```

Proof. This follows immediately by Lemma 7.14. \square

We first prove two propositions that show that, in two specific cases, S is large enough to extract the number of agents required.

Proposition 7.2. *In Algorithm `wjPathsCycles`, after initialising \mathcal{R} , if $|\mathcal{R}| \bmod 3 = 2$ and $|S| < 4$ then $|S| = 1$.*

Proof. Suppose $|\mathcal{R}| \bmod 3 = 2$ and $|S| < 4$ after initialising \mathcal{R} . Then there exists some constant $k_1 \geq 0$ such that $|\mathcal{R}| = 3k_1 + 2$, so the number of agents in N that belong to 4-cycles is $4|\mathcal{R}| = 12k_1 + 8$. It follows that the number of agents in N that do not belong to 4-cycles is $3n - 12k_1 - 8$. Since $(3n - 12k_1 - 8) \bmod 3 = 1$ there exists some constant $k_2 \geq 0$ such that the number of agents in N that do not belong to 4-cycles is $3k_2 + 1$. By Proposition 7.1, $S \cup \bigcup T$ is the set of agents that do not belong to 4-cycles. Since $|S \cup \bigcup T| = 3k_2 + 1$, T is a set of disjoint triples, and $|S| < 4$, it must be that $k_2 = 0$ and $|S| = 1$. \square

Proposition 7.3. *In Algorithm `wjPathsCycles`, after initialising \mathcal{R} , if $|\mathcal{R}| \bmod 3 = 1$ then $|S| \geq 2$.*

Proof. Suppose $|\mathcal{R}| \bmod 3 = 1$ after initialising \mathcal{R} . Then there exists some constant $k_1 \geq 0$ such that $|\mathcal{R}| = 3k_1 + 1$, so the number of agents in N that belong to 4-cycles is $4|\mathcal{R}| = 12k_1 + 4$. It follows that the number of agents in N that do not belong to 4-cycles is $3n - 12k_1 - 4$. Since $(3n - 12k_1 - 4) \bmod 3 = 2$ there exists some constant $k_2 \geq 0$ where the number of agents in N that do not belong to 4-cycles is $3k_2 + 2$. By Proposition 7.1, $S \cup \bigcup T$ is the set of agents that do not belong to 4-cycles. Since $|S \cup \bigcup T| = 3k_2 + 2$ and T is a set of disjoint triples it must be that $|S| \geq 2$. \square

We now show that Algorithm `wjPathsCycles` is bound to terminate and has a linear running time with respect to the number of agents.

Lemma 7.21. *Algorithm `wjPathsCycles` terminates in $O(|N|)$ time.*

Proof. The pseudocode describes the algorithm at a high level. To analyse the worst-case asymptotic time complexity we describe one possible system of data structures and analyse the algorithm with respect to the number of basic operations on these data structures. We begin the analysis at the start of the pseudocode.

The initialisation of M , \hat{T} and \hat{S} can be performed in constant time. The set of connected components C that are not 4-cycles can be identified in $O(|N|)$ time using breadth-first search, since the maximum degree of (N, E) is two.

By Lemma 7.14, the call to Subroutine `nonC4Components` takes $O(|\bigcup C|) = O(|N|)$ time.

Like C , the set of connected components \mathcal{R} that are 4-cycles can be constructed in $O(|N|)$ time. Each nested branch of the if/else statement involves removing a constant number of elements from S , at most two calls to Subroutine `oneC4TwoSingles` (which has constant running time by Lemma 7.18), and an assignment to \hat{T} and \hat{S} (which can be performed in $O(|N|)$ time). It follows that the total running time of the if/else statement is $O(|N|)$.

By Lemma 7.18, Subroutine `multipleOfThreeC4s` has $O(|\mathcal{R}|) = O(|N|)$ running time. By Lemma 7.20, Subroutine `configureSurplusAgents` has $O(|S|) = O(|N|)$ running time. It follows that the asymptotic worst-case running time of Algorithm `wjPathsCycles` is $O(|N|)$. \square

Having established that Algorithm `wjPathsCycles` is bound to terminate, we prove its correctness using a sequence of lemmas. First we show that if (N, E) belongs to \mathcal{I}^* then the algorithm correctly identifies it as such.

Lemma 7.22. *If (N, E) belongs to \mathcal{I}^* then Algorithm `wjPathsCycles` returns “ (N, E) belongs to \mathcal{I}^* ”.*

Proof. Suppose (N, E) belongs to \mathcal{I}^* . In the algorithm, the set of connected components C that are not 4-cycles contains exactly one element C_1 where C_1 contains a single agent c_1 in (N, E) . By Lemma 7.14, Subroutine `nonC4Components` must return $(\emptyset, \{c_1\})$ so $T = \emptyset$ and $S = \{c_1\}$. Consider the outermost if/else statement in the algorithm. By Definition 7.1, it must be that $3n = 4|\mathcal{R}| + 1$ so $4|\mathcal{R}| + 1 \bmod 3 = 0$. This implies that $4|\mathcal{R}| + 4 \bmod 3 = 0$ so $4(|\mathcal{R}| + 1) \bmod 3 = 0$. It follows that that $|\mathcal{R}| + 1 \bmod 3 = 0$ and thus that $|\mathcal{R}| \bmod 3 = 2$. It follows that the algorithm enters the first branch of the outermost if/else statement. Since $|S| = 1 < 4$ and $T = \emptyset$ the algorithm must then return “ (N, E) belongs to \mathcal{I}^* ”. \square

We now consider the case in which (N, E) does not belong to \mathcal{I}^* . We first show that in this case Algorithm `wjPathsCycles` returns a matching.

Lemma 7.23. *If (N, E) does not belong to \mathcal{I}^* then Algorithm `wjPathsCycles` returns a matching M .*

Proof. Consider an arbitrary connected component C in (N, E) . We show that each agent in C is added to exactly one triple in M .

Suppose C is not a 4-cycle. By the design of Algorithm `wjPathsCycles`, exactly one call is made to Subroutine `nonC4` with argument C . Consider an arbitrary agent $c_i \in C$. There are two cases: either $i \leq \lfloor |C|/3 \rfloor$ or $i > \lfloor |C|/3 \rfloor$. In the former case, exactly one triple containing c_i is added to T in Subroutine `nonC4`, which is then added to M in the main algorithm. In the latter case, c_i is eventually added to S . We can see from the pseudocode of

Subroutine `configureSurplusAgents` that c_i is therefore eventually added to exactly one triple in M .

Suppose C is a 4-cycle, so by definition $C \in \mathcal{R}$. If $C \in \mathcal{R}'$ then each agent in C is added to exactly one triple in M in some call to Subroutine `multipleOfThreeC4s`. If $C \notin \mathcal{R}'$ then some call to Subroutine `oneC4TwoSingles` was made with the first argument equal to C and the returned set of two triples was then added to M . It follows that each agent in each 4-cycle is added to exactly one triple in M . \square

We now show that if (N, E) does not belong to \mathcal{I}^* then the algorithm returns a matching M that is wj-envy-free. In the next four lemmas we consider subsets of N and show that in each subset no agent is wj-envied in M . The results of these lemmas are then combined in Lemma 7.28, in which we show that if the algorithm returns a matching M then M is wj-envy-free.

Lemma 7.24. *If Algorithm `wjPathsCycles` returns a matching M then no agent in $\bigcup T$ is wj-envied in M .*

Proof. Suppose Algorithm `wjPathsCycles` has returned some matching M . Consider an arbitrary triple $t \in T$. By the pseudocode of Algorithm `wjPathsCycles` there are two possibilities: either $t \in \hat{T}$ or t was labelled \hat{t} . If $t \in \hat{T}$ then by Lemma 7.13 no agent in t is wj-envied in M . Suppose then that t was labelled \hat{t} . By the pseudocode of Algorithm `wjPathsCycles`, for any agent c_i in \hat{t} it must be that some call was made to Subroutine `oneC4TwoSingles` in which the second or third argument was equal to c_i and then the two triples returned by the subroutine were added to M . By Lemma 7.15, it follows that no agent in \hat{t} is wj-envied in M . \square

Lemma 7.25. *If Algorithm `wjPathsCycles` returns a matching M then no agent in \hat{S} is wj-envied in M .*

Proof. Suppose Algorithm `wjPathsCycles` has returned some matching M in which some agent $\alpha_i \in N$ has wj-envy for some $\hat{s}_{j_1} \in \hat{S}$. By the pseudocode, it must be that $M(\hat{s}_{j_1})$ contains three agents in \hat{S} so we label $M(\hat{s}_{j_1}) = \{\hat{s}_{j_1}, \hat{s}_{j_2}, \hat{s}_{j_3}\}$. Note that since $|\hat{S}| > 0$ it must be that $\hat{T} = T$.

Since α_i has wj-envy for \hat{s}_{j_1} it must be that $u_{\alpha_i}(\{\hat{s}_{j_2}, \hat{s}_{j_3}\}) \geq 1$ so without loss of generality assume that $\{\alpha_i, \hat{s}_{j_2}\} \in E$. We now consider two possibilities: $\alpha_i \in S$ and $\alpha_i \notin S$.

First, suppose $\alpha_i \in S$. If $\alpha_i \in \hat{S}$ then Lemma 7.19 is contradicted, so it must be that $\alpha_i \in S \setminus \hat{S}$. By the pseudocode, α_i was labelled either w_1 , w_2 , w_3 or w_4 during algorithm execution and must belong to some set of agents returned by a call to Subroutine `pickLowDegree`. Since $\{\alpha_i, \hat{s}_{j_2}\} \subset S$ the degree of α_i in the subgraph induced by S in (N, E) is 1. By the definition of Subroutine `pickLowDegree` it must be that the degree of each agent in \hat{S} is also 1. With this

in mind, consider the call `configureSurplusAgents(\hat{S})`. It must be that $|Q| = |\hat{S}|/2$. It follows that $X \subset Q$. By the pseudocode of Subroutine `configureSurplusAgents`, We shall consider the values of the variables in Subroutine `configureRemainingAgents` inside this call. It must be that for each triple in the set of triples returned by this subroutine contains two agents that are adjacent in (N, E) . If $\{\hat{s}_{j_2}, \hat{s}_{j_1}\} \in E$ or $\{\hat{s}_{j_2}, \hat{s}_{j_3}\} \in E$ then the degree of \hat{s}_{j_2} in the subgraph induced by S in (N, E) is 2, which is a contradiction. It remains that $\{\hat{s}_{j_1}, \hat{s}_{j_3}\} \in E$. Since w_k has wj-envy for \hat{s}_{j_1} it must be that $v_{\hat{s}_{j_3}}(\hat{s}_{k_1}) \geq v_{\hat{s}_{j_3}}(\hat{s}_{j_1}) = 1$. It follows that \hat{s}_{j_3} has degree 2 in the subgraph induced by S in (N, E) , which is a contradiction.

Second, suppose $\alpha_i \notin S$. Since $\{\alpha_i, \hat{s}_{j_2}\} \in E$, by the pseudocode of Algorithm `wjPathsCycles` it must be that α_i belongs to the same connected component in (N, E) as \hat{s}_{j_2} . Since $\hat{s}_{j_2} \in S$ it must be that the connected component that contains α_i and \hat{s}_{j_2} is not a 4-cycle and belongs to C . Since $\alpha_i \notin S$ it must also be that $M(\alpha_i) \in T$. Since $\hat{T} = T$ it follows that $M(\alpha_i) \in \hat{T}$. Lemma 7.12 now implies that α_i is not wj-envy in M , which is a contradiction. \square

Lemma 7.26. *If Algorithm `wjPathsCycles` returns a matching M then no agent in S is wj-envied in M .*

Proof. Suppose Algorithm `wjPathsCycles` has returned some matching M . Consider an arbitrary agent $s_i \in S$. If $s_i \in \hat{S}$ then by Lemma 7.25 it must be that s_i is not wj-envied in M . Suppose then that $s_i \notin \hat{S}$. There are three cases: either $|\mathcal{R}| \bmod 3 = 2$, $|S| \geq 4$, and s_i was labelled w_1, w_2, w_3 , or w_4 ; $|\mathcal{R}| \bmod 3 = 2$, $|T| \geq 1$, and s_i was labelled w_1 ; or $|\mathcal{R}| \bmod 3 = 1$ and s_i was labelled either w_1 or w_2 . In each of the three cases, some call was then made to Subroutine `oneC4TwoSingles` in which the second or third argument was equal to s_i and then two triples returned by the subroutine were added to M . By Lemma 7.15 it follows that s_i is not wj-envied in M . \square

Lemma 7.27. *If Algorithm `wjPathsCycles` returns a matching M then no agent in $\bigcup \mathcal{R}$, i.e. in some 4-cycle in (N, E) , is wj-envied in M .*

Proof. Consider an arbitrary $R_j \in \mathcal{R}$ where $1 \leq j \leq |\mathcal{R}|$. We show that no agent in R_j is wj-envied in M . In this case, let l' be the final value assigned to the variable l before the algorithm terminated. There are two possibilities: either $j > l'$ or $j \leq l'$.

Suppose $j > l'$. It must be that $R_j \in \mathcal{R}'$, by the construction of \mathcal{R}' in Algorithm `wjPathsCycles`. By Lemma 7.17 it follows that no agent in R_j is wj-envied in M .

Suppose $j \leq l'$. By the design of Algorithm `wjPathsCycles` there are two possibilities: either $|\mathcal{R}| \bmod 3 = 2$ and $l' = 2$, or $|\mathcal{R}| \bmod 3 = 1$ and $l' = 1$. In either case, by the pseudocode of Algorithm `wjPathsCycles` it must be that some call to Subroutine `oneC4TwoSingles` was made with the first argument equal to R_j and the returned set of two triples was then added to M . It follows by Lemma 7.15 that no agent in R_j is wj-envied in M . \square

Lemma 7.28. *If (N, E) does not belong to I^* then Algorithm `wjPathsCycles` returns a matching M that is *wj-envy-free*.*

Proof. By definition, $C \cup \mathcal{R}$ is the set of all connected components in (N, E) . By Lemma 7.23, Algorithm `wjPathsCycles` returns a matching M in N . By Proposition 7.1, the set of agents $\bigcup C = S \cup \bigcup T$. By Lemma 7.24, no agent in $\bigcup T$ is *wj-envied* in M . By Lemma 7.26, no agent in S is *wj-envied* in M . By Lemma 7.27, no agent in $\bigcup \mathcal{R}$ is *wj-envied* in M . \square

We now prove our main theorem on *wj-envy-free* matchings and instances with binary and symmetric preferences and maximum degree 2.

Theorem 7.3. *Consider an instance of 3DR-AS with binary and symmetric preferences and maximum degree 2. There exists an $O(|N|)$ -time algorithm that can either find a *wj-envy-free* matching in the instance or report that the instance belongs to I^* , and thus contains no *wj-envy-free* matching.*

Proof. Lemma 7.21 shows that Algorithm `wjPathsCycles` terminates in $O(|N|)$ time. Lemmas 7.22 and 7.28 establish the correctness of this algorithm and show that the algorithm either returns a *wj-envy-free* matching or reports that “ (N, E) belongs to I^* ”. In the latter case, Lemma 7.11 shows that the supplied instance contains no *wj-envy-free* matching. \square

7.3.2 Symmetric binary preferences with maximum degree three

As before in Theorem 7.2, in this section we consider instances of 3DR-AS with binary and symmetric preferences and maximum degree 3. Here we show that, in contrast with Theorem 7.3, deciding if a given instance of 3DR-AS contains a *wj-envy-free* matching is NP-complete, even when preferences are binary and symmetric and the maximum degree is 3.

As before in Section 7.2.2, we reduce from $X3SAT_+^3$ (Problem 7.1). Here we assume that the number of clauses m satisfies $m = 4l$ for some $l \geq 1$. We can show that the *wj-envy-free* existence problem remains NP-complete under this restriction as follows. Construct four distinct copies of the set of variables $X(C)$ and formula C . Let the constructed formula C' be the union of the four copies of C . It is straightforward to show that C' is exactly satisfiable if and only if each of the four copies is exactly satisfiable, which is true if and only if the original formula C is exactly satisfiable.

The main difference between this reduction and the reduction for the 3DR-AS *envy-free* decision problem is in the second direction, in which we show that C is exactly satisfiable if a *wj-envy-free* matching M exists in the constructed instance (N, E) . Here we associate true literals with variable gadgets that belong to one triple (rather than three) and false literals

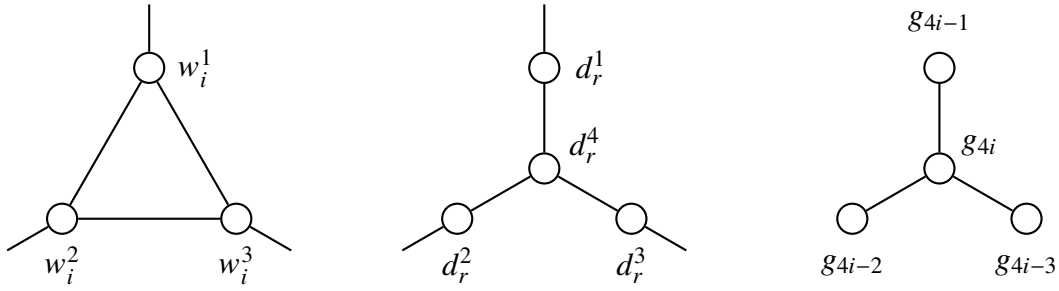


Figure 7.2: The reduction from X3SAT_+^3 to the problem of deciding if a given instance of 3DR-AS contains an wj-envy-free matching. A variable gadget W_i , clause gadget D_r , and garbage collector gadget G_i are represented as undirected graphs.

with variable gadgets that belong to three triples (rather than one). Another difference is that we construct a number of garbage collector gadgets. We will show that there is only one possible configuration of the triples of agents in the garbage collector gadgets in a wj-envy-free matching.

The reduction, illustrated in Figure 7.2, is as follows. Suppose C is an arbitrary instance of X3SAT_+^3 . We shall construct an instance (N, E) of 3DR-AS.

For each variable $x_i \in X(C)$ construct a set of three agents $W_i = \{w_i^1, w_i^2, w_i^3\}$, which we refer to as the i^{th} *variable gadget*. Add the edges $\{w_i^1, w_i^2\}$, $\{w_i^2, w_i^3\}$, and $\{w_i^3, w_i^1\}$ to E . Next, for each clause c_r in C construct a set of four agents $D_r = \{d_r^1, d_r^2, d_r^3, d_r^4\}$, which we refer to as the r^{th} *clause gadget*. Add the edges $\{d_r^1, d_r^4\}$, $\{d_r^2, d_r^4\}$, and $\{d_r^3, d_r^4\}$. Recall that 3 divides m and $m = 4l$ for some integer $l > 1$. Construct a set of $12l$ agents labelled g_1, g_2, \dots, g_{12l} . For any i where $1 \leq i \leq 3l$, we shall refer to $G_i = \{g_{4i-3}, g_{4i-2}, g_{4i-1}, g_{4i}\}$ as the i^{th} *garbage collector gadget*. For each $1 \leq i \leq 3l$, add the edges $\{g_{4i}, g_{4i-1}\}$, $\{g_{4i}, g_{4i-2}\}$, and $\{g_{4i}, g_{4i-3}\}$ to E . Note that each garbage collector gadget G_i is an isolated claw $K_{1,3}$. We shall connect the variable and clause gadgets are connected in the way as the reduction in Section 7.2.2. Consider each clause $c_r = \{x_i, x_j, x_k\}$. If c_r contains the j^{th} occurrence of x_i then add the edge $\{d_r^1, w_i^j\}$. Similarly, add an edge between d_r^2 and an agent in W_j depending on the index of the occurrence of x_j in the clause c_r and an edge between d_r^3 and an agent in W_k depending on the index of the occurrence of x_k in the clause c_r .

This completes the construction of (N, E) . Note that each agent in a variable gadget has degree 3, d_r^1, d_r^2, d_r^3 for each $1 \leq r \leq m$ have degree 2, d_r^4 for each $1 \leq r \leq m$ has degree 3, g_i where $1 \leq i \leq 12l$ and i is not divisible by four has degree 1, and g_j where $1 \leq j \leq 12l$ and j is divisible by four has degree 3. It follows that the maximum degree of (N, E) is 3.

It is straightforward to show that this reduction can be performed in polynomial time. To prove that the reduction is correct we show that the 3DR-AS instance (N, E) contains a wj-envy-free matching if and only if the X3SAT_+^3 instance C is exactly satisfiable.

We first show that if the $X3SAT_+^=3$ instance C is exactly satisfiable then the 3DR-AS instance (N, E) contains a wj -envy-free matching.

Lemma 7.29. *If C is exactly satisfiable then (N, E) contains a wj -envy-free matching.*

Proof. Suppose f is an exact model in C . We shall construct a matching M in (N, E) that is wj -envy-free. For each variable x_i in $X(C)$ where $f(x_i)$ is true, add $\{w_i^1, w_i^2, w_i^3\}$ to M . Next, consider each clause $c_r = \{x_i, x_j, x_k\}$ and the corresponding clause gadget D_r , labelling i, j, k such that W_i contains an agent adjacent to d_r^1 , W_j contains an agent adjacent to d_r^2 , and W_k contains an agent adjacent to d_r^3 . There are three cases: $f(x_i)$ is true while both $f(x_j)$ and $f(x_k)$ are false, $f(x_j)$ is true while both $f(x_i)$ and $f(x_k)$ are false, and $f(x_k)$ is true while both $f(x_i)$ and $f(x_j)$ are false. In the first case, suppose c_r contains the a^{th} occurrence of x_j and the b^{th} occurrence of x_k . Add the triples $\{d_r^1, d_r^4, g_{3r}\}$, $\{d_r^2, w_j^a, g_{3r-1}\}$, and $\{d_r^3, w_k^b, g_{3r-2}\}$. The constructions in the second and third cases are symmetric: in the second case, suppose c_r contains the a^{th} occurrence of x_i and the b^{th} occurrence of x_k . Add the triples $\{d_r^2, d_r^4, g_{3r}\}$, $\{d_r^1, w_i^a, g_{3r-1}\}$, and $\{d_r^3, w_k^b, g_{3r-2}\}$. In the third case, suppose c_r contains the a^{th} occurrence of x_i and the b^{th} occurrence of x_j . Add the triples $\{d_r^3, d_r^4, g_{3r}\}$, $\{d_r^1, w_i^a, g_{3r-1}\}$, and $\{d_r^2, w_j^b, g_{3r-2}\}$.

Note that for any triple $t \in M$, either $t = W_i$ for some variable gadget W_i ; $t = \{d_r^4, d_r^a, g_{3r}\}$ where $1 \leq r \leq m$ and $1 \leq a \leq 3$; or $t = \{d_r^a, w_i^b, g_j\}$ where $1 \leq i, r \leq m$, $1 \leq a, b \leq 3$ and $1 \leq j \leq 12l$ where j is not divisible by three. We shall show that in each case t does not contain an agent with wj -envy.

First, consider some triple $t \in M$ where $t = W_i$ for some variable gadget W_i . Since each agent in t has utility 2, no agent in t is envious.

Second, consider some triple $t \in M$ where $t = \{d_r^4, d_r^a, g_{3r}\}$, $1 \leq r \leq m$, and $1 \leq a \leq 3$. Since $M(d_r^1) \neq M(d_r^2)$, $M(d_r^1) \neq M(d_r^3)$, and $M(d_r^2) \neq M(d_r^3)$, it must be that $\sigma(N(d_r^4), M) = 3$. Since $u_{d_r^4}(M) = 1$, it follows by Lemma 7.2 that d_r^4 is not envious. Similarly, since $M(d_r^a) = \{d_r^4, d_r^a, g_{3r}\}$ it follows that $\sigma(N(d_r^a), M) = 2$ so since $u_{d_r^a}(M) = 1$ by Lemma 7.2 it must be that d_r^a is also not envious. Suppose for a contradiction that g_{3r} wj -envies some agent α_j . It must be that $u_{g_{3r}}(M(\alpha_j) \setminus \{\alpha_j\}) \geq 1$ so $M(\alpha_j)$ contains some g_q where $\{g_{3r}, g_q\} \in E$. Let $M(\alpha_j) = \{\alpha_j, g_q, \alpha_k\}$. By construction of M , it must be that $\alpha_j, \alpha_k \subset D_s$ where $1 \leq s \leq m$ and $u_{\alpha_j}(M) = u_{\alpha_k}(M) = 1$. It follows that $\{\alpha_k, g_q\} \notin E$ so $u_{\alpha_k}(M) < u_{\alpha_k}(\{g_q, g_{3r}\})$, which contradicts the supposition that g_{3r} wj -envies α_j .

Third, consider some triple $t \in M$ where $t = \{d_r^a, w_i^b, g_j\}$, $1 \leq i, r \leq m$, $a \in \{1, 2, 3\}$, $b \in \{1, 2, 3\}$ and $1 \leq j \leq 12l$ where j is not divisible by three. By construction, $\sigma(N(w_i^b), M) = 3$ so by Lemma 7.2 w_i^b is not envious. Similarly, since $M(d_r^a) = \{d_r^a, w_i^b, g_j\}$ it follows that $\sigma(N(d_r^a), M) = 2$ so since $u_{d_r^a}(M) = 1$ by Lemma 7.2 it must be that d_r^a is also not envious. As before, suppose for a contradiction that g_j wj -envies some agent α_k . It must be that

$u_{g_j}(M(\alpha_k) \setminus \{\alpha_k\}) \geq 1$ so $M(\alpha_k)$ contains some g_q where $\{g_j, g_q\} \in E$. Let $M(\alpha_k) = \{\alpha_k, g_q, \alpha_h\}$. By construction of M , it must be that $\alpha_k, \alpha_h \in D_s$ for some s where $1 \leq s \leq m$ and $u_{\alpha_k}(M) = u_{\alpha_h}(M) = 1$. It follows that $\{\alpha_h, g_q\} \notin E$ so $u_{\alpha_h}(M) < u_{\alpha_h}(\{g_q, g_j\})$, which contradicts the supposition that g_j wj-envies α_k . \square

We now show, using a sequence of lemmas, that if the 3DR-AS instance (N, E) contains an wj-envy-free matching then the X3SAT_+^3 instance C is exactly satisfiable.

Lemma 7.30. *If (N, E) contains an envy-free matching M then for any variable gadget W_i , either $\sigma(W_i, M) = 1$ or $\sigma(W_i, M) = 3$.*

Proof. The proof is similar to Lemma 7.5. If some triple in M contains exactly two agents in W_i then the third agent in W_i is wj-envious. \square

Lemma 7.31. *If (N, E) contains an envy-free matching M then $u_{g_i}(M) = 0$ for any i where $1 \leq i \leq 12l$.*

Proof. Suppose M is a wj-envy-free matching. By construction, the structure of G_i for each $1 \leq i \leq 3l$ is identical, so to simplify the proof we assume, without loss of generality, that $i = 1$ and $G_1 = \{g_1, g_2, g_3, g_4\}$. We shall prove that $\sigma(G_1, M) = 4$, from which it follows directly that $u_{g_1}(M) = u_{g_2}(M) = u_{g_3}(M) = u_{g_4}(M) = 0$. Since $|G_1| = 4$ clearly $\sigma(G_1, M) \leq 4$. Suppose for a contradiction that $\sigma(G_1, M) \leq 3$. Then there exists two agents $g_a, g_b \in G_1$ where $g_a \in M(g_b)$. Label the third agent in $M(g_b)$ as α_j . By symmetry, we need only consider two cases. In the first case, $a = 1$ and $b = 4$. In the second case, $a = 1$ and $b = 2$. We will show that in both cases it is relatively straightforward to identify an envious agent, which is a contradiction.

First, suppose $a = 1$ and $b = 4$. Since $\{g_1, g_4\} \subset M(g_4)$, by construction it must be that either $u_{g_2}(M) = 0$ or $u_{g_3}(M) = 0$. Assume without loss of generality that $u_{g_2}(M) = 0$. It follows that g_2 wj-envies g_1 , since $u_{g_4}(\{g_1, \alpha_j\}) = u_{g_4}(\{g_2, \alpha_j\})$ and, by construction of G_1 , $u_{\alpha_j}(\{g_4, g_1\}) = u_{\alpha_j}(\{g_4, g_2\})$.

Second, suppose $a = 1$ and $b = 2$. There are two cases: $g_4 = \alpha_j$ or $g_4 \neq \alpha_j$. If $g_4 = \alpha_j$ then g_3 wj-envies g_2 , since $u_{g_1}(\{g_2, g_4\}) = u_{g_1}(\{g_3, g_4\})$ and $u_{g_4}(\{g_1, g_2\}) = u_{g_4}(\{g_2, g_3\})$. If $g_4 \neq \alpha_j$ then $u_{g_4}(M) \leq 1$ so g_4 wj-envies α_j , since $u_{g_4}(\{g_1, g_2\}) = 2$, $u_{g_1}(M) = u_{g_2}(M) = 0$, and $u_{g_1}(\{g_2, g_4\}) = u_{g_2}(\{g_1, g_4\}) = 1$. \square

Lemma 7.32. *If (N, E) contains an envy-free matching M then $M(g_i) = \{g_i, \alpha_a, \alpha_b\}$ where $\{\alpha_a, \alpha_b\} \in E$ for any $1 \leq i \leq 12l$.*

Proof. Suppose M is a wj-envy-free matching. Consider an arbitrary g_i where $1 \leq i \leq 12l$. Let g_j be some agent for which $\{g_i, g_j\} \in E$. By Lemma 7.31, $u_{g_i}(M) = u_{g_j}(M) = 0$. It

remains to show $\{\alpha_a, \alpha_b\} \in E$. Suppose for a contradiction that $\{\alpha_a, \alpha_b\} \notin E$. It follows that $u_{\alpha_a}(M) = u_{\alpha_b}(M) = u_{g_i}(M) = 0$. Now g_j wj-envies α_a , since $u_{g_j}(\{g_i, \alpha_b\}) = 1$, $u_{\alpha_b}(\{g_j, g_i\}) = u_{\alpha_b}(M)$, and $u_{g_i}(\{g_j, \alpha_b\}) = 1$, which is a contradiction. \square

Lemma 7.33. *Suppose (N, E) contains a wj-envy-free matching M . For each $1 \leq r \leq m$, there exist exactly three triples t_1, t_2, t_3 that each contains one agent in G and at least one agent $d_r^w \in D_r$ where $u_{d_r^w}(M) = 1$.*

Proof. By Lemma 7.32, $M(g_i) = \{g_i, \alpha_a, \alpha_b\}$ where $\{\alpha_a, \alpha_b\} \in E$ for any i where $1 \leq i \leq 12l$. It follows that there exists $12l$ triples in M of the form $\{g_i, \alpha_a, \alpha_b\}$ where $\{\alpha_a, \alpha_b\} \in E$. Let T be this set of triples. Since $\{\alpha_a, \alpha_b\} \in E$, it must be that $u_{\alpha_a}(M) = u_{\alpha_b}(M) = 1$. It follows that $\alpha_a \notin G$ and $\alpha_b \notin G$. It must be that for any $\{g_i, \alpha_a, \alpha_b\} \in T$ there exists some $1 \leq r \leq m$ where either $\alpha_a \in D_r$ or $\alpha_b \in D_r$, for otherwise the only possibility is that $\{\alpha_a, \alpha_b\} \subset W_i$ for some variable gadget W_i , which would contradict Lemma 7.30. We have now shown that each $t \in T$ comprises $\{g_i, d_r^w, \alpha_b\}$ where $u_{d_r^w}(M) = u_{\alpha_b}(M) = 1$ for some $1 \leq r \leq m$, $1 \leq w \leq 4$ and $\alpha_b \in N$. It remains to show that for a given r , there exist exactly three triples in T where each triple contains at least one agent in D_r .

Suppose for a contradiction that there exists some $1 \leq r \leq m$ where the number of triples in T that contain an agent in D_r is not three. Recall that $|T| = 12l$, $m = 4l$, and each $t \in T$ contains at least one agent d_r^w where $1 \leq r \leq m$ and $1 \leq w \leq 4$. A counting argument shows that there must exist some $1 \leq s \leq m$ where there are at least four triples $t_1, t_2, t_3, t_4 \in T$ that each contain at least one agent in D_s . Without loss of generality, it must be that $d_s^1 \in t_1$, $d_s^2 \in t_2$, $d_s^3 \in t_3$, and $d_s^4 \in t_4$. Since we have previously established that $u_{d_s^4}(M) = 1$ this leads to a contradiction since $N(d_s^4) = \{d_s^1, d_s^2, d_s^3\}$ but $d_s^1 \notin t_4$, $d_s^2 \notin t_4$, and $d_s^3 \notin t_4$. It follows that for each $1 \leq r \leq m$, there exists exactly three triples in T where each triple contains at least one agent in D_r . \square

Lemma 7.34. *If (N, E) contains an envy-free matching then C is exactly satisfiable.*

Proof. Suppose M is a wj-envy-free matching in (N, E) . By Lemma 7.30, for any variable gadget W_i either $\sigma(W_i, M) = 1$ or $\sigma(W_i, M) = 3$. Construct a truth assignment f in C by setting $f(x_i)$ to be true if $\sigma(W_i, M) = 1$ and false otherwise. Each variable x_i corresponds to exactly one variable gadget W_i so it follows that f is a valid truth assignment. By the construction of (N, E) , each clause c_r corresponds to exactly one clause gadget D_r . Each clause gadget is adjacent to three variable gadgets that correspond to the three variables in that clause. To show that f is an exact model of C , it is sufficient to show that for each clause gadget D_r there exists exactly one variable gadget W_i such that D_r is adjacent to W_i and $\sigma(W_i, M) = 1$.

Consider an arbitrary clause gadget D_r and the corresponding clause $c_r = \{x_i, x_j, x_k\}$, labelling i, j, k such that d_r^1 is adjacent to some agent $w_i^{a1} \in W_i$, d_r^2 is adjacent to some

agent $w_j^{a_2} \in W_j$ and d_r^3 is adjacent to some agent $w_k^{a_3} \in W_k$. By Lemma 7.33, there exists exactly three triples t_1, t_2, t_3 such that t_1 contains one agent $g_{h_1} \in G$ and at least one agent $d_r^{b_1} \in D_r$ where $u_{d_r^{b_1}}(M) = 1$; t_2 contains one agent $g_{h_2} \in G$ and at least one agent $d_r^{b_2} \in D_r$ where $u_{d_r^{b_2}}(M) = 1$; and t_3 contains one agent in $g_{h_3} \in G$ and at least one agent $d_r^{b_3} \in D_r$ where $u_{d_r^{b_3}}(M) = 1$. Consider d_r^4 . If $u_{d_r^4}(M) = 0$ then $\{b_1, b_2, b_3\} = \{1, 2, 3\}$ so without loss of generality we may assume that $t_1 = \{d_r^1, w_i^{a_1}, g_{h_1}\}$, $t_2 = \{d_r^2, w_j^{a_2}, g_{h_2}\}$, and $t_3 = \{d_r^3, w_k^{a_3}, g_{h_3}\}$. In this configuration d_r^4 wj-envies $w_i^{a_1}$ since $u_{d_r^4}(\{d_r^1, g_{h_1}\}) = 1$, $u_{d_r^1}(\{d_r^4, g_{h_1}\}) = u_{d_r^1}(M)$, and $u_{g_{h_1}}(\{d_r^4, d_r^1\}) = u_{g_{h_1}}(M) = 0$. It follows that $u_{d_r^4}(M) > 0$. Recalling our earlier observation (in this proof) on the contents of t_1, t_2 , and t_3 it must be that $u_{d_r^4}(M) < 2$ and so $u_{d_r^4}(M) = 1$. There are three possible cases: either $d_r^1 \in M(d_r^4)$, $d_r^2 \in M(d_r^4)$, or $d_r^3 \in M(d_r^4)$. By the symmetry of the clause gadget, we describe only the case in which $d_r^1 \in M(d_r^4)$. Without loss of generality we may assume that $t_1 = \{d_r^1, d_r^4, g_{h_1}\}$. Furthermore, we may assume that $t_2 = \{d_r^2, w_j^{a_2}, g_{h_2}\}$ and $t_3 = \{d_r^3, w_k^{a_3}, g_{h_3}\}$. Consider $w_i^{a_1}$. If $u_{w_i^{a_1}}(M) = 0$ then $w_i^{a_1}$ wj-envies d_r^4 , since $u_{w_i^{a_1}}(\{d_r^1, g_{h_1}\}) = 1$, $u_{d_r^1}(\{w_i^{a_1}, g_{h_1}\}) = u_{d_r^1}(M)$, and $u_{g_{h_1}}(\{w_i^{a_1}, d_r^1\}) = u_{g_{h_1}}(M) = 0$. It must be that $u_{w_i^{a_1}}(M) > 0$. If $u_{w_i^{a_1}}(M) = 1$ then Lemma 7.30 is contradicted. It follows that $u_{w_i^{a_1}}(M) = 2$ and thus $\sigma(W_i, M) = 1$. Now consider $w_j^{a_2}$ and $w_k^{a_3}$. Since $u_{w_j^{a_2}}(M) = 1$ it must be that $\sigma(W_j, M) = 3$. Similarly, since $u_{w_k^{a_3}}(M) = 1$ it must be that $\sigma(W_k, M) = 3$. To recap, after we supposed that $d_r^1 \in M(d_r^4)$, we showed that $\sigma(W_i, M) = 1$ and $\sigma(W_j, M) = \sigma(W_k, M) = 3$, as required. The cases in which $d_r^2 \in M(d_r^4)$ or $d_r^3 \in M(d_r^4)$ are symmetric. \square

We have now shown that the 3DR-AS instance (N, E) contains a wj-envy-free matching if and only if the X3SAT_+^3 instance C is exactly satisfiable. This shows that the reduction is correct.

Theorem 7.4. *Deciding if a given instance of 3DR-AS contains a wj-envy-free matching is NP-complete, even when preferences are binary and symmetric and maximum degree is 3.*

Proof. It is straightforward to show that this decision problem belongs to NP, since for any two agents $\alpha_i, \alpha_j \in N$ we can test if α_i wj-envies α_j in constant time.

We have presented a polynomial-time reduction from X3SAT_+^3 , which is NP-complete [88]. Given an arbitrary instance C of X3SAT_+^3 , the reduction constructs an instance (N, E) of 3DR-AS with binary and symmetric preferences and maximum degree 3. Lemmas 7.29 and 7.34 show that (N, E) contains a wj-envy-free matching if and only if C is exactly satisfiable and thus that this decision problem is NP-hard. \square

7.4 Justified envy-freeness

7.4.1 Binary preferences

It is straightforward to show that if a matching is stable then it is j -envy-free. A corollary is that if we are given an instance of 3DR-AS in which preferences are binary and symmetric then we can find a j -envy-free matching by finding a stable matching, which, as we showed in Chapter 6, is bound to exist and can be found in polynomial time (Theorem 6.1 in Chapter 6). We state this corollary as Observation 7.1.

Observation 7.1. *Given an instance of 3DR-AS with binary and symmetric preferences, a j -envy-free matching always exists and can be found in $O(|N|^3)$ time.*

We showed in Chapter 6 that when preferences are binary but not necessarily symmetric, a stable matching need not exist and the associated decision problem is NP-complete. In contrast, we show that a j -envy-free matching is bound to exist and can be found in polynomial time.

Theorem 7.5. *Given an instance of 3DR-AS with binary preferences, a j -envy-free matching must exist and can be found in $O(|N|^3)$ time.*

Proof. Suppose (N, V) is an instance of 3DR-AS with binary preferences. We describe an $O(|N|^3)$ -time algorithm that can construct a j -envy-free matching M in (N, V) , as follows.

First, the algorithm constructs from (N, V) another instance (N, V') of 3DR-AS, which has binary and symmetric preferences, by deleting asymmetric arcs. Next, the algorithm calls Algorithm findStableUW, shown in Algorithm 6.4, in Chapter 6, on the instance (N, V') to construct a matching M , which is returned.

Since (N, V') can be constructed in $O(|N|^2)$ time, and the worst-case time complexity of Algorithm findStableUW is $O(|N|^3)$ (by Theorem 6.3 in Chapter 6), the worst-case time complexity of this algorithm is also $O(|N|^3)$.

It remains to show that M is j -envy-free in (N, V) . In the proof, which follows, we refer to the variables and subroutines in Algorithm findStableUW. We shall also write v in the context of the instance (N, V') , which has binary preferences, and write v' in the context of the instance (N, V) , which has binary and symmetric preferences. Similarly, we shall use $u'_{\alpha_i}(M)$ to denote $u'_{\alpha_i}(S) = \sum_{\alpha_j \in S \setminus \{\alpha_i\}} v'_{\alpha_i}(\alpha_j)$ for some set of agents $S \subseteq N$.

Note that by the design of Algorithm findStableUW, M meets the following condition: if there exists a pair of agents $\alpha_i, \alpha_j \in N$ where $v'_{\alpha_i}(\alpha_j) = 1$ and $u'_{\alpha_i}(M) = u'_{\alpha_j}(M) = 0$ then no triple contains three agents each with utility 0 in M .

For a contradiction, suppose there exists some $\alpha_{i_1} \in N$ where α_{i_1} has j-envy in M for some $\alpha_{j_1} \in N \setminus \{\alpha_{i_1}\}$.

Let $M(\alpha_{i_1}) = \{\alpha_{i_1}, \alpha_{i_2}, \alpha_{i_3}\}$ and $M(\alpha_{j_1}) = \{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$. Since α_{i_1} has j-envy for α_{j_1} it must be that $u_{\alpha_{i_1}}(\{\alpha_{j_2}, \alpha_{j_3}\}) > 0$ so either $v_{\alpha_{i_1}}(\alpha_{j_2}) = 1$, $v_{\alpha_{i_1}}(\alpha_{j_3}) = 1$, or both. Assume without loss of generality that $v_{\alpha_{i_1}}(\alpha_{j_2}) = 1$. Note that it must also be that $u_{\alpha_{j_2}}(\{\alpha_{i_1}, \alpha_{j_3}\}) > u_{\alpha_{j_2}}(M)$ and $u_{\alpha_{j_3}}(\{\alpha_{i_1}, \alpha_{j_2}\}) > u_{\alpha_{j_3}}(M)$. It follows that $v_{\alpha_{j_2}}(\alpha_{i_1}) > v_{\alpha_{j_2}}(\alpha_{j_1})$ and $v_{\alpha_{j_3}}(\alpha_{i_1}) > v_{\alpha_{j_3}}(\alpha_{j_1})$. Since preferences are binary, it must be that $v_{\alpha_{j_2}}(\alpha_{i_1}) = v_{\alpha_{j_3}}(\alpha_{i_1}) = 1$ and $v_{\alpha_{j_2}}(\alpha_{j_1}) = v_{\alpha_{j_3}}(\alpha_{j_1}) = 0$. It follows by the construction of (N, V') that $v'_{\alpha_{j_2}}(\alpha_{j_1}) = v'_{\alpha_{j_3}}(\alpha_{j_1}) = 0$. Since $v_{\alpha_{i_1}}(\alpha_{j_2}) = 1$ and $v_{\alpha_{j_2}}(\alpha_{i_1}) = 1$ it also follows that $v'_{\alpha_{i_1}}(\alpha_{j_2}) = 1$.

We first claim that $\{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\} \notin M_1$. Since $v_{\alpha_{j_2}}(\alpha_{j_1}) = v_{\alpha_{j_3}}(\alpha_{j_1}) = 0$, it follows that $v'_{\alpha_{j_2}}(\alpha_{j_1}) = v'_{\alpha_{j_3}}(\alpha_{j_1}) = 0$ and thus that $u'_{\alpha_{j_1}}(M_1) = 0$. Since M_1 is a P -matching, no triple in M_1 contains an agent with utility 0, which shows that $\{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\} \notin M_1$ as required.

Note that it follows that $\{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\} \subseteq U$ and thus that $u'_{\alpha_{j_2}}(M_1) = u'_{\alpha_{j_3}}(M_1) = 0$.

Next, we claim that $u'_{\alpha_{i_1}}(M) = 0$. Since α_{i_1} envies α_{j_1} in (N, V) it must be that $u_{\alpha_{i_1}}(M) < 2$ and thus that $u'_{\alpha_{i_1}}(M) < 2$. It remains that $u'_{\alpha_{i_1}}(M) \in \{0, 1\}$. Suppose for a contradiction that $u'_{\alpha_{i_1}}(M) = 1$ and without loss of generality that $v'_{\alpha_{i_1}}(\alpha_{i_2}) = 1$. It follows that $v_{\alpha_{i_1}}(\alpha_{i_2}) = 1$ and thus that $u_{\alpha_{i_1}}(M) \geq 1$. In fact, since α_{i_1} has j-envy for α_{j_1} it must be that $u_{\alpha_{i_1}}(M) = 1$. Moreover, since $u_{\alpha_{i_1}}(\{\alpha_{j_2}, \alpha_{j_3}\}) > u_{\alpha_{i_1}}(M) = 1$, it must be that $v_{\alpha_{i_1}}(\alpha_{j_2}) = v_{\alpha_{i_1}}(\alpha_{j_3}) = 1$. Recall that since α_{i_1} has justified envy for α_{j_1} , it must be that $v_{\alpha_{j_2}}(\alpha_{j_1}) = v_{\alpha_{j_3}}(\alpha_{j_1}) = 0$ and $v_{\alpha_{j_2}}(\alpha_{i_1}) = v_{\alpha_{j_3}}(\alpha_{i_1}) = 1$. We have now shown that $v'_{\alpha_{j_2}}(\alpha_{i_1}) = v'_{\alpha_{j_3}}(\alpha_{i_1}) = 1$. Recall our earlier note that $u'_{\alpha_{j_2}}(M_1) = u'_{\alpha_{j_3}}(M_1) = 0$. Now, the triple $\{\alpha_{i_1}, \alpha_{j_2}, \alpha_{j_3}\}$ blocks M_1 in (N, V') , since $u'_{\alpha_{i_1}}(M_1) = 1 < 2 = u'_{\alpha_{i_1}}(\{\alpha_{j_2}, \alpha_{j_3}\})$, $u'_{\alpha_{j_2}}(M_1) = 0 < 1 \leq u'_{\alpha_{j_2}}(\{\alpha_{i_1}, \alpha_{j_3}\})$, and $u'_{\alpha_{j_3}}(M_1) = 0 < 1 \leq u'_{\alpha_{j_3}}(\{\alpha_{i_1}, \alpha_{j_2}\})$. Since M_1 is a stable matching in (N, V') , by the correctness of Algorithm findStableUW (shown in Theorem 6.3 in Chapter 6) this is a contradiction. It remains that $u'_{\alpha_{i_1}}(M) = 0$, as required.

Note that since M_1 is a P -matching and $u'_{\alpha_{i_1}}(M) = 0$ it must be that α_{i_1} is unmatched in M_1 so $\alpha_{i_1} \in U$.

We finally claim that $v'_{\alpha_{j_2}}(\alpha_{j_3}) = 0$. Suppose for a contradiction that $v'_{\alpha_{j_2}}(\alpha_{j_3}) = 1$. We established earlier that $\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}$ are unmatched in M_1 so $u'_{\alpha_{j_2}}(M_1) = u'_{\alpha_{j_3}}(M_1) = 0$. We also established that $u'_{\alpha_{i_1}}(M) = 0$ and thus that $u'_{\alpha_{i_1}}(M_1) = 0$. Recall also that, by assumption, $v'_{\alpha_{i_1}}(\alpha_{j_2}) = 1$, so since $v'_{\alpha_{j_2}}(\alpha_{j_3}) = 1$ it now follows that $\{\alpha_{i_1}, \alpha_{j_2}, \alpha_{j_3}\}$ blocks M_1 in (N, V') , which is a contradiction since M_1 is stable in (N, V') . It remains that $v'_{\alpha_{j_2}}(\alpha_{j_3}) = 0$, as required.

Since $v'_{\alpha_{j_1}}(\alpha_{j_3}) = 0$, $v'_{\alpha_{j_2}}(\alpha_{j_1}) = 0$, and $v'_{\alpha_{j_2}}(\alpha_{j_3}) = 0$ it follows that $u'_{\alpha_{j_1}}(M) = u'_{\alpha_{j_2}}(M) = u'_{\alpha_{j_3}}(M) = 0$. This violates our stated condition on M , since there exists a pair of agents $\alpha_{i_1}, \alpha_{j_2} \in N$ where $v'_{\alpha_{i_1}}(\alpha_{j_2})$ and $u'_{\alpha_{i_1}}(M) = u'_{\alpha_{j_2}}(M) = 0$, but each agent in the triple $\{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$ has utility 0 in M , which is a contradiction. \square

7.4.2 Ternary preferences

A natural next step would be to ask if the polynomial-time algorithm described in Theorem 7.5 can be extended to the setting in which preferences are ternary, i.e. $v_{\alpha_i}(\alpha_j) \in \{0, 1, 2\}$. We show that, assuming $P \neq NP$, this is not the case, and the problem of deciding if a given instance of 3DR-AS contains a j -envy-free matching is NP-complete, even when preferences are ternary.

We present a polynomial-time reduction from a special case of *Directed Triangle Packing* (DTP, Problem 7.2).

Problem 7.2. Directed Triangle Packing (DTP)

Input: a simple directed graph $G = (W, A)$ where $W = \{w_1, w_2, \dots, w_{3q}\}$ for some integer q

Question: Can the vertices of G be partitioned into q disjoint sets $X = \{X_1, X_2, \dots, X_q\}$, each set containing exactly three vertices, such that each $X_p = \{w_i, w_j, w_k\}$ in X is a directed 3-cycle, i.e. the arcs (w_i, w_j) , (w_j, w_k) , and (w_k, w_i) belong to A ?

We claim that DTP is NP-complete, even when G is antisymmetric (i.e. it contains no bidirectional arcs). As noted by Cechlárová, Fleiner, and Manlove [89], the proof of this claim can be obtained using a simple modification to the reduction presented by Garey and Johnson for Partition Into Triangles [43, Theorem 3.7]. It is this restricted variant of DTP, in which G is antisymmetric, that we reduce from to show that deciding if a given instance of 3DR-AS with ternary (but not necessarily symmetric) preferences contains a j -envy-free matching.

We shall first describe the reduction from DTP in detail and then provide some intuition with respect to its design.

The reduction, shown in Figure 7.3, is as follows. Suppose $G = (W, A)$ is an arbitrary instance of DTP. We shall construct an instance (N, V) of 3DR-AS. Unless otherwise specified, assume that $v_{\alpha_i}(\alpha_j) = 0$ for any $\alpha_i, \alpha_j \in N$. To simplify the description of the valuations in the reduction, in this section we write $i \oplus y$ to denote $((i + y - 1) \bmod 5) + 1$.

First construct a set of five agents $H = \{h_1, h_2, h_3, h_4, h_5\}$. For each i where $1 \leq i \leq 5$ let $v_{h_i}(h_{i \oplus 1}) = v_{h_i}(h_i) = 1$, $v_{h_i}(h_{i \oplus 3}) = 1$, and $v_{h_i}(h_{i \oplus 2}) = 2$. Next, construct a set $L = \{l_1, l_2, l_3, l_4\}$ of four agents. Let $v_{l_1}(l_2) = v_{l_2}(l_1) = v_{l_3}(l_4) = v_{l_4}(l_3) = 2$ and $v_{l_1}(l_3) = v_{l_1}(l_4) = v_{l_2}(l_3) = v_{l_2}(l_4) = v_{l_3}(l_1) = v_{l_3}(l_2) = v_{l_4}(l_1) = v_{l_4}(l_2) = 1$. Next, construct a set $C = \{c_1, c_2, \dots, c_{3q}\}$ of $3q$ agents. For each i where $1 \leq i \leq 3q$ let $v_{c_i}(l_3) = v_{l_3}(c_i) = v_{l_4}(c_i) = 1$ and $v_{c_i}(l_4) = 2$. For each i and j where $1 \leq i, j \leq 3q$ let $v_{c_i}(c_j) = 2$ if $(w_i, w_j) \in A$ otherwise 1. This completes the construction of (N, V) . Note that the structure of the valuations among the agents in C reflects the directed graph G .

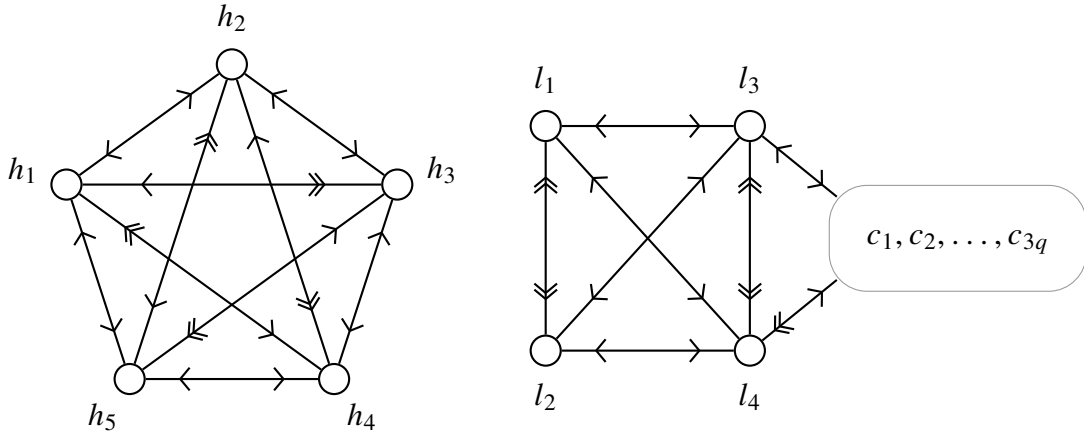


Figure 7.3: The reduction from DTP to the problem of deciding if a given instance of 3DR-AS with ternary preferences contains a j -envy-free matching

We make some remarks on the design of the constructed instance. The design of H is derived from a particular instance that contains no j -envy-free matching. This instance comprises H as well as a single isolated agent α_z , where $v_{\alpha_z}(h_i) = v_{h_i}(\alpha_z) = 0$ for each i where $1 \leq i \leq 5$. In fact, the proof that this instance contains no j -envy-free matching can be directly derived from the proof of a lemma that appears later in this section (Lemma 7.37).

It is straightforward to show that the reduction runs in polynomial time. To prove that the reduction is correct we show that the 3DR-AS instance (N, V) contains a j -envy-free matching if and only if the DTP instance G contains a directed triangle packing.

We first show that if the DTP instance G contains a directed triangle packing then the 3DR-AS instance (N, V) contains a j -envy-free matching.

Lemma 7.35. *If G contains a directed triangle packing then (N, V) contains a j -envy-free matching.*

Proof. Suppose G contains a directed triangle packing $X = \{X_1, X_2, \dots, X_q\}$. We shall construct a matching M that is j -envy-free. First, add $\{h_1, h_2, h_3\}$, $\{h_4, l_1, l_2\}$, and $\{h_5, l_3, l_4\}$ to M . Next, for each directed 3-cycle $X_p = \{w_i, w_j, w_k\}$ in X , add $\{c_i, c_j, c_k\}$ to M .

Suppose for a contradiction that some agent α_j exists where α_j has j -envy for some other agent α_{k_1} where $M(\alpha_{k_1}) = \{\alpha_{k_1}, \alpha_{k_2}, \alpha_{k_3}\}$. Since $N = H \cup L \cup C$ it must be that either $\alpha_{k_1} \in H$, $\alpha_{k_1} \in L$, or $\alpha_{k_1} \in C$. We show that each case leads to a contradiction. It follows that no such α_j exists and thus that M is j -envy-free.

- Suppose $\alpha_{k_1} \in H$. By the construction of M there are two possibilities: either $\alpha_{k_1} \in \{h_1, h_2, h_3\}$ or $\alpha_{k_1} \in \{h_4, h_5\}$.

- Suppose firstly that $\alpha_{k_1} \in \{h_4, h_5\}$ then by the construction of M either $\{\alpha_{k_2}, \alpha_{k_3}\} = \{l_1, l_2\}$ or $\{\alpha_{k_2}, \alpha_{k_3}\} = \{l_3, l_4\}$. Note that $u_{l_1}(M) = u_{l_2}(M) = u_{l_3}(M) = u_{l_4}(M) = 2$. Since $u_{l_1}(\{l_3, l_4\}) = 2$ and $u_{l_2}(\{l_3, l_4\}) = 2$, neither l_1 nor l_2 has j-envy for α_{k_1} , so $\alpha_j \notin \{l_1, l_2\}$. Similarly, since $u_{l_3}(\{l_1, l_2\}) = 2$ and $u_{l_4}(\{l_1, l_2\}) = 2$ neither l_3 nor l_4 has j-envy for α_{k_1} , so $\alpha_j \notin \{l_3, l_4\}$. Since $u_{c_i}(M) = 3$, $u_{c_i}(\{l_1, l_2\}) = 0$, and $u_{c_i}(\{l_1, l_2\}) = 2$ for any i where $1 \leq i \leq 3q$, it must be no agent in C has j-envy for α_{k_1} , so $\alpha_j \notin C$. It remains that $\alpha_j \in H$. Since this implies $v_{l_1}(\alpha_j) = v_{l_2}(\alpha_j) = v_{l_3}(\alpha_j) = v_{l_4}(\alpha_j) = 0$ it follows that α_j does not have j-envy for α_{k_1} and thus that $\alpha_{k_1} \notin \{h_4, h_5\}$.
- Suppose then that $\alpha_{k_1} \in \{h_1, h_2, h_3\}$. Since α_j has j-envy for α_{k_1} it must be that $v_{\alpha_j}(\alpha_{k_2}) \geq 1$ so it follows that $\alpha_j \in \{h_4, h_5\}$. If $\alpha_{k_1} = h_1$ and $\alpha_j = h_4$ then we reach a contradiction since h_4 has j-envy for h_1 but $v_{h_3}(h_1) = 1 = v_{h_3}(h_4)$. Similarly, if $\alpha_{k_1} = h_1$ and $\alpha_j = h_5$ then we reach a contradiction since $v_{h_2}(h_1) = 1 = v_{h_2}(h_5)$. If $\alpha_{k_1} = h_2$ or $\alpha_{k_1} = h_3$ then we also reach a contradiction since $v_{h_1}(h_4) = v_{h_1}(h_5) = 1 = v_{h_1}(h_2) < 2 = v_{h_1}(h_3)$.
- Suppose $\alpha_{k_1} \in C$. By the construction of M it must be that $\alpha_{k_2} \in C$ and $\alpha_{k_3} \in C$ so we label $\alpha_{k_1} = c_{i_1}$, $\alpha_{k_2} = c_{i_2}$, and $\alpha_{k_3} = c_{i_3}$. By the construction of (N, V) in the reduction it follows that $v_{c_{i_2}}(c_{i_1}) \geq 1$ and $v_{c_{i_3}}(c_{i_1}) \geq 1$. Since α_j has j-envy for c_{i_1} it follows then that $v_{c_{i_2}}(\alpha_j) = 2$ and $v_{c_{i_3}}(\alpha_j) = 2$. By the construction of the instance there are two possibilities: either $\alpha_j = l_4$ or $\alpha_j \in C$. If $\alpha_j = l_4$ then $u_{l_4}(\{c_{i_2}, c_{i_3}\}) = 2$ which is a contradiction since by assumption l_4 has j-envy for c_{i_1} but $u_{l_4}(M) = 2$. If $\alpha_j \in C$ then label $\alpha_j = c_{i_4}$. Since $v_{c_{i_2}}(c_{i_4}) = 2$ and $v_{c_{i_3}}(c_{i_4}) = 2$, by the construction of C it must be that $(w_{i_2}, w_{i_4}) \in A$ and $(w_{i_3}, w_{i_4}) \in A$, where the vertices $w_{i_2}, w_{i_3}, w_{i_4}$ are the vertices in W that correspond respectively to the agents $c_{i_2}, c_{i_3}, c_{i_4}$ in C . Since G is antisymmetric, it follows that $(w_{i_4}, w_{i_2}) \notin A$ and $(w_{i_4}, w_{i_3}) \notin A$ so it must be that $v_{c_{i_4}}(c_{i_2}) = v_{c_{i_4}}(c_{i_3}) = 1$. This is also a contradiction since by assumption c_{i_4} has j-envy for c_{i_1} but $u_{c_{i_4}}(\{c_{i_2}, c_{i_3}\}) = 2$ and by the construction of M it must be that $u_{c_{i_4}}(M) = 3$.
- Suppose $\alpha_{k_1} \in L$. It must be that $\alpha_{k_1} = l_{i_1}$, $\alpha_{k_2} = l_{i_2}$, and $\alpha_{k_3} = h_{i_3}$, for some i_1, i_2 where $1 \leq i_1, i_2 \leq 4$ and $i_3 \in \{4, 5\}$. If $\alpha_j \in H$ then it must be that $v_{l_{i_2}}(\alpha_j) = 0$ which contradicts the supposition that α_j has j-envy for l_{i_1} . Otherwise, if $\alpha_j \notin H$ then $v_{h_{i_3}}(\alpha_j) = 0$, which also contradicts the supposition that α_j has j-envy for l_{i_1} .

□

We now show that if the 3DR-AS instance (N, V) contains a j-envy-free matching then the DTP instance G contains a directed triangle packing.

Lemma 7.36. *If (N, V) contains a j-envy-free matching M then $\sigma(H, M) \geq 3$.*

Proof. Since $|H| = 5$ it must be that $\sigma(H, M) \geq 2$. Suppose for a contradiction that $\sigma(H, M) = 2$. It must be that one triple in M contains three agents in H and one triple in M contains two agents in H . Suppose the former triple is $\{h_{i_1}, h_{i_2}, h_{i_3}\}$ and the latter triple is $\{h_{i_4}, h_{i_5}, \alpha_j\}$, where $1 \leq i_1, i_2, \dots, i_5 \leq 5$ and $\alpha_j \in N \setminus H$. There are five symmetries in H and $\binom{5}{2} = 10$ possible assignments of $\{h_{i_4}, h_{i_5}\}$ to two agents in H , so we need only consider the two assignments $i_4 = 1, i_5 = 2$ and $i_4 = 1, i_5 = 3$, which are not symmetric. If $i_4 = 1$ and $i_5 = 2$ then it remains that $\{i_1, i_2, i_3\} = \{3, 4, 5\}$. In this case, h_5 has j-envy for α_j since $u_{h_5}(M) = 2 < 3 \leq u_{h_5}(\{h_1, h_2\})$, $v_{h_1}(\alpha_j) = 0 < 1 = v_{h_1}(h_5)$, and $v_{h_2}(\alpha_j) = 0 < 1 = v_{h_2}(h_5)$. If $i_4 = 1$ and $i_5 = 3$ then it remains that $\{i_1, i_2, i_3\} = \{2, 4, 5\}$. In this case, h_4 has j-envy for α_j since $u_{h_4}(M) = 2 < 3 \leq u_{h_4}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 1 = v_{h_1}(h_4)$, and $v_{h_3}(\alpha_j) = 0 < 1 = v_{h_3}(h_4)$. \square

Lemma 7.37. *If (N, V) contains a j-envy-free matching M then at least two triples in M each contain exactly one agent in H .*

Proof. By Lemma 7.36, $\sigma(H, M) \geq 3$. If, contrary to the lemma statement, at most one triple in M contains exactly one agent in H then it must be that two triples in M each contain two agents in H and one triple in M contains exactly one agent in H . Suppose one of the two former triples is $\{h_{i_1}, h_{i_2}, \alpha_{j_1}\}$ and the latter triple is $\{h_{i_3}, \alpha_{j_2}, \alpha_{j_3}\}$, where $1 \leq i_1, i_2, i_3 \leq 5$ and $\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3} \in N \setminus H$. By the construction of the instance it must be that $v_{h_{i_1}}(\alpha_{j_1}) = v_{h_{i_2}}(\alpha_{j_1}) = 0$, $v_{h_{i_1}}(h_{i_3}) \geq 1$ and $v_{h_{i_2}}(h_{i_3}) \geq 1$. It follows that h_{i_3} has j-envy for α_{j_1} since $u_{h_{i_3}}(M) = 0 < 2 \leq u_{h_{i_3}}(\{h_{i_1}, h_{i_2}\})$, $v_{h_{i_1}}(\alpha_j) = 0 < 1 \leq v_{h_{i_1}}(h_{i_3})$, and $v_{h_{i_2}}(\alpha_j) = 0 < 1 \leq v_{h_{i_2}}(h_{i_3})$. This contradicts the supposition that M is j-envy-free. \square

We have shown in Lemma 7.37 that if (N, V) contains a j-envy-free matching M then at least two triples in M each contain exactly one agent in H . Suppose $t_\beta, t_\gamma \in M$ are two such triples and $t_\beta = \{h_{a_1}, \alpha_{b_1}, \alpha_{b_2}\}$ and $t_\gamma = \{h_{a_2}, \alpha_{b_3}, \alpha_{b_4}\}$.

Lemma 7.38. *If (N, V) contains a j-envy-free matching then $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} = L$.*

Proof. Suppose for a contradiction that $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} \neq L$.

By definition, $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} \cap H = \emptyset$ and $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} \neq L$ it must be that at least one agent in $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\}$ belongs to C . Assume without loss of generality that $\alpha_{b_1} \in C$.

Note that by construction of the instance, the valuation of any agent not in H for any other agent not in H is at least 1.

Since $\alpha_{b_2} \notin H$, it must be that $u_{\alpha_{b_2}}(M) = v_{\alpha_{b_2}}(\alpha_{b_1})$. By the design of the instance, since $\alpha_{b_2} \notin H$ and $\alpha_{b_1} \notin H$ it must be that $v_{\alpha_{b_2}}(\alpha_{b_1}) \in \{1, 2\}$. We consider each possibility of $u_{\alpha_{b_2}}(M) = v_{\alpha_{b_2}}(\alpha_{b_1})$.

Firstly, suppose $u_{\alpha_{b_2}}(M) = 1$. As noted earlier in this proof, since $\alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4} \in N \setminus H$ it must be that $v_{\alpha_{b_2}}(\alpha_{b_3}) \geq 1$ and $v_{\alpha_{b_2}}(\alpha_{b_4}) \geq 1$. It follows that α_{b_2} has j-envy for h_{a_2} , since $u_{\alpha_{b_2}}(M) = 1 < 2 \leq u_{\alpha_{b_2}}(\{\alpha_{b_3}, \alpha_{b_4}\})$, $v_{\alpha_{b_3}}(h_{a_2}) = 0 < 1 \leq v_{\alpha_{b_3}}(\alpha_{b_2})$, and $v_{\alpha_{b_4}}(h_{a_2}) = 0 < 1 \leq v_{\alpha_{b_4}}(\alpha_{b_2})$. This contradicts the supposition that M is j-envy-free.

Suppose then that $u_{\alpha_{b_2}}(M) = 2$, so $v_{\alpha_{b_2}}(\alpha_{b_1}) = 2$. Since $\alpha_{b_1} \in C$ by assumption, by the design of the instance it must be $\alpha_{b_2} \in C$. For the remainder of this lemma only, label $\alpha_{b_1} = c_{i_1}$ and $\alpha_{b_2} = c_{i_2}$. Since $v_{c_{i_2}}(c_{i_1}) = 2$ it follows that $(w_{i_2}, w_{i_1}) \in A$. Since G is antisymmetric it must be that $(w_{i_1}, w_{i_2}) \notin A$ and thus that $v_{c_{i_1}}(c_{i_2}) = 1$. Since $M(c_{i_1}) = \{c_{i_1}, c_{i_2}, h_{a_1}\}$ it follows that $u_{c_{i_1}}(M) = v_{c_{i_1}}(c_{i_2}) = 1$. Now α_{b_1} has j-envy for h_{a_2} , since $u_{\alpha_{b_1}}(M) = 1 < 2 \leq u_{\alpha_{b_1}}(\{\alpha_{b_3}, \alpha_{b_4}\})$, $v_{\alpha_{b_3}}(h_{a_2}) = 0 < 1 \leq v_{\alpha_{b_3}}(\alpha_{b_1})$, and $v_{\alpha_{b_4}}(h_{a_2}) = 0 < 1 \leq v_{\alpha_{b_4}}(\alpha_{b_1})$. This contradicts the supposition that M is j-envy-free. \square

Lemma 7.39. *If (N, V) contains a j-envy-free matching then $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_2\}, \{l_3, l_4\}\}$.*

Proof. By Lemma 7.38, $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} = L$. There are now three possibilities: first that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_3\}, \{l_2, l_4\}\}$, second that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_4\}, \{l_2, l_3\}\}$, and third that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_2\}, \{l_3, l_4\}\}$.

First suppose $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_3\}, \{l_2, l_4\}\}$. Without loss of generality assume that $\alpha_{b_1} = l_1$. Now l_1 has j-envy for h_{a_2} since $u_{l_1}(\{h_{a_1}, l_3\}) = 1 < 3 = u_{l_1}(\{l_2, l_4\})$, $v_{l_2}(h_{a_2}) = 0 < 2 = v_{l_2}(l_1)$, and $v_{l_4}(h_{a_2}) = 0 < 1 = v_{l_4}(l_1)$.

Second suppose $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_4\}, \{l_2, l_3\}\}$. Without loss of generality assume that $\alpha_{b_1} = l_1$. As before, l_1 has j-envy for h_{a_2} since $u_{l_1}(\{h_{a_1}, l_4\}) = 1 < 3 = u_{l_1}(\{l_2, l_3\})$, $v_{l_2}(h_{a_2}) = 0 < 2 = v_{l_2}(l_1)$, and $v_{l_3}(h_{a_2}) = 0 < 1 = v_{l_3}(l_1)$.

It remains that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_2\}, \{l_3, l_4\}\}$. \square

By Lemma 7.39, either $\{\alpha_{b_1}, \alpha_{b_2}\} = \{l_1, l_2\}$ or $\{\alpha_{b_1}, \alpha_{b_2}\} = \{l_3, l_4\}$. Without loss of generality assume that $\{\alpha_{b_1}, \alpha_{b_2}\} = \{l_3, l_4\}$.

Lemma 7.40. *If (N, V) contains a j-envy-free matching M then $u_{c_i}(M) \geq 3$ for each i where $1 \leq i \leq 3q$.*

Proof. Suppose to the contrary that some $1 \leq i \leq 3q$ exists where $u_{c_i}(M) < 3$. Then c_i has j-envy for h_{a_1} since $u_{c_i}(M) \leq 2 < 3 = u_{c_i}(\{l_3, l_4\})$, $v_{l_3}(h_{a_1}) = 0 < 1 = v_{l_3}(c_i)$, and $v_{l_4}(h_{a_1}) = 0 < 1 = v_{l_4}(c_i)$. This contradicts our supposition that M is j-envy-free. \square

Lemma 7.41. *If (N, V) contains a j-envy-free matching M then G contains a directed triangle packing.*

Proof. Suppose (N, V) contains a j -envy-free matching M . Lemma 7.40 shows that $u_{c_i}(M) \geq 3$ for each i where $1 \leq i \leq 3q$. By construction, it follows that $M(c_i)$ contains two agents c_j, c_k such that $v_{c_i}(c_j) \geq 1$ and $v_{c_i}(c_k) = 2$. Hence c_k corresponds to a vertex $w_k \in W$ where $(w_i, w_k) \in A$ and, since G is antisymmetric, $(w_k, w_i) \notin A$. Since c_i was chosen arbitrarily it follows that $\{w_i, w_j, w_k\}$ is a directed 3-cycle in G . It follows thus that there are exactly q triples in M each containing three agents $\{c_i, c_j, c_k\}$ where the three corresponding vertices $\{w_i, w_j, w_k\}$ form a directed 3-cycle in G . From these triples a directed triangle packing X can be easily constructed. \square

We have now shown that the 3DR-AS instance (N, V) contains a j -envy-free matching if and only if the DTP instance G contains a directed triangle packing. This shows that the reduction is correct.

Theorem 7.6. *Deciding if a given instance of 3DR-AS contains a j -envy-free matching is NP-complete, even when preferences are ternary.*

Proof. It is straightforward to show that this decision problem belongs to NP, since for any two agents $\alpha_i, \alpha_j \in N$ we can test if α_i j -envies α_j in constant time.

We have presented a polynomial-time reduction from a restricted version of Directed Triangle Packing (DTP), which is NP-complete [89]. Given a directed antisymmetric graph G , the reduction constructs an instance (N, V) of 3DR-AS with ternary preferences. Lemmas 7.35 and 7.41 show that (N, V) contains a j -envy-free matching if and only if G contains a directed triangle packing and thus that this decision problem is NP-hard. \square

7.4.3 Symmetric non-binary preferences

From Theorems 7.5 and 7.6, a natural question arises: is it the symmetry of agents' preferences that guarantees the existence of a j -envy-free matching? In this section we show that this is not the case, and a j -envy-free matching may not exist even when agents' preferences are symmetric, and the associated existence problem is NP-complete. We remark, however, that the instances shown that do not contain j -envy-free matchings are relatively contrived and involve non-negative integer valuations up to 6. We leave open the case in which preferences are symmetric and the maximum valuation is strictly less than 6.

To show that this existence problem is NP-complete, we present a polynomial-time reduction from *Partition into Triangles* (PIT, Problem 5.1), which is NP-complete [43]. This reduction is similar to the reduction we presented in Section 7.4.2 for the analogous problem involving ternary preferences that are not (necessarily) symmetric. Note that section we reduced from Directed Triangle Packing (DTP, Problem 7.2) but here we reduce from PIT.

We describe the reduction in detail and then provide some intuition with respect to its design. The reduction, illustrated in Figure 7.4, is as follows. Suppose G is an arbitrary instance of PIT. We shall construct an instance (N, V) of 3DR-AS with symmetric preferences and maximum valuation 6. Since valuations are symmetric in (N, V) , we shall usually specify valuations in one direction only. For example, instead of writing “let $v_{\alpha_i}(\alpha_j) = v_{\alpha_j}(\alpha_i) = 1$ ” we write “let $v_{\alpha_i}(\alpha_j) = 1$ ”. Unless otherwise specified assume that $v_{\alpha_i}(\alpha_j) = 0$ for any $\alpha_i, \alpha_j \in N$. To simplify the description of the valuations in the reduction, in this section we write $i \oplus y$ to denote $((i + y - 2) \bmod 10) + 2$.

First, construct a set of eleven agents $H = \{h_1, h_2, \dots, h_{11}\}$. For each i where $2 \leq i \leq 11$ let $v_{h_1}(h_i) = 2$. For each i where $2 \leq i \leq 11$, let:

- $v_{h_i}(h_{i \oplus 1}) = 4$ if i is even otherwise 5
- $v_{h_i}(h_{i \oplus 2}) = 6$ if i is even otherwise 3
- $v_{h_i}(h_{i \oplus 3}) = 1$
- $v_{h_i}(h_{i \oplus 4}) = 1$
- $v_{h_i}(h_{i \oplus 5}) = 3$.

Next, construct a set of four agents $L = \{l_1, l_2, l_3, l_4\}$. Let $v_{l_1}(l_2) = v_{l_3}(l_4) = 2$ and $v_{l_1}(l_3) = v_{l_1}(l_4) = v_{l_2}(l_3) = v_{l_2}(l_4) = 1$.

Next, construct a set of $3q$ agents $C = \{c_1, c_2, \dots, c_{3q}\}$. Let $v_{c_i}(l_r) = 3$ for each i and r where $1 \leq i \leq 3q$ and $1 \leq r \leq 4$. For each i and j where $1 \leq i, j \leq 3q$ let $v_{c_i}(c_j) = 3$ if $\{w_i, w_j\} \in E$ otherwise 2. This completes the construction of (N, V) . Note that the structure of the valuations among the agents in C reflects the graph G .

We make some remarks on the design of the constructed instance. Like before, in the reduction presented in Section 7.4.2, the design of H is derived from a particular instance that contains no j -envy-free matching. This instance comprises H as well as a single isolated agent α_z , where $v_{\alpha_z}(h_i) = 0$ for each i where $1 \leq i \leq 11$. In fact, the proof that this instance contains no j -envy-free matching can be directly derived from the proofs of lemmas appearing later in this section (Lemmas 7.44–7.48). These proofs involve lengthy case analyses, and we leave open the problem of finding a more intuitive or succinct argument (see Section 7.5 for more discussion on this).

It is straightforward to show that the reduction runs in polynomial time. To prove that the reduction is correct we show that the 3DR-AS instance (N, V) contains a j -envy-free matching if and only if the PIT instance G contains a partition into triangles.

We first show that if the PIT instance G contains a partition into triangles then the 3DR-AS instance (N, V) contains a j -envy-free matching.

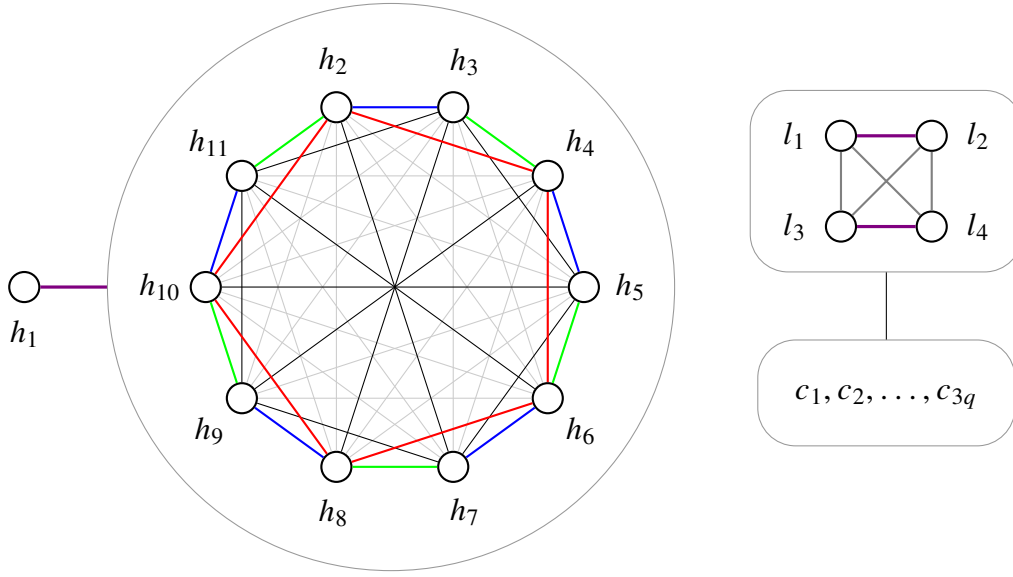


Figure 7.4: The reduction from PIT to the problem of deciding if an instance of 3DR-AS with symmetric preferences contains a j -envy-free matching. Valuation colour key: **red** - 6, **green** - 5, **blue** - 4, black - 3, **purple** - 2, grey - 1.

Lemma 7.42. *If G contains a partition into triangles then (N, V) contains a j -envy-free matching.*

Proof. Suppose G contains a partition into triangles $X = \{X_1, X_2, \dots, X_q\}$. We shall construct a matching M in (N, V) that is j -envy-free. First, add $\{h_2, h_{10}, h_{11}\}$, $\{h_5, h_6, h_8\}$, $\{h_1, h_9, h_4\}$, $\{h_3, l_1, l_2\}$ and $\{h_7, l_3, l_4\}$ to M . Next, for each triangle $X_p = \{w_i, w_j, w_k\}$ in X , add $\{c_i, c_j, c_k\}$ to M .

Suppose for a contradiction that some agent α_j exists where α_j has j -envy for some other agent α_{k_1} where $M(\alpha_{k_1}) = \{\alpha_{k_1}, \alpha_{k_2}, \alpha_{k_3}\}$. Since $N = H \cup L \cup C$ it must be that either $\alpha_{k_1} \in H$, $\alpha_{k_1} \in L$, or $\alpha_{k_1} \in C$. We show that each case leads to a contradiction. It follows that no such α_j exists and thus that M is j -envy-free.

- Suppose $\alpha_{k_1} \in H$. Either $\alpha_{k_1} \in \{h_3, h_7\}$ or $\alpha_{k_1} \in H \setminus \{h_3, h_7\}$.
 - Suppose $\alpha_{k_1} \in \{h_3, h_7\}$. Then it must be that either $\{\alpha_{k_2}, \alpha_{k_3}\} = \{l_1, l_2\}$ or $\{\alpha_{k_2}, \alpha_{k_3}\} = \{l_3, l_4\}$. Suppose firstly that $\{\alpha_{k_2}, \alpha_{k_3}\} = \{l_1, l_2\}$. We can see immediately that $\alpha_j \notin H$ since otherwise $u_{\alpha_j}(\{l_1, l_2\}) = 0$. It must also be that $\alpha_j \notin C$, since $u_{c_a}(\{l_1, l_2\}) = 6 = u_{c_a}(M)$ for any $c_a \in C$. Similarly, $u_{l_3}(\{l_1, l_2\}) = 2 = u_{l_3}(M)$ and $u_{l_4}(\{l_1, l_2\}) = 2 = u_{l_4}(M)$ so $\alpha_j \neq l_3$ and $\alpha_j \neq l_4$. This shows $\alpha_j \notin L$. We have shown that $\alpha_j \notin H$, $\alpha_j \notin C$, and $\alpha_j \notin L$, which is a contradiction. The proof for the case in which $\{\alpha_{k_2}, \alpha_{k_3}\} = \{l_3, l_4\}$ is symmetric and also leads to a contradiction.

- Suppose $\alpha_{k_1} \in H \setminus \{h_3, h_7\}$. If $\alpha_{k_1} = h_1$ then $\{\alpha_{k_2}, \alpha_{k_3}\} = \{h_4, h_9\}$ so it must be that $v_{h_4}(\alpha_j) > 2 = v_{h_4}(h_1)$ and $v_{h_9}(\alpha_j) > 2 = v_{h_9}(h_1)$, which is impossible by the design of H . The proof for every other assignment of α_{k_1} is similar: if $\alpha_{k_1} = h_2$ then $v_{h_{11}}(\alpha_j) > 5$, which is impossible. If $\alpha_{k_1} = h_4$ then $v_{h_1}(\alpha_j) > 2$, which is impossible. If $\alpha_{k_1} = h_5$ then $v_{h_6}(\alpha_j) > 5$ and $v_{h_8}(\alpha_j) > 1$, which is impossible. If $\alpha_{k_1} = h_6$ then $v_{h_8}(\alpha_j) > 6$, which is impossible. If $\alpha_{k_1} = h_8$ then $v_{h_6}(\alpha_j) > 6$, which is impossible. If $\alpha_{k_1} = h_9$ then $v_{h_1}(\alpha_j) > 2$, which is impossible. If $\alpha_{k_1} = h_{10}$ then $v_{h_2}(\alpha_j) > 6$, which is impossible. If $\alpha_{k_1} = h_{11}$ then $v_{h_2}(\alpha_j) > 5$ and $v_{h_{10}}(\alpha_j) > 4$, which is impossible.
- Suppose $\alpha_{k_1} \in C$. By construction, it must be that $\alpha_{k_1} = c_{i_1}$, $\alpha_{k_2} = c_{i_2}$, and $\alpha_{k_3} = c_{i_3}$ where $\{c_{i_1}, c_{i_2}, c_{i_3}\} \subseteq C$, where the corresponding vertices $\{w_{i_1}, w_{i_2}, w_{i_3}\}$ in G are a triangle. It follows that $v_{c_{i_2}}(c_{i_1}) = 3$. By assumption, α_j has j-envy for c_{i_1} so it must be that $v_{c_{i_2}}(\alpha_j) > v_{c_{i_2}}(c_{i_1}) = 3$, which is impossible by the design of C .
- Suppose $\alpha_{k_1} \in L$. It must be that $\alpha_{k_1} = l_{i_1}$ for some i_1 where $1 \leq i_1 \leq 4$, $\alpha_{k_2} = l_{i_2}$ for some i_2 where $1 \leq i_2 \leq 4$ and $\alpha_{k_3} = h_{i_3}$ where $i_3 \in \{3, 7\}$. If $\alpha_j \in H$ then $v_{l_{i_2}}(\alpha_j) = 0$ which contradicts the supposition that α_j has j-envy for l_{i_1} . Otherwise, if $\alpha_j \notin H$ then $v_{h_{i_3}}(\alpha_j) = 0$, which also contradicts the supposition that α_j has j-envy for l_{i_1} .

□

We now show that if the 3DR-AS instance (N, V) contains a j-envy-free matching then the PIT instance G contains a partition into triangles.

In the first part of this proof (up to and including Lemma 7.48) we focus on H . To begin, we define two possible configurations of H in an arbitrary j-envy-free matching M . If some triple $t \in M$ contains exactly one agent in H then we say that H has an *open configuration* in M . Otherwise, we say that H has a *closed configuration* in M . We shall eventually show, in Lemma 7.48, that the only possible configuration of H in M is an open configuration. In Lemmas 7.43–7.47 we prove a sequence of intermediary results.

Lemma 7.43. *If (N, V) contains a j-envy-free matching M then no triples t_1, t_2 in M exist such that t_1 contains exactly two agents in H and t_2 contains exactly one agent in H .*

Proof. Suppose for a contradiction that some such $t_1, t_2 \in M$ exist. Suppose $t_1 = \{h_{i_1}, h_{i_2}, \alpha_{j_1}\}$ and $t_2 = \{h_{i_3}, \alpha_{j_2}, \alpha_{j_3}\}$ where $1 \leq i_1, i_2, i_3 \leq 11$ and $\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3} \in N \setminus H$. Now h_{i_3} has j-envy for α_{j_1} since $u_{h_{i_3}}(M) = 0 < 2 \leq u_{h_{i_3}}(\{h_{i_1}, h_{i_2}\})$, $v_{h_{i_1}}(\alpha_{j_1}) = 0 < 1 \leq v_{h_{i_1}}(h_{i_3})$ and $v_{h_{i_2}}(\alpha_{j_1}) = 0 < 1 \leq v_{h_{i_2}}(h_{i_3})$. This contradicts our supposition that M is j-envy-free. □

Lemma 7.44. *If (N, V) contains a j-envy-free matching M , $\sigma(H, M) = 4$, and $u_{h_1}(M) < 4$ then H has an open configuration in M .*

Proof. Suppose to the contrary that $\sigma(H, M) = 4$, $u_{h_1}(M) < 4$, and H has a closed configuration in M . Since $\sigma(H, M) = 4$ it must be that three triples in M each contain exactly three agents in H and one triple in M contains exactly two agents in H . Suppose then that $t_1, t_2, t_3 \in M$ each contain exactly three agents in H and $t_4 \in M$ contains exactly two agents in H . Since $u_{h_1}(M) < 4$ by assumption, by the design of H it follows that $M(h_1)$ contains at most one agent in $H \setminus \{h_1\}$ and therefore $h_1 \in t_4$. It follows that $t_4 = \{h_1, h_{i_1}, \alpha_j\}$ where $2 \leq i_1 \leq 11$ and $\alpha_j \in N \setminus H$. We use a case analysis to prove a contradiction occurs for each possible assignment of i_1 .

- Suppose first $i_1 = 2$. If $u_{h_4}(M) \leq 7$ then h_4 has j-envy for α_j , since $u_{h_4}(M) \leq 7 < 8 = u_{h_4}(\{h_1, h_2\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_4)$, and $v_{h_2}(\alpha_j) = 0 < 6 = v_{h_2}(h_4)$. It follows that $u_{h_4}(M) \geq 8$. By the assumptions regarding the structure of M , it must be that $M(h_4)$ contains three agents in H . Since $t_4 = \{h_1, h_2, \alpha_j\}$ it must be that $M(h_4) = \{h_4, h_{i_2}, h_{i_3}\}$ where $\{i_2, i_3\} \subset \{3, 5, 6, 7, 8, 9, 10, 11\}$. Recall that $v_{h_4}(h_3) = 5$, $v_{h_4}(h_5) = 4$, $v_{h_4}(h_6) = 6$, $v_{h_4}(h_7) = 1$, $v_{h_4}(h_8) = 1$, $v_{h_4}(h_9) = 3$, $v_{h_4}(h_{10}) = 1$, and $v_{h_4}(h_{11}) = 1$. Since we established $u_{h_4}(M) \geq 8$ it follows that there are 5 possibilities: $\{i_2, i_3\} = \{3, 9\}$, $\{i_2, i_3\} = \{3, 5\}$, $\{i_2, i_3\} = \{3, 6\}$, $\{i_2, i_3\} = \{5, 6\}$, and $\{i_2, i_3\} = \{6, 9\}$, which we shall now consider.
 - Suppose $\{i_2, i_3\} = \{3, 9\}$. It follows that h_2 has j-envy for h_9 , since $u_{h_2}(M) = 2 < 10 = u_{h_2}(\{h_3, h_4\})$, $v_{h_3}(h_9) = 1 < 4 = v_{h_3}(h_2)$, and $v_{h_4}(h_9) = 3 < 6 = v_{h_4}(h_2)$. This contradicts the supposition that M is j-envy-free.
 - Suppose $\{i_2, i_3\} = \{3, 5\}$. It follows that h_2 has j-envy for h_5 , since $u_{h_2}(M) = 2 < 10 = u_{h_2}(\{h_3, h_4\})$, $v_{h_3}(h_5) = 3 < 4 = v_{h_3}(h_2)$, and $v_{h_4}(h_5) = 4 < 6 = v_{h_4}(h_2)$.
 - Suppose $\{i_2, i_3\} = \{3, 6\}$. Consider h_{11} . If $u_{h_{11}}(M) \leq 6$ then h_{11} has j-envy for α_j , since $u_{h_{11}}(M) \leq 6 < 7 = u_{h_{11}}(\{h_1, h_2\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_{11})$, and $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$. It follows that $u_{h_{11}}(M) \geq 7$. We have established that $h_2 \notin M(h_{11})$, $h_3 \notin M(h_{11})$, and $h_6 \notin M(h_{11})$ so, by the design of H , it must be that $M(h_{11}) = \{h_9, h_{10}, h_{11}\}$. Now h_2 has j-envy for h_9 , since $u_{h_2}(M) = 2 < 11 = u_{h_2}(\{h_{10}, h_{11}\})$, $v_{h_{10}}(h_9) = 5 < 6 = v_{h_{10}}(h_2)$, and $v_{h_{11}}(h_9) = 3 < 5 = v_{h_{11}}(h_2)$.
 - Suppose $\{i_2, i_3\} = \{5, 6\}$. Consider h_3 . If $u_{h_3}(M) \leq 5$ then h_3 has j-envy for α_j , since $u_{h_3}(M) \leq 5 < 6 = u_{h_3}(\{h_1, h_2\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_3)$, and $v_{h_2}(\alpha_j) = 0 < 4 = v_{h_2}(h_3)$. It follows that $u_{h_3}(M) \geq 6$. We have established that $h_2 \notin M(h_3)$, $h_4 \notin M(h_3)$, and $h_5 \notin M(h_3)$ so, by the design of H , it must be that $M(h_3) = \{h_3, h_8, h_{11}\}$. Now h_{11} has j-envy for α_j , since $u_{h_{11}}(M) = 4 < 7 = u_{h_{11}}(\{h_1, h_2\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_{11})$, and $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$.
 - Suppose $\{i_2, i_3\} = \{6, 9\}$. Consider h_{11} . If $u_{h_{11}}(M) \leq 6$ then h_{11} has j-envy for α_j , since $u_{h_{11}}(M) \leq 6 < 7 = u_{h_{11}}(\{h_1, h_2\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_{11})$,

and $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$. It follows that $u_{h_{11}}(M) \geq 7$. We have established that $h_2 \notin M(h_{11})$, $h_6 \notin M(h_{11})$, and $h_9 \notin M(h_{11})$ so, by the design of H , it must be that $M(h_{11}) = \{h_3, h_{10}, h_{11}\}$. Now h_2 has j-envy for h_{10} , since $u_{h_2}(M) = 2 < 9 = u_{h_2}(\{h_3, h_{11}\})$, $v_{h_3}(h_{10}) = 1 < 4 = v_{h_3}(h_2)$, and $v_{h_{11}}(h_{10}) = 4 < 5 = v_{h_{11}}(h_2)$.

- Suppose next $i_1 = 3$. If $u_{h_4}(M) \leq 6$ then h_4 has j-envy for α_j , since $u_{h_4}(M) \leq 6 < 7 = u_{h_4}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_4)$, and $v_{h_3}(\alpha_j) = 0 < 5 = v_{h_3}(h_4)$. It follows that $u_{h_4}(M) \geq 7$. Since $M(h_4) \neq t_4$ it must be that $M(h_4) = \{h_4, h_{i_2}, h_{i_3}\}$ where $\{i_2, i_3\} \subset \{2, 5, 6, 7, 8, 9, 10, 11\}$. Recall that $v_{h_4}(h_2) = 6$, $v_{h_4}(h_5) = 4$, $v_{h_4}(h_6) = 6$, $v_{h_4}(h_7) = 1$, $v_{h_4}(h_8) = 1$, $v_{h_4}(h_9) = 3$, $v_{h_4}(h_{10}) = 1$, and $v_{h_4}(h_{11}) = 1$. Since we established $u_{h_4}(M) \geq 7$ it follows that there are 14 possibilities: $\{i_2, i_3\} = \{2, 5\}$, $\{i_2, i_3\} = \{2, 6\}$, $\{i_2, i_3\} = \{2, 7\}$, $\{i_2, i_3\} = \{2, 8\}$, $\{i_2, i_3\} = \{2, 9\}$, $\{i_2, i_3\} = \{2, 10\}$, $\{i_2, i_3\} = \{2, 11\}$, $\{i_2, i_3\} = \{5, 6\}$, $\{i_2, i_3\} = \{5, 9\}$, $\{i_2, i_3\} = \{6, 7\}$, $\{i_2, i_3\} = \{6, 8\}$, $\{i_2, i_3\} = \{6, 9\}$, $\{i_2, i_3\} = \{6, 10\}$, and $\{i_2, i_3\} = \{6, 11\}$, which we shall now consider.

- Suppose $\{i_2, i_3\} = \{2, 5\}$. It follows that h_3 has j-envy for h_5 , since $u_{h_3}(M) = 2 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(h_5) = 1 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(h_5) = 4 < 5 = v_{h_4}(h_3)$.
- Suppose $\{i_2, i_3\} = \{2, 6\}$. If $u_{h_5}(M) \leq 4$ then h_5 has j-envy for α_j , since $u_{h_5}(M) \leq 4 < 5 = u_{h_5}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_5)$, and $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_5)$. It follows that $u_{h_5}(M) \geq 5$. We have established that $h_3 \notin M(h_5)$, $h_4 \notin M(h_5)$, and $h_6 \notin M(h_5)$ so, by the design of H , it must be that $M(h_5) = \{h_5, h_7, h_{10}\}$. It remains that $M(h_{11}) = \{h_8, h_9, h_{11}\}$. Now h_{11} has j-envy for α_j , since $u_{h_{11}}(M) = 4 < 5 = u_{h_{11}}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_{11})$, and $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_{11})$.
- Suppose $\{i_2, i_3\} = \{2, 7\}$. It follows that h_3 has j-envy for h_7 , since $u_{h_3}(M) = 2 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(h_7) = 3 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(h_7) = 1 < 5 = v_{h_4}(h_3)$.
- Suppose $\{i_2, i_3\} = \{2, 8\}$. It follows that h_3 has j-envy for h_8 , since $u_{h_3}(M) = 2 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(h_8) = 1 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(h_8) = 1 < 5 = v_{h_4}(h_3)$.
- Suppose $\{i_2, i_3\} = \{2, 9\}$. It follows that h_3 has j-envy for h_9 , since $u_{h_3}(M) = 2 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(h_9) = 1 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(h_9) = 3 < 5 = v_{h_4}(h_3)$.
- Suppose $\{i_2, i_3\} = \{2, 10\}$. If $u_{h_{11}}(M) \leq 4$ then h_{11} has j-envy for α_j , since $u_{h_{11}}(M) \leq 4 < 5 = u_{h_{11}}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_{11})$, and $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_{11})$. It follows that $u_{h_{11}}(M) \geq 5$. We have established that $h_2 \notin M(h_{11})$, $h_3 \notin M(h_{11})$, and $h_{10} \notin M(h_{11})$ so, by the design of H , it must be that $M(h_{11}) = \{h_6, h_9, h_{11}\}$. Since $h_5 \notin t_4$, it must be that $M(h_5)$ contains three agents in H and thus $M(h_5) = \{h_5, h_7, h_8\}$. Now h_6 has j-envy for h_5 , since $u_{h_6}(M) = 4 < 10 = u_{h_6}(\{h_7, h_8\})$, $v_{h_7}(h_5) = 3 < 4 = v_{h_7}(h_6)$, and $v_{h_8}(h_5) = 1 < 6 = v_{h_8}(h_6)$.

- Suppose $\{i_2, i_3\} = \{2, 11\}$. If $u_{h_5}(M) \leq 4$ then h_5 has j-envy for α_j , since $u_{h_5}(M) \leq 4 < 5 = u_{h_5}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_5)$, and $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_5)$. It follows that $u_{h_5}(M) \geq 5$. We have established that $h_3 \notin M(h_5)$ and $h_4 \notin M(h_5)$ so, by the design of H , there are three possibilities: either $M(h_5) = \{h_5, h_6, h_7\}$, $M(h_5) = \{h_5, h_6, h_{10}\}$, or $M(h_5) = \{h_5, h_7, h_{10}\}$.
 - * If $M(h_5) = \{h_5, h_6, h_7\}$ then h_4 has j-envy for h_7 , since $u_{h_4}(M) = 7 < 10 = u_{h_4}(\{h_5, h_6\})$, $v_{h_5}(h_7) = 3 < 4 = v_{h_5}(h_4)$, and $v_{h_6}(h_7) = 4 < 6 = v_{h_6}(h_4)$.
 - * If $M(h_5) = \{h_5, h_6, h_{10}\}$ then h_4 has j-envy for h_{10} , since $u_{h_4}(M) = 7 < 10 = u_{h_4}(\{h_5, h_6\})$, $v_{h_5}(h_{10}) = 3 < 4 = v_{h_5}(h_4)$, and $v_{h_6}(h_{10}) = 1 < 6 = v_{h_6}(h_4)$.
 - * If $M(h_5) = \{h_5, h_7, h_{10}\}$ then it remains that $M(h_6) = \{h_6, h_8, h_9\}$. Now h_6 has j-envy for h_{10} , since $u_{h_6}(M) = 7 < 9 = u_{h_6}(\{h_5, h_7\})$, $v_{h_5}(h_{10}) = 3 < 5 = v_{h_5}(h_6)$, and $v_{h_7}(h_{10}) = 1 < 4 = v_{h_7}(h_6)$.
- Suppose $\{i_2, i_3\} = \{5, 6\}$. If $u_{h_2}(M) \leq 5$ then h_2 has j-envy for α_j , since $u_{h_2}(M) \leq 5 < 6 = u_{h_2}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_2)$, and $v_{h_3}(\alpha_j) = 0 < 4 = v_{h_3}(h_2)$. It follows that $u_{h_2}(M) \geq 6$. Since $h_2 \notin t_4$ it must be that $M(h_2)$ contains three agents in H . Since $t_4 = \{h_1, h_3, \alpha_j\}$ and $M(h_4) = \{h_4, h_5, h_6\}$ it follows that $M(h_2) = \{h_2, h_{i_4}, h_{i_5}\}$ where $\{i_4, i_5\} \subset \{7, 8, 9, 10, 11\}$. Recall that $v_{h_2}(h_7) = 3$, $v_{h_2}(h_8) = 1$, $v_{h_2}(h_9) = 1$, $v_{h_2}(h_{10}) = 6$, and $v_{h_2}(h_{11}) = 5$. Since we established $u_{h_2}(M) \geq 6$ it follows that there are 7 possibilities: $\{h_{i_4}, h_{i_5}\} = \{7, 10\}$, $\{h_{i_4}, h_{i_5}\} = \{7, 11\}$, $\{h_{i_4}, h_{i_5}\} = \{8, 10\}$, $\{h_{i_4}, h_{i_5}\} = \{8, 11\}$, $\{h_{i_4}, h_{i_5}\} = \{9, 10\}$, $\{h_{i_4}, h_{i_5}\} = \{9, 11\}$, and $\{h_{i_4}, h_{i_5}\} = \{10, 11\}$, which we shall now consider.
 - * If $\{h_{i_4}, h_{i_5}\} = \{7, 10\}$ then it remains that $M(h_{11}) = \{h_8, h_9, h_{11}\}$. Now h_{11} has j-envy for h_7 , since $u_{h_{11}}(M) = 4 < 9 = u_{h_{11}}(\{h_2, h_{10}\})$, $v_{h_2}(h_7) = 3 < 5 = v_{h_2}(h_{11})$, and $v_{h_{10}}(h_7) = 1 < 4 = v_{h_{10}}(h_{11})$.
 - * If $\{h_{i_4}, h_{i_5}\} = \{7, 11\}$ then h_3 has j-envy for h_7 , since $u_{h_3}(M) = 2 < 7 = u_{h_3}(\{h_2, h_{11}\})$, $v_{h_2}(h_7) = 3 < 4 = v_{h_2}(h_3)$, and $v_{h_{11}}(h_7) = 1 < 3 = v_{h_{11}}(h_3)$.
 - * If $\{h_{i_4}, h_{i_5}\} = \{8, 10\}$ then it remains that $M(h_7) = \{h_7, h_9, h_{11}\}$. Now h_8 has j-envy for h_{11} , since $u_{h_8}(M) = 7 < 9 = u_{h_8}(\{h_7, h_9\})$, $v_{h_7}(h_{11}) = 1 < 5 = v_{h_7}(h_8)$, and $v_{h_9}(h_{11}) = 3 < 4 = v_{h_9}(h_8)$.
 - * If $\{h_{i_4}, h_{i_5}\} = \{8, 11\}$ then h_3 has j-envy for h_8 , since $u_{h_3}(M) = 2 < 7 = u_{h_3}(\{h_2, h_{11}\})$, $v_{h_2}(h_8) = 1 < 4 = v_{h_2}(h_3)$, and $v_{h_{11}}(h_8) = 1 < 3 = v_{h_{11}}(h_3)$.
 - * If $\{h_{i_4}, h_{i_5}\} = \{9, 10\}$ then it remains that $M(h_7) = \{h_7, h_8, h_{11}\}$. Now h_9 has j-envy for h_{11} , since $u_{h_9}(M) = 6 < 7 = u_{h_9}(\{h_7, h_8\})$, $v_{h_7}(h_{11}) = 1 < 3 = v_{h_7}(h_9)$, and $v_{h_8}(h_{11}) = 1 < 4 = v_{h_8}(h_9)$.
 - * If $\{h_{i_4}, h_{i_5}\} = \{9, 11\}$ then it remains that $M(h_{10}) = \{h_7, h_8, h_{10}\}$. Now h_{10} has j-envy for h_9 , since $u_{h_{10}}(M) = 7 < 10 = u_{h_{10}}(\{h_2, h_{11}\})$, $v_{h_2}(h_9) = 1 <$

$6 = v_{h_2}(h_{10})$, and $v_{h_{11}}(h_9) = 3 < 4 = v_{h_{11}}(h_{10})$.

* If $\{h_{i_4}, h_{i_5}\} = \{10, 11\}$ then it remains that $M(h_7) = \{h_7, h_8, h_9\}$. Now h_{10} has j-envy for h_7 , since $u_{h_{10}}(M) = 10 < 11 = u_{h_{10}}(\{h_8, h_9\})$, $v_{h_8}(h_7) = 5 < 6 = v_{h_8}(h_{10})$, and $v_{h_9}(h_7) = 3 < 5 = v_{h_9}(h_{10})$.

- Suppose $\{i_2, i_3\} = \{5, 9\}$. It follows that h_3 has j-envy for h_9 , since $u_{h_3}(M) = 2 < 8 = u_{h_3}(\{h_4, h_5\})$, $v_{h_4}(h_9) = 3 < 5 = v_{h_4}(h_3)$, and $v_{h_5}(h_9) = 1 < 3 = v_{h_5}(h_3)$.
- Suppose $\{i_2, i_3\} = \{6, 7\}$. Consider h_5 . Since $h_3 \notin M(h_5)$, $h_4 \notin M(h_5)$, $h_6 \notin M(h_5)$, and $h_7 \notin M(h_5)$ by the design of H it must be that $u_{h_5}(M) \leq 4$. It follows that h_5 has j-envy for h_7 , since $u_{h_5}(M) \leq 4 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_7) = 1 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_7) = 4 < 5 = v_{h_6}(h_5)$.
- Suppose $\{i_2, i_3\} = \{6, 8\}$. Consider h_5 . If $u_{h_5}(M) \leq 4$ then h_5 has j-envy for α_j , since $u_{h_5}(M) \leq 4 < 5 = u_{h_2}(\{h_1, h_3\})$, $v_{h_1}(\alpha_j) = 0 < 2 = v_{h_1}(h_5)$, and $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_5)$. It follows that $u_{h_5}(M) \geq 5$. Since $h_3 \notin M(h_5)$, $h_4 \notin M(h_5)$, and $h_6 \notin M(h_5)$, the only possibility is that $M(h_5) = \{h_5, h_7, h_{10}\}$. It remains that $M(h_2) = \{h_2, h_9, h_{11}\}$. Now h_{10} has j-envy for h_9 , since $u_{h_{10}}(M) = 4 < 10 = u_{h_{10}}(\{h_2, h_{11}\})$, $v_{h_2}(h_9) = 1 < 6 = v_{h_2}(h_{10})$, and $v_{h_{11}}(h_9) = 3 < 4 = v_{h_{11}}(h_{10})$.
- Suppose $\{i_2, i_3\} = \{6, 9\}$. Consider h_5 . Since $h_3 \notin M(h_5)$, $h_4 \notin M(h_5)$, and $h_6 \notin M(h_5)$ by the design of H it must be that $u_{h_5}(M) \leq 6$. It follows that h_5 has j-envy for h_9 , since $u_{h_5}(M) \leq 6 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_9) = 3 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_9) = 1 < 5 = v_{h_6}(h_5)$.
- Suppose $\{i_2, i_3\} = \{6, 10\}$. Consider h_5 . Since $h_3 \notin M(h_5)$, $h_4 \notin M(h_5)$, $h_6 \notin M(h_5)$, and $h_{10} \notin M(h_5)$ by the design of H it must be that $u_{h_5}(M) \leq 4$. It follows that h_5 has j-envy for h_{10} , since $u_{h_5}(M) \leq 4 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_{10}) = 1 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_{10}) = 1 < 5 = v_{h_6}(h_5)$.
- Suppose $\{i_2, i_3\} = \{6, 11\}$. Consider h_5 . Since $h_3 \notin M(h_5)$, $h_4 \notin M(h_5)$, and $h_6 \notin M(h_5)$ by the design of H it must be that $u_{h_5}(M) \leq 6$. It follows that h_5 has j-envy for h_{11} , since $u_{h_5}(M) \leq 6 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_{11}) = 1 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_{11}) = 3 < 5 = v_{h_6}(h_5)$.

□

Lemma 7.45. *If (N, V) contains a j-envy-free matching M , $\sigma(H, M) = 4$, and $u_{h_1}(M) = 4$ then H has an open configuration in M .*

Proof. Suppose to the contrary that $\sigma(H, M) = 4$, $u_{h_1}(M) = 4$, and H has an open configuration in M . Since $\sigma(H, M) = 4$ it must be that three triples in M each contain exactly three agents in H and one triple in M contains exactly two agents in H . Suppose $t_1, t_2, t_3 \in M$

each contain exactly three agents in H and $t_4 \in M$ contains exactly two agents in H . Since $u_{h_1}(M) = 4$, by the design of H it follows that $M(h_1)$ contains two agents in $H \setminus \{h_1\}$ and therefore $h_1 \notin t_4$. It follows that $t_4 = \{h_{i_1}, h_{i_2}, \alpha_j\}$ where $2 \leq i_1, i_2 \leq 11$ and $\alpha_j \in (N \setminus H)$. We use a case analysis to prove a contradiction occurs for each possible assignment of $\{h_{i_1}, h_{i_2}\}$ where $2 \leq i_1, i_2 \leq 11$.

As before, in the proof of Lemma 7.44, the symmetries of H allow us to shorten the case analysis. Recall that the structure of the valuations between agents $H \setminus \{h_1\}$ has five symmetries. It follows that for any possible assignment of $\{i_1, i_2\}$ there exist in total five symmetric assignments, where the case analysis for one assignment is symmetric to the case analysis for the other four assignments. Since there are $\binom{10}{2} = 45$ possible assignments of $\{i_1, i_2\}$ where $2 \leq i_1, i_2 \leq 11$ and five symmetries we need only consider the nine assignments $\{2, 3\}, \{2, 4\}, \{2, 5\}, \{2, 6\}, \{2, 7\}, \{3, 4\}, \{3, 5\}, \{3, 6\}, \{3, 7\}$, of which no two are symmetric.

- Suppose $\{i_1, i_2\} = \{2, 3\}$. Since $h_2 \notin M(h_4)$ and $h_3 \notin M(h_4)$ it must be that $u_{h_4}(M) \leq 10$. It follows that h_4 has j-envy for α_j since $u_{h_4}(M) \leq 10 < 12 = u_{h_4}(\{h_2, h_3\})$, $v_{h_2}(\alpha_j) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_3}(\alpha_j) = 0 < 5 = v_{h_3}(h_4)$. This contradicts the supposition that M is j-envy-free. The following proofs concerning other possible assignments of $\{i_1, i_2\}$ are similar in technique although in some cases we must make further deductions about the utilities of agents in H .
- Suppose $\{i_1, i_2\} = \{2, 4\}$. Since $h_2 \notin M(h_3)$ and $h_4 \notin M(h_3)$ it follows that $u_{h_3}(M) \leq 6$. It then follows that h_3 has j-envy for α_j since $u_{h_3}(M) \leq 6 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(\alpha_j) = 0 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(\alpha_j) = 0 < 5 = v_{h_4}(h_3)$.
- Suppose $\{i_1, i_2\} = \{2, 5\}$. If $u_{h_4}(M) \leq 9$ then h_4 has j-envy for α_j , since $u_{h_4}(M) \leq 9 < 10 = u_{h_4}(\{h_2, h_5\})$, $v_{h_2}(\alpha_j) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_5}(\alpha_j) = 0 < 4 = v_{h_5}(h_4)$. It follows that $u_{h_4}(M) \geq 10$. The only possibility is that $M(h_4) = \{h_3, h_4, h_6\}$. Now h_3 has j-envy for α_j since $u_{h_3}(M) = 6 < 7 = u_{h_3}(\{h_2, h_5\})$, $v_{h_2}(\alpha_j) = 0 < 4 = v_{h_2}(h_3)$, and $v_{h_5}(\alpha_j) = 0 < 3 = v_{h_5}(h_3)$.
- Suppose $\{i_1, i_2\} = \{2, 6\}$. Since $h_2 \notin M(h_4)$ and $h_6 \notin M(h_4)$ it follows that $u_{h_4}(M) \leq 9$. Now h_4 has j-envy for α_j since $u_{h_4}(M) \leq 9 < 12 = u_{h_4}(\{h_2, h_6\})$, $v_{h_2}(\alpha_j) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_6}(\alpha_j) = 0 < 6 = v_{h_6}(h_4)$.
- Suppose $\{i_1, i_2\} = \{2, 7\}$. If $u_{h_4}(M) \leq 6$ then h_4 has j-envy for α_j , since $u_{h_4}(M) \leq 6 < 7 = u_{h_4}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_4)$. It follows that $u_{h_4}(M) \geq 7$. By the design of H , it follows that $M(h_4)$ contains three agents in H . Since $t_4 = \{h_2, h_7, \alpha_j\}$ it follows that $M(h_4) = \{h_4, h_{i_3}, h_{i_4}\}$ where $\{i_3, i_4\} \subset \{1, 3, 5, 6, 8, 9, 10, 11\}$. Recall that $v_{h_4}(h_1) = 2$, $v_{h_4}(h_3) = 5$, $v_{h_4}(h_5) = 4$, $v_{h_4}(h_6) = 6$,

$v_{h_4}(h_8) = 1$, $v_{h_4}(h_9) = 3$, $v_{h_4}(h_{10}) = 1$, and $v_{h_4}(h_{11}) = 1$. Since we established $u_{h_4}(M) \geq 7$ it follows that there are 11 possibilities: $\{i_3, i_4\} = \{1, 3\}$, $\{i_3, i_4\} = \{1, 6\}$, $\{i_3, i_4\} = \{3, 5\}$, $\{i_3, i_4\} = \{3, 6\}$, $\{i_3, i_4\} = \{3, 9\}$, $\{i_3, i_4\} = \{5, 6\}$, $\{i_3, i_4\} = \{5, 9\}$, $\{i_3, i_4\} = \{6, 8\}$, $\{i_3, i_4\} = \{6, 9\}$, $\{i_3, i_4\} = \{6, 10\}$, and $\{i_3, i_4\} = \{6, 11\}$, which we shall now consider.

- Suppose $\{i_3, i_4\} = \{1, 3\}$. It follows that h_2 has j-envy for h_1 since $u_{h_2}(M) = 3 < 10 = u_{h_2}(\{h_3, h_4\})$, $v_{h_3}(h_1) = 2 < 4 = v_{h_3}(h_2)$, and $v_{h_4}(h_1) = 2 < 6 = v_{h_4}(h_2)$.
- Suppose $\{i_3, i_4\} = \{1, 6\}$. Consider h_5 . It must be that $u_{h_5}(M) \leq 6$. Now h_5 has j-envy for h_1 since $u_{h_5}(M) \leq 6 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_1) = 2 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_1) = 2 < 5 = v_{h_6}(h_5)$.
- Suppose $\{i_3, i_4\} = \{3, 5\}$. It follows that h_2 has j-envy for h_5 since $u_{h_2}(M) = 3 < 10 = u_{h_2}(\{h_3, h_4\})$, $v_{h_3}(h_5) = 3 < 4 = v_{h_3}(h_2)$, and $v_{h_4}(h_5) = 4 < 6 = v_{h_4}(h_2)$.
- Suppose $\{i_3, i_4\} = \{3, 6\}$. In this case, consider $M(h_1)$. Since $h_1 \notin t_4$ it follows that $M(h_1)$ contains three agents in H . Suppose $M(h_1) = \{h_1, h_{i_5}, h_{i_6}\}$ where $2 \leq i_5, i_6 \leq 11$. Since we have established $M(h_2) = \{h_2, h_7, \alpha_j\}$ and $M(h_4) = \{h_3, h_4, h_6\}$ it follows that $\{h_{i_5}, h_{i_6}\} \subset \{5, 8, 9, 10, 11\}$. Thus there are $\binom{5}{2} = 10$ possible assignments of $\{h_{i_5}, h_{i_6}\}$, which we shall now consider.
 - * If $\{h_{i_5}, h_{i_6}\} = \{5, 8\}$ then h_7 has j-envy for h_1 since $u_{h_7}(M) = 3 < 8 = u_{h_7}(\{h_5, h_8\})$, $v_{h_5}(h_1) = 2 < 3 = v_{h_5}(h_7)$, and $v_{h_8}(h_1) = 2 < 5 = v_{h_8}(h_7)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{5, 9\}$ then h_7 has j-envy for h_1 since $u_{h_7}(M) = 3 < 6 = u_{h_7}(\{h_5, h_9\})$, $v_{h_5}(h_1) = 2 < 3 = v_{h_5}(h_7)$, and $v_{h_9}(h_1) = 2 < 3 = v_{h_9}(h_7)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{5, 10\}$ then $u_{h_{10}}(M) = 5$. It follows that h_{10} has j-envy for α_j since $u_{h_{10}}(M) = 5 < 7 = u_{h_{10}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 6 = v_{h_2}(h_{10})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{10})$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{5, 11\}$ then $u_{h_{11}}(M) = 3$. It follows that h_{11} has j-envy for α_j since $u_{h_{11}}(M) = 3 < 6 = u_{h_{11}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{11})$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{8, 9\}$ then h_7 has j-envy for h_1 since $u_{h_7}(M) = 3 < 8 = u_{h_7}(\{h_8, h_9\})$, $v_{h_8}(h_1) = 2 < 5 = v_{h_8}(h_7)$, and $v_{h_9}(h_1) = 2 < 3 = v_{h_9}(h_7)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{8, 10\}$ then it remains that $M(h_9) = \{h_5, h_9, h_{11}\}$ and thus $u_{h_9}(M) = u_{\{h_5, h_{11}\}} = 4$. It follows that h_9 has j-envy for h_1 since $u_{h_9}(M) = 4 < 9 = u_{h_9}(\{h_8, h_{10}\})$, $v_{h_8}(h_1) = 2 < 4 = v_{h_8}(h_9)$, and $v_{h_{10}}(h_1) = 2 < 5 = v_{h_{10}}(h_9)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{8, 11\}$ then $u_{h_{11}}(M) = 3$. It follows that h_{11} has j-envy for α_j since $u_{h_{11}}(M) = 3 < 6 = u_{h_{11}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{11})$.

- * If $\{h_{i_5}, h_{i_6}\} = \{9, 10\}$ then it remains that $M(h_9) = \{h_5, h_8, h_{11}\}$ and thus $u_{h_{11}}(M) = u_{\{h_5, h_8\}} = 2$. It follows that h_{11} has j-envy for α_j since $u_{h_{11}}(M) = 2 < 6 = u_{h_{11}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{11})$.
- * If $\{h_{i_5}, h_{i_6}\} = \{9, 11\}$ then $u_{h_{11}}(M) = 5$. It follows that h_{11} has j-envy for α_j since $u_{h_{11}}(M) = 5 < 6 = u_{h_{11}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{11})$.
- * If $\{h_{i_5}, h_{i_6}\} = \{10, 11\}$ then h_2 has j-envy for h_1 since $u_{h_2}(M) = 3 < 11 = u_{h_2}(\{h_{10}, h_{11}\})$, $v_{h_{10}}(h_1) = 2 < 6 = v_{h_{10}}(h_2)$, and $v_{h_{11}}(h_1) = 2 < 5 = v_{h_{11}}(h_2)$.
- Suppose $\{i_3, i_4\} = \{3, 9\}$. It follows that h_2 has j-envy for h_9 since $u_{h_2}(M) = 3 < 10 = u_{h_2}(\{h_3, h_4\})$, $v_{h_3}(h_9) = 1 < 4 = v_{h_3}(h_2)$, and $v_{h_4}(h_9) = 3 < 6 = v_{h_4}(h_2)$.
- Suppose $\{i_3, i_4\} = \{5, 6\}$. Consider h_3 . If $u_{h_3}(M) \leq 4$ then h_3 has j-envy for α_j , since $u_{h_3}(M) \leq 4 < 5 = u_{h_3}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 4 = v_{h_2}(h_3)$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_3)$. It follows that $u_{h_3}(M) \geq 5$. We have established that $h_4 \notin M(h_3)$, $h_5 \notin M(h_3)$ and $h_2 \notin M(h_3)$ so, by the design of H , there are three possibilities: either $M(h_3) = \{h_1, h_3, h_8\}$, $M(h_3) = \{h_1, h_3, h_{11}\}$, or $M(h_3) = \{h_3, h_8, h_{11}\}$.
 - * If $M(h_3) = \{h_1, h_3, h_8\}$ then $u_{h_8}(M) = 5$. It follows that h_8 has j-envy for α_j , since $u_{h_8}(M) = 5 < 6 = u_{h_8}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 1 = v_{h_2}(h_8)$, and $v_{h_7}(\alpha_j) = 0 < 5 = v_{h_7}(h_8)$.
 - * If $M(h_3) = \{h_1, h_3, h_{11}\}$ then $u_{h_{11}}(M) = 5$. It follows that h_{11} has j-envy for α_j , since $u_{h_{11}}(M) = 5 < 6 = u_{h_{11}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{11})$.
 - * If $M(h_3) = \{h_3, h_8, h_{11}\}$ then $u_{h_8}(M) = 4$. It follows that h_8 has j-envy for α_j , since $u_{h_8}(M) = 4 < 6 = u_{h_8}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 1 = v_{h_2}(h_8)$, and $v_{h_7}(\alpha_j) = 0 < 5 = v_{h_7}(h_8)$.
- Suppose $\{i_3, i_4\} = \{5, 9\}$. Consider h_6 . It must be that $u_{h_6}(M) \leq 9$. Now h_6 has j-envy for h_9 since $u_{h_6}(M) \leq 9 < 11 = u_{h_6}(\{h_4, h_5\})$, $v_{h_4}(h_9) = 3 < 6 = v_{h_4}(h_6)$, and $v_{h_5}(h_9) = 1 < 5 = v_{h_5}(h_6)$.
- Suppose $\{i_3, i_4\} = \{6, 8\}$. Consider h_{10} . If $u_{h_{10}}(M) \leq 6$ then h_{10} has j-envy for α_j , since $u_{h_{10}}(M) \leq 6 < 7 = u_{h_{10}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 6 = v_{h_2}(h_{10})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{10})$. It follows that $u_{h_{10}}(M) \geq 7$. We have established that $h_2 \notin M(h_{10})$ and $h_8 \notin M(h_{10})$ so, by the design of H , there are four possibilities: either $M(h_{10}) = \{h_1, h_9, h_{10}\}$, $M(h_{10}) = \{h_5, h_9, h_{10}\}$, $M(h_{10}) = \{h_5, h_{10}, h_{11}\}$, or $M(h_{10}) = \{h_9, h_{10}, h_{11}\}$.

- * If $M(h_{10}) = \{h_1, h_9, h_{10}\}$ then h_8 has j-envy for h_1 , since $u_{h_8}(M) = 7 < 10 = u_{h_8}(\{h_9, h_{10}\})$, $v_{h_9}(h_1) = 2 < 4 = v_{h_9}(h_8)$, and $v_{h_{10}}(h_1) = 2 < 6 = v_{h_{10}}(h_8)$.
 - * If $M(h_{10}) = \{h_5, h_9, h_{10}\}$ then h_8 has j-envy for h_5 , since $u_{h_8}(M) = 7 < 10 = u_{h_8}(\{h_9, h_{10}\})$, $v_{h_9}(h_5) = 1 < 4 = v_{h_9}(h_8)$, and $v_{h_{10}}(h_5) = 3 < 6 = v_{h_{10}}(h_8)$.
 - * If $M(h_{10}) = \{h_5, h_{10}, h_{11}\}$ then $u_{h_{11}}(M) = 5$. It follows that h_{11} has j-envy for α_j , since $u_{h_{11}}(M) = 5 < 6 = u_{h_{11}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_j) = 0 < 5 = v_{h_2}(h_{11})$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_{11})$.
 - * If $M(h_3) = \{h_9, h_{10}, h_{11}\}$ then h_8 has j-envy for h_{11} , since $u_{h_8}(M) = 7 < 10 = u_{h_8}(\{h_9, h_{10}\})$, $v_{h_9}(h_{11}) = 3 < 4 = v_{h_9}(h_8)$, and $v_{h_{10}}(h_{11}) = 4 < 6 = v_{h_{10}}(h_8)$.
- Suppose $\{i_3, i_4\} = \{6, 9\}$. It must be that $u_{h_5}(M) \leq 6$. Now h_5 has j-envy for h_9 since $u_{h_5}(M) \leq 6 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_9) = 3 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_9) = 1 < 5 = v_{h_6}(h_5)$.
 - Suppose $\{i_3, i_4\} = \{6, 10\}$. It must be that $u_{h_5}(M) \leq 5$. Now h_5 has j-envy for h_{10} since $u_{h_5}(M) \leq 5 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_{10}) = 1 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_{10}) = 1 < 5 = v_{h_6}(h_5)$.
 - Suppose $\{i_3, i_4\} = \{6, 11\}$. It must be that $u_{h_5}(M) \leq 6$. Now h_5 has j-envy for h_{11} since $u_{h_5}(M) \leq 6 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_{11}) = 1 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_{11}) = 3 < 5 = v_{h_6}(h_5)$.
- Suppose $\{i_1, i_2\} = \{3, 4\}$. If $u_{h_2}(M) \leq 9$ then h_2 has j-envy for α_j , since $u_{h_2}(M) \leq 9 < 10 = u_{h_2}(\{h_3, h_4\})$, $v_{h_3}(\alpha_j) = 0 < 4 = v_{h_3}(h_2)$, and $v_{h_4}(\alpha_j) = 0 < 6 = v_{h_4}(h_2)$. It follows that $u_{h_2}(M) \geq 10$. The only possibility is that $M(h_2) = \{h_2, h_{10}, h_{11}\}$. Now consider h_5 . If $u_{h_5}(M) \leq 6$ then h_5 has j-envy for α_j , since $u_{h_5}(M) \leq 6 < 7 = u_{h_5}(\{h_3, h_4\})$, $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_5)$, and $v_{h_4}(\alpha_j) = 0 < 4 = v_{h_4}(h_5)$. It follows that $u_{h_5}(M) \geq 7$. Since we have established $M(h_2) = \{h_2, h_{10}, h_{11}\}$ and $t_4 = \{h_3, h_4, \alpha_j\}$, there are just two possibilities: either $M(h_5) = \{h_1, h_5, h_6\}$ or $M(h_5) = \{h_5, h_6, h_7\}$.
 - If $M(h_5) = \{h_1, h_5, h_6\}$ then h_4 has j-envy for h_1 since $u_{h_4}(M) = 5 < 10 = u_{h_4}(\{h_5, h_6\})$, $v_{h_5}(h_1) = 2 < 4 = v_{h_5}(h_4)$, and $v_{h_6}(h_1) = 2 < 6 = v_{h_6}(h_4)$.
 - If $M(h_5) = \{h_5, h_6, h_7\}$ then h_4 has j-envy for h_7 since $u_{h_4}(M) = 5 < 10 = u_{h_4}(\{h_5, h_6\})$, $v_{h_5}(h_7) = 3 < 4 = v_{h_5}(h_4)$, and $v_{h_6}(h_7) = 4 < 6 = v_{h_6}(h_4)$.
 - Suppose $\{i_1, i_2\} = \{3, 5\}$. If $u_{h_4}(M) \leq 8$ then h_4 has j-envy for α_j , since $u_{h_4}(M) \leq 8 < 9 = u_{h_4}(\{h_3, h_5\})$, $v_{h_3}(\alpha_j) = 0 < 5 = v_{h_3}(h_4)$, and $v_{h_5}(\alpha_j) = 0 < 4 = v_{h_5}(h_4)$. It follows that $u_{h_4}(M) \geq 9$. There are three possibilities: either $M(h_4) = \{h_2, h_4, h_6\}$, $M(h_4) = \{h_2, h_4, h_9\}$, or $M(h_4) = \{h_4, h_6, h_9\}$.

- Suppose $M(h_4) = \{h_2, h_4, h_6\}$. In this case, consider $M(h_1)$. Since $h_1 \notin t_4$ it follows that $M(h_1)$ contains three agents in H . Suppose $M(h_1) = \{h_1, h_{i_5}, h_{i_6}\}$ where $1 \leq i_5, i_6 \leq 11$. Since we have established $M(h_3) = \{h_3, h_5, \alpha_j\}$ and $M(h_2) = \{h_2, h_4, h_6\}$ it follows that $\{h_{i_5}, h_{i_6}\} \subset \{7, 8, 9, 10, 11\}$. Thus there are $\binom{5}{2} = 10$ possible assignments of $\{h_{i_5}, h_{i_6}\}$, which we shall now consider.
 - * If $\{h_{i_5}, h_{i_6}\} = \{7, 8\}$ then h_6 has j-envy for h_1 , since $u_{h_6}(M) = 7 < 10 = u_{h_6}(\{h_7, h_8\})$, $v_{h_7}(h_1) = 2 < 4 = v_{h_7}(h_6)$, and $v_{h_8}(h_1) = 2 < 6 = v_{h_8}(h_6)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{7, 9\}$ then it remains that $M(h_8) = \{h_8, h_{10}, h_{11}\}$. It follows that h_8 has j-envy for h_1 , since $u_{h_8}(M) = 7 < 9 = u_{h_8}(\{h_7, h_9\})$, $v_{h_7}(h_1) = 2 < 5 = v_{h_7}(h_8)$, and $v_{h_9}(h_1) = 2 < 4 = v_{h_9}(h_8)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{7, 10\}$ then h_7 has j-envy for α_j , since $u_{h_7}(M) = 3 < 4 = u_{h_7}(\{h_3, h_5\})$, $v_{h_3}(\alpha_j) = 0 < 1 = v_{h_3}(h_7)$, and $v_{h_5}(\alpha_j) = 0 < 3 = v_{h_5}(h_7)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{7, 11\}$ then h_7 has j-envy for α_j , since $u_{h_7}(M) = 3 < 4 = u_{h_7}(\{h_3, h_5\})$, $v_{h_3}(\alpha_j) = 0 < 1 = v_{h_3}(h_7)$, and $v_{h_5}(\alpha_j) = 0 < 3 = v_{h_5}(h_7)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{8, 9\}$ then it remains that $M(h_7) = \{h_7, h_{10}, h_{11}\}$. It follows that h_7 has j-envy for h_1 , since $u_{h_7}(M) = 2 < 8 = u_{h_7}(\{h_8, h_9\})$, $v_{h_8}(h_1) = 2 < 5 = v_{h_8}(h_7)$, and $v_{h_9}(h_1) = 2 < 3 = v_{h_9}(h_7)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{8, 10\}$ then it remains that $M(h_9) = \{h_7, h_9, h_{11}\}$. It follows that h_9 has j-envy for h_1 , since $u_{h_9}(M) = 6 < 9 = u_{h_9}(\{h_8, h_{10}\})$, $v_{h_8}(h_1) = 2 < 4 = v_{h_8}(h_9)$, and $v_{h_{10}}(h_1) = 2 < 5 = v_{h_{10}}(h_9)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{8, 11\}$ then h_{11} has j-envy for α_j , since $u_{h_{11}}(M) = 3 < 4 = u_{h_{11}}(\{h_3, h_5\})$, $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_{11})$, and $v_{h_5}(\alpha_j) = 0 < 1 = v_{h_5}(h_{11})$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{9, 10\}$ then it remains that $M(h_{11}) = \{h_7, h_8, h_{11}\}$. It follows that h_{11} has j-envy for h_1 , since $u_{h_{11}}(M) = 2 < 7 = u_{h_{11}}(\{h_9, h_{10}\})$, $v_{h_9}(h_1) = 2 < 3 = v_{h_9}(h_{11})$, and $v_{h_{10}}(h_1) = 2 < 4 = v_{h_{10}}(h_{11})$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{9, 11\}$ then it remains that $M(h_{10}) = \{h_7, h_8, h_{10}\}$. It follows that h_{10} has j-envy for h_1 , since $u_{h_{10}}(M) = 7 < 9 = u_{h_{10}}(\{h_9, h_{11}\})$, $v_{h_9}(h_1) = 2 < 5 = v_{h_9}(h_{10})$, and $v_{h_{11}}(h_1) = 2 < 4 = v_{h_{11}}(h_{10})$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{10, 11\}$ then h_2 has j-envy for h_1 , since $u_{h_2}(M) = 7 < 11 = u_{h_2}(\{h_{10}, h_{11}\})$, $v_{h_{10}}(h_1) = 2 < 6 = v_{h_{10}}(h_2)$, and $v_{h_{11}}(h_1) = 2 < 5 = v_{h_{11}}(h_2)$.
- Suppose $M(h_4) = \{h_2, h_4, h_9\}$. It follows that h_3 has j-envy for h_9 , since $u_{h_3}(M) = 3 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(h_9) = 1 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(h_9) = 3 < 5 = v_{h_4}(h_3)$.
- Suppose $M(h_4) = \{h_4, h_6, h_9\}$. It follows that h_5 has j-envy for h_9 , since $u_{h_5}(M) = 3 < 9 = u_{h_5}(\{h_4, h_6\})$, $v_{h_4}(h_9) = 3 < 4 = v_{h_4}(h_5)$, and $v_{h_6}(h_9) = 1 < 5 = v_{h_6}(h_5)$.

- Suppose $\{i_1, i_2\} = \{3, 6\}$. It must be that $u_{h_4}(M) \leq 10$. It follows that h_4 has j-envy for α_j since $u_{h_4}(M) \leq 10 < 11 = u_{h_4}(\{h_3, h_6\})$, $v_{h_3}(\alpha_j) = 0 < 5 = v_{h_3}(h_4)$, and $v_{h_6}(\alpha_j) = 0 < 6 = v_{h_6}(h_4)$.
- Suppose $\{i_1, i_2\} = \{3, 7\}$. If $u_{h_8}(M) \leq 7$ then h_8 has j-envy for α_j , since $u_{h_8}(M) \leq 7 < 8 = u_{h_8}(\{h_3, h_7\})$, $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_8)$, and $v_{h_7}(\alpha_j) = 0 < 5 = v_{h_7}(h_8)$. It follows that $u_{h_8}(M) \geq 8$. By the design of H , it follows that $M(h_8)$ contains three agents in H . Since $t_4 = \{h_3, h_7, \alpha_j\}$ it must be that $M(h_8) = \{h_8, h_{i_3}, h_{i_4}\}$ where $\{i_3, i_4\} \subset \{1, 2, 4, 5, 6, 9, 10, 11\}$. Recall that $v_{h_8}(h_1) = 2$, $v_{h_8}(h_2) = 1$, $v_{h_8}(h_4) = 1$, $v_{h_8}(h_5) = 1$, $v_{h_8}(h_6) = 6$, $v_{h_8}(h_9) = 4$, $v_{h_8}(h_{10}) = 6$, and $v_{h_8}(h_{11}) = 1$. Since we established $u_{h_8}(M) \geq 8$ it follows that there are five possibilities: $\{i_3, i_4\} = \{1, 6\}$, $\{i_3, i_4\} = \{1, 10\}$, $\{i_3, i_4\} = \{6, 9\}$, $\{i_3, i_4\} = \{6, 10\}$, and $\{i_3, i_4\} = \{9, 10\}$, which we shall now consider.

- Suppose $\{i_3, i_4\} = \{1, 6\}$. It follows that h_7 has j-envy for h_1 , since $u_{h_7}(M) = 1 < 9 = u_{h_7}(\{h_6, h_8\})$, $v_{h_6}(h_1) = 2 < 4 = v_{h_6}(h_7)$, and $v_{h_8}(h_1) = 2 < 5 = v_{h_8}(h_7)$.
- Suppose $\{i_3, i_4\} = \{1, 10\}$. Consider h_9 . It follows that $u_{h_9}(M) \leq 6$. Now h_9 has j-envy for h_1 , since $u_{h_9}(M) \leq 6 < 9 = u_{h_9}(\{h_8, h_{10}\})$, $v_{h_8}(h_1) = 2 < 4 = v_{h_8}(h_9)$, and $v_{h_{10}}(h_1) = 2 < 5 = v_{h_{10}}(h_9)$.
- Suppose $\{i_3, i_4\} = \{6, 9\}$. It follows that h_7 has j-envy for h_9 , since $u_{h_7}(M) = 1 < 9 = u_{h_7}(\{h_6, h_8\})$, $v_{h_6}(h_9) = 1 < 4 = v_{h_6}(h_7)$, and $v_{h_8}(h_9) = 4 < 5 = v_{h_8}(h_7)$.
- Suppose $\{i_3, i_4\} = \{6, 10\}$. In this case, consider $M(h_1)$. Since $h_1 \notin t_4$ it follows that $M(h_1)$ contains three agents in H . Suppose $M(h_1) = \{h_1, h_{i_5}, h_{i_6}\}$ where $2 \leq i_5, i_6 \leq 11$. Since we have established $M(h_3) = \{h_3, h_7, \alpha_j\}$ and $M(h_6) = \{h_6, h_8, h_{10}\}$ it follows that $\{h_{i_5}, h_{i_6}\} \subset \{2, 4, 5, 9, 11\}$. Thus there are $\binom{5}{2} = 10$ possible assignments of $\{h_{i_5}, h_{i_6}\}$, which we shall now consider.
 - * If $\{h_{i_5}, h_{i_6}\} = \{2, 4\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(h_1) = 2 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(h_1) = 2 < 5 = v_{h_4}(h_3)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{2, 5\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 7 = u_{h_3}(\{h_2, h_5\})$, $v_{h_2}(h_1) = 2 < 4 = v_{h_2}(h_3)$, and $v_{h_5}(h_1) = 2 < 3 = v_{h_5}(h_3)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{2, 9\}$ then it remains that $M(h_{11}) = \{h_4, h_5, h_{11}\}$. Now h_3 has j-envy for h_{11} , since $u_{h_3}(M) = 1 < 8 = u_{h_3}(\{h_4, h_5\})$, $v_{h_4}(h_{11}) = 1 < 5 = v_{h_4}(h_3)$, and $v_{h_5}(h_{11}) = 1 < 3 = v_{h_5}(h_3)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{2, 11\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 7 = u_{h_3}(\{h_2, h_{11}\})$, $v_{h_2}(h_1) = 2 < 4 = v_{h_2}(h_3)$, and $v_{h_{11}}(h_1) = 2 < 3 = v_{h_{11}}(h_3)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{4, 5\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 8 = u_{h_3}(\{h_5, h_5\})$, $v_{h_4}(h_1) = 2 < 5 = v_{h_4}(h_3)$, and $v_{h_5}(h_1) = 2 < 3 = v_{h_5}(h_3)$.

- * If $\{h_{i_5}, h_{i_6}\} = \{4, 9\}$ then h_4 has j-envy for α_j , since $u_{h_4}(M) = 5 < 6 = u_{h_4}(\{h_3, h_7\})$, $v_{h_3}(\alpha_j) = 0 < 5 = v_{h_3}(h_4)$, and $v_{h_7}(\alpha_j) = 0 < 1 = v_{h_7}(h_4)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{4, 11\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 8 = u_{h_3}(\{h_4, h_{11}\})$, $v_{h_4}(h_1) = 2 < 5 = v_{h_4}(h_3)$, and $v_{h_{11}}(h_1) = 2 < 3 = v_{h_{11}}(h_3)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{5, 9\}$ then h_5 has j-envy for α_j , since $u_{h_5}(M) = 3 < 6 = u_{h_5}(\{h_3, h_7\})$, $v_{h_3}(\alpha_j) = 0 < 3 = v_{h_3}(h_5)$, and $v_{h_7}(\alpha_j) = 0 < 3 = v_{h_7}(h_5)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{5, 11\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 6 = u_{h_3}(\{h_5, h_{11}\})$, $v_{h_5}(h_1) = 2 < 3 = v_{h_5}(h_3)$, and $v_{h_{11}}(h_1) = 2 < 3 = v_{h_{11}}(h_3)$.
 - * If $\{h_{i_5}, h_{i_6}\} = \{9, 11\}$ then h_{10} has j-envy for h_1 , since $u_{h_{10}}(M) = 7 < 9 = u_{h_{10}}(\{h_9, h_{11}\})$, $v_{h_9}(h_1) = 2 < 5 = v_{h_9}(h_{10})$, and $v_{h_{11}}(h_1) = 2 < 4 = v_{h_{11}}(h_{10})$.
- Suppose $\{i_3, i_4\} = \{9, 10\}$. Consider h_2 . If $u_{h_2}(M) \leq 6$ then h_2 has j-envy for α_j , since $u_{h_2}(M) \leq 6 < 7 = u_{h_2}(\{h_3, h_7\})$, $v_{h_3}(\alpha_j) = 0 < 4 = v_{h_3}(h_2)$, and $v_{h_7}(\alpha_j) = 0 < 3 = v_{h_7}(h_2)$. It follows that $u_{h_2}(M) \geq 7$. We have established that $h_3 \notin M(h_2)$, $h_7 \notin M(h_2)$, and $h_{10} \notin M(h_2)$ so, by the design of H , there are three possibilities: either $M(h_2) = \{h_1, h_2, h_4\}$, $M(h_2) = \{h_1, h_2, h_{11}\}$, or $M(h_2) = \{h_2, h_4, h_{11}\}$.
- * If $M(h_2) = \{h_1, h_2, h_4\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(h_1) = 2 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(h_1) = 2 < 5 = v_{h_4}(h_3)$.
 - * If $M(h_2) = \{h_1, h_2, h_{11}\}$ then h_3 has j-envy for h_1 , since $u_{h_3}(M) = 1 < 7 = u_{h_3}(\{h_2, h_{11}\})$, $v_{h_2}(h_1) = 2 < 4 = v_{h_2}(h_3)$, and $v_{h_{11}}(h_1) = 2 < 3 = v_{h_{11}}(h_3)$.
 - * If $M(h_2) = \{h_2, h_4, h_{11}\}$ then it remains that $M(h_1) = \{h_1, h_5, h_6\}$. Now h_7 has j-envy for h_1 , since $u_{h_7}(M) = 1 < 7 = u_{h_7}(\{h_5, h_6\})$, $v_{h_5}(h_1) = 2 < 3 = v_{h_5}(h_7)$, and $v_{h_6}(h_1) = 2 < 4 = v_{h_6}(h_7)$.

□

Lemma 7.46. *If (N, V) contains a j-envy-free matching M and $\sigma(H, M) = 4$ then H has an open configuration in M .*

Proof. Suppose $\sigma(H, M) = 4$. Consider $u_{h_1}(M)$. By the design of H , it must be that $2 \leq u_{h_1}(M) \leq 4$. If $u_{h_1}(M) < 4$ then Lemma 7.44 shows that H has an open configuration in M . If $u_{h_1}(M) = 4$ then Lemma 7.45 shows that H has an open configuration in M . □

Lemma 7.47. *If (N, V) contains a j-envy-free matching M and $\sigma(H, M) = 5$ then H has an open configuration in M .*

Proof. Suppose, to the contrary, that $\sigma(H, M) = 5$ and H has a closed configuration in M . Then it must be that four triples in M each contain exactly two agents in H and one triple

in M contains exactly three agents in H . Suppose $t_1, t_2, t_3, t_4 \in M$ each contain exactly two agents in H and $t_5 \in M$ contains exactly three agents in H , where $t_1 = \{h_{i_1}, h_{i_2}, \alpha_{j_1}\}$ where $1 \leq i_1, i_2 \leq 11$ and $\alpha_{j_1} \in N \setminus H$. We use a case analysis on $v_{h_{i_1}}(h_{i_2})$ to prove a contradiction. Note that by the design of H it must be that $1 \leq v_{h_{i_1}}(h_{i_2}) \leq 6$.

- Suppose $v_{h_{i_1}}(h_{i_2}) = 6$. By the symmetry of H , assume without loss of generality that $\{i_1, i_2\} = \{2, 4\}$. It follows that $u_{h_3}(M) \leq 6$. Now h_3 has j-envy for α_{j_1} since $u_{h_3}(M) \leq 6 < 9 = u_{h_3}(\{h_2, h_4\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 4 = v_{h_2}(h_3)$, and $v_{h_4}(\alpha_{j_1}) = 0 < 5 = v_{h_4}(h_3)$. This contradicts the supposition that M is j-envy-free. We shall use a similar technique to prove a contradiction when considering the other cases of $v_{h_{i_1}}(h_{i_2})$.
- Suppose $v_{h_{i_1}}(h_{i_2}) = 5$. Assume without loss of generality that $\{i_1, i_2\} = \{3, 4\}$. If $u_{h_2}(M) \leq 9$ then h_2 has j-envy for α_{j_1} since $u_{h_2}(M) \leq 9 < 10 = u_{h_2}(\{h_3, h_4\})$, $v_{h_3}(\alpha_{j_1}) = 0 < 4 = v_{h_3}(h_2)$, and $v_{h_4}(\alpha_{j_1}) = 0 < 6 = v_{h_4}(h_2)$. It follows that $u_{h_2}(M) \geq 10$. The only possibility is that $M(h_2) = \{h_2, h_{10}, h_{11}\}$. It must be that $M(h_2) = t_5$. We now consider h_6 . Since $h_6 \notin t_5$ it must be that either $h_6 \in t_2$, $h_6 \in t_3$, or $h_6 \in t_4$. Assume without loss of generality that $h_6 \in t_2$ and that $t_2 = \{h_6, h_{i_3}, \alpha_{j_2}\}$ where $1 \leq i_3 \leq 11$ and $\alpha_{j_2} \in N \setminus H$. If $u_{h_6}(M) \leq 6$ then h_6 has j-envy for α_{j_1} since $u_{h_6}(M) \leq 6 < 7 = u_{h_6}(\{h_3, h_4\})$, $v_{h_3}(\alpha_{j_1}) = 0 < 1 = v_{h_3}(h_6)$, and $v_{h_4}(\alpha_{j_1}) = 0 < 6 = v_{h_4}(h_6)$. It follows that $u_{h_6}(M) \geq 7$. Since $v_{h_6}(\alpha_{j_2}) = 0$ it follows that $v_{h_6}(h_{i_3}) = u_{h_6}(M) \geq 7$, which is a contradiction.
- Suppose $v_{h_{i_1}}(h_{i_2}) = 4$. Assume without loss of generality that $\{i_1, i_2\} = \{2, 3\}$. If $u_{h_4}(M) \leq 10$ then h_4 has j-envy for α_{j_1} since $u_{h_4}(M) \leq 10 < 11 = u_{h_4}(\{h_2, h_3\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_3}(\alpha_{j_1}) = 0 < 5 = v_{h_3}(h_4)$. It follows that $u_{h_4}(M) \geq 11$ which, since $h_2 \notin M(h_4)$, is impossible. This contradicts our supposition that $v_{h_{i_1}}(h_{i_2}) = 4$.
- Suppose $v_{h_{i_1}}(h_{i_2}) = 3$. We assume without loss of generality that either $\{i_1, i_2\} = \{3, 11\}$ or $\{i_1, i_2\} = \{2, 7\}$.
 - Suppose $\{i_1, i_2\} = \{3, 11\}$. If $u_{h_2}(M) \leq 8$ then h_2 has j-envy for α_{j_1} since $u_{h_2}(M) \leq 8 < 9 = u_{h_2}(\{h_3, h_{11}\})$, $v_{h_3}(\alpha_{j_1}) = 0 < 4 = v_{h_3}(h_2)$, and $v_{h_{11}}(\alpha_{j_1}) = 0 < 5 = v_{h_{11}}(h_2)$. It follows that $u_{h_2}(M) \geq 9$. By the design of H , it follows that $M(h_2) = t_5$. Now consider h_1 . Since $h_1 \notin t_5$ it must be that either $h_1 \in t_2$, $h_1 \in t_3$, or $h_1 \in t_4$. Assume without loss of generality that $h_1 \in t_2$ and that $t_2 = \{h_1, h_{i_3}, \alpha_{j_2}\}$ where $1 \leq i_3 \leq 11$ and $\alpha_{j_2} \in N \setminus H$. Since $v_{h_1}(\alpha_{j_2}) = 0$ it follows that $u_{h_1}(M) = v_{h_1}(h_{i_3})$. By the design of H , it must be that $v_{h_1}(h_{i_3}) = 2$ so $u_{h_1}(M) = 2$. Now h_1 has j-envy for α_{j_1} , since $u_{h_1}(M) = 2 < 4 = u_{h_1}(\{h_3, h_{11}\})$, $v_{h_3}(\alpha_{j_1}) = 0 < 2 = v_{h_3}(h_1)$, and $v_{h_{11}}(\alpha_{j_1}) = 0 < 2 = v_{h_{11}}(h_1)$. This is a contradiction.

- Suppose $\{i_1, i_2\} = \{2, 7\}$. Consider h_1 . As before, if $u_{h_1}(M) \leq 3$ then h_1 has j-envy for α_{j_1} , since $u_{h_1}(M) \leq 3 < 4 = u_{h_1}(\{h_2, h_7\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 2 = v_{h_2}(h_1)$, and $v_{h_7}(\alpha_{j_1}) = 0 < 2 = v_{h_7}(h_1)$. It follows that $u_{h_1}(M) \geq 4$. By the design of H , it must be that $M(h_1) = \{h_1, h_{i_3}, h_{i_4}\}$ where $2 \leq i_3, i_4 \leq 11$. It follows that $M(h_1) = t_5$. Now consider h_4 . If $u_{h_4}(M) \leq 6$ then h_4 has j-envy for α_{j_1} , since $u_{h_4}(M) \leq 6 < 7 = u_{h_4}(\{h_2, h_7\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_7}(\alpha_{j_1}) = 0 < 1 = v_{h_7}(h_4)$. It follows that $u_{h_4}(M) \geq 7$ so, similarly, $M(h_4)$ must contain three agents in H and thus $h_4 \in t_5$. Assume without loss of generality that $h_4 = h_{i_3}$ so $t_5 = \{h_1, h_4, h_{i_4}\}$. Since $u_{h_4}(M) \geq 7$ and $v_{h_4}(h_1) = 2$ it must be that $v_{h_4}(h_{i_4}) \geq 5$ and thus that either $i_4 = 3$ or $i_4 = 6$. Consider h_{10} . If $u_{h_{10}}(M) \leq 6$ then h_{10} has j-envy for α_{j_1} , since $u_{h_{10}}(M) \leq 6 < 7 = u_{h_{10}}(\{h_2, h_7\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 6 = v_{h_2}(h_{10})$, and $v_{h_7}(\alpha_{j_1}) = 0 < 1 = v_{h_7}(h_{10})$. It follows that $u_{h_{10}}(M) \geq 7$. Since $h_{10} \notin t_5$ it must be that either $h_{10} \in t_2$, $h_{10} \in t_3$, or $h_{10} \in t_4$. Assume without loss of generality that $h_{10} \in t_2$ and that $t_2 = \{h_{10}, h_{i_5}, \alpha_{j_2}\}$ where $1 \leq i_5 \leq 11$ and $\alpha_{j_2} \in N \setminus H$. Since $v_{h_{10}}(\alpha_{j_2}) = 0$ it follows that $v_{h_{10}}(h_{i_5}) = u_{h_{10}}(M) \geq 7$, which is a contradiction.
- Suppose $v_{h_{i_1}}(h_{i_2}) = 2$. It follows that either $i_1 = 1$ or $i_2 = 1$. Assume without loss of generality that $i_1 = 1$. Note that $2 \leq i_2 \leq 11$. Consider h_{i_2} . Note that since $v_{h_{i_2}}(\alpha_{j_1}) = 0$ it must be that $u_{h_{i_2}}(M) = v_{h_{i_2}}(h_1) = 2$. By the design of H , for each possible assignment of i_2 , namely $2 \leq i_2 \leq 11$, there exist five agents $h_{i_3}, h_{i_4}, h_{i_5}, h_{i_6}, h_{i_7}$ such that $v_{h_{i_2}}(h_{i_k}) > 2$ for $3 \leq k \leq 7$. A counting argument shows that at least one of these five agents does not belong to t_5 and hence must belong to either t_2 , t_3 , or t_4 . Assume without loss of generality that $h_{i_3} \in t_2$ and $t_2 = \{h_{i_3}, h_{i_8}, \alpha_{j_2}\}$ where $2 \leq i_8 \leq 11$ and $\alpha_{j_2} \in N \setminus H$. Recall that $v_{h_{i_2}}(h_{i_3}) > 2$. By the design of H it follows that $u_{h_{i_2}}(\{h_{i_3}, h_{i_8}\}) > 3$. Now h_{i_2} has j-envy for α_{j_2} since $u_{h_{i_2}}(M) = 2 < 3 < u_{h_{i_2}}(\{h_{i_3}, h_{i_8}\})$, $v_{h_{i_3}}(\alpha_{j_2}) = 0 < 1 \leq v_{h_{i_3}}(h_{i_2})$, and $v_{h_{i_8}}(\alpha_{j_2}) = 0 < 1 \leq v_{h_{i_8}}(h_{i_2})$.
- Suppose $v_{h_{i_1}}(h_{i_2}) = 1$. Without loss of generality we assume that either $\{i_1, i_2\} = \{2, 5\}$ or $\{i_1, i_2\} = \{2, 6\}$.
 - Suppose $\{i_1, i_2\} = \{2, 5\}$. If $u_{h_4}(M) \leq 9$ then h_4 has j-envy for α_{j_1} , since $u_{h_4}(M) \leq 9 < 10 = u_{h_4}(\{h_2, h_5\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_5}(\alpha_{j_1}) = 0 < 4 = v_{h_5}(h_4)$. It follows that $u_{h_4}(M) \geq 10$. The only possibility is that $M(h_4) = \{h_3, h_4, h_6\}$ and hence $M(h_4) = t_5$. Consider h_1 . If $u_{h_1}(M) \leq 3$ then h_1 has j-envy for α_{j_1} , since $u_{h_1}(M) \leq 3 < 4 = u_{h_1}(\{h_2, h_5\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 2 = v_{h_2}(h_1)$, and $v_{h_5}(\alpha_{j_1}) = 0 < 2 = v_{h_5}(h_1)$. It follows that $u_{h_1}(M) \geq 4$. By the design of H , it follows that $M(h_1)$ must contain three agents in H , which is a contradiction since $h_1 \notin t_5$.
 - Suppose $\{i_1, i_2\} = \{2, 6\}$. It follows that $u_{h_4}(M) \leq 9$ so h_4 has j-envy for α_{j_1}

since $u_{h_4}(M) \leq 9 < 12 = u_{h_4}(\{h_2, h_6\})$, $v_{h_2}(\alpha_{j_1}) = 0 < 6 = v_{h_2}(h_4)$, and $v_{h_6}(\alpha_{j_1}) = 0 < 6 = v_{h_6}(h_4)$.

□

Lemma 7.48. *If (N, V) contains a j -envy-free matching M then H has an open configuration in M .*

Proof. By definition, $4 \leq \sigma(H, M) \leq 11$. If $\sigma(H, M) \leq 5$ then H has an open configuration in M , by Lemmas 7.46 and 7.47. If $6 \leq \sigma(H, M) \leq 11$ then, by a counting argument, at least one triple in M must contain exactly one agent in H . In other words, H has an open configuration in M . □

We have shown, in Lemma 7.48, that if (N, V) contains a j -envy-free matching M then H has an open configuration in M . By definition, some triple t_β in M contains exactly one agent in H . Since $|H| = 11$, if t_β is the only triple in M to contain exactly one agent in H then there must exist some triple in M that contains exactly two agents in H . By Lemma 7.43, this is a contradiction. It follows that at least two triples in M exist that each contain exactly one agent in H . Suppose $t_\beta, t_\gamma \in M$ are two such triples and $t_\beta = \{h_{a_1}, \alpha_{b_1}, \alpha_{b_2}\}$ and $t_\gamma = \{h_{a_2}, \alpha_{b_3}, \alpha_{b_4}\}$.

Lemma 7.49. *If (N, V) contains a j -envy-free matching then $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} = L$.*

Proof. Suppose for a contradiction that $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} \neq L$.

By definition, $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} \cap H = \emptyset$ and $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} \neq L$ it must be that at least one agent in $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\}$ belongs to C . Assume without loss of generality that $\alpha_{b_1} \in C$.

We have already shown that t_β contains exactly one agent in H . Since $v_{\alpha_{b_1}}(h_{a_1}) = 0$, by the design of the instance it must be that $u_{\alpha_{b_1}}(M) = v_{\alpha_{b_1}}(\alpha_{b_2}) \leq 3$. By the design of the instance $v_{\alpha_{b_1}}(\alpha_{b_3}) \geq 2$ and $v_{\alpha_{b_1}}(\alpha_{b_4}) \geq 2$ so $u_{\alpha_{b_1}}(\{\alpha_{b_3}, \alpha_{b_4}\}) \geq 4$. Now α_{b_1} has j -envy for h_{a_2} since $u_{\alpha_{b_1}}(M) \leq 3 < 4 \leq u_{\alpha_{b_1}}(\{\alpha_{b_3}, \alpha_{b_4}\})$, $v_{\alpha_{b_3}}(h_{a_2}) = 0 < 2 \leq v_{\alpha_{b_3}}(\alpha_{b_1})$, and $v_{\alpha_{b_4}}(h_{a_2}) = 0 < 2 \leq v_{\alpha_{b_4}}(\alpha_{b_1})$. □

Lemma 7.50. *If (N, V) contains a j -envy-free matching then $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_2\}, \{l_3, l_4\}\}$.*

Proof. By Lemma 7.49, $\{\alpha_{b_1}, \alpha_{b_2}, \alpha_{b_3}, \alpha_{b_4}\} = L$. There are now three possibilities: first that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_3\}, \{l_2, l_4\}\}$, second that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_4\}, \{l_2, l_3\}\}$, and third that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_2\}, \{l_3, l_4\}\}$.

First suppose $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_3\}, \{l_2, l_4\}\}$. Now l_1 has j -envy for h_{a_2} since $u_{l_1}(\{h_{a_1}, l_3\}) = 1 < 3 \leq u_{l_1}(\{l_2, l_4\})$, $v_{l_2}(h_{a_2}) = 0 < 2 = v_{l_2}(l_1)$, and $v_{l_4}(h_{a_2}) = 0 < 1 \leq v_{l_4}(l_1)$.

Second suppose $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_4\}, \{l_2, l_3\}\}$. As before, l_1 has j-envy for h_{a_2} since $u_{l_1}(\{h_{a_1}, l_4\}) = 1 < 3 \leq u_{l_1}(\{l_2, l_3\})$, $v_{l_2}(h_{a_2}) = 0 < 2 = v_{l_2}(l_1)$, and $v_{l_3}(h_{a_2}) = 0 < 1 \leq v_{l_3}(l_1)$.

It remains that $\{\{\alpha_{b_1}, \alpha_{b_2}\}, \{\alpha_{b_3}, \alpha_{b_4}\}\} = \{\{l_1, l_2\}, \{l_3, l_4\}\}$. \square

By Lemma 7.50, either $\{\alpha_{b_1}, \alpha_{b_2}\} = \{l_1, l_2\}$ or $\{\alpha_{b_1}, \alpha_{b_2}\} = \{l_3, l_4\}$. Without loss of generality assume that $\{\alpha_{b_1}, \alpha_{b_2}\} = \{l_1, l_2\}$.

Lemma 7.51. *If (N, V) contains a j-envy-free matching then $u_{c_i}(M) = 6$ for each i where $1 \leq i \leq 3q$.*

Proof. Suppose to the contrary that some $1 \leq i \leq 3q$ exists where $u_{c_i}(M) < 6$. Then c_i has j-envy for h_{a_1} since $u_{c_i}(M) \leq 5 < 6 \leq u_{c_i}(\{l_1, l_2\})$, $v_{l_1}(h_{a_1}) = 0 < 3 = v_{l_1}(c_i)$, and $v_{l_2}(h_{a_1}) = 0 < 3 = v_{l_2}(c_i)$. This contradicts our supposition that M is j-envy-free. \square

Lemma 7.52. *If (N, V) contains a j-envy-free matching then G contains a partition into triangles.*

Proof. Suppose (N, V) contains a j-envy-free partition into triangles M . Lemma 7.51 shows that $u_{c_i}(M) = 6$ for each i where $1 \leq i \leq 3q$. By construction, it follows that $M(c_i)$ contains two agents c_j, c_k such that $v_{c_i}(c_j) = v_{c_i}(c_k) = 3$. By construction, c_j and c_k therefore correspond to vertices $w_j, w_k \in W$ where $\{w_i, w_j\} \in E$ and $\{w_i, w_k\} \in E$. It follows thus that there are exactly q triples in M each containing three agents $\{c_i, c_j, c_k\}$, where the three corresponding vertices w_i, w_j, w_k are pairwise adjacent in G . From these triples a partition into triangles X can be easily constructed. \square

We have now shown that the 3DR-AS instance (N, V) contains a j-envy-free matching if and only if the PIT instance G contains a partition into triangles. This shows that the reduction is correct.

Theorem 7.7. *Deciding if a given instance of 3DR-AS contains a j-envy-free matching is NP-complete, even when preferences are symmetric and the maximum possible valuation is 6.*

Proof. It is straightforward to show that this decision problem belongs to NP, since for any two agents $\alpha_i, \alpha_j \in N$ we can test if α_i j-envies α_j in constant time.

We have presented a polynomial-time reduction from Partition Into Triangles (PIT, Problem 5.1), which is NP-complete [43]. Given a graph G , the reduction constructs an instance (N, V) of 3DR-AS with symmetric preferences in which the maximum valuation is 6. Lemmas 7.42 and 7.52 show that (N, V) contains a j-envy-free matching if and only if G contains a partition into triangles and thus that this decision problem is NP-hard. \square

7.5 Summary and open problems

In this chapter we considered the existence of envy-free, wj -envy-free, and j -envy-free matchings in 3DR-AS and the complexity of the associated decision and construction problems. For each of the three solution concepts, we considered various restrictions on the agents' valuations.

We first showed that an arbitrary instance of 3DR-AS may not contain an envy-free matching, even when preferences are binary and symmetric and the maximum degree of the underlying graph is 2. We described a polynomial-time algorithm for this case that can, in a given instance of 3DR-AS, either construct an envy-free matching or report that no such matching exists. We then contrasted this result by showing that the corresponding existence problem is NP-complete even when the maximum degree of the underlying graph is 3.

Next, we considered wj -envy-freeness. Our results for wj -envy-freeness were similar to those for envy-freeness. We first showed that, as in the case of envy-freeness, a wj -envy-free matching may not exist even when preferences are binary and symmetric and the maximum degree of the underlying graph is 2. We described a slightly more complex polynomial-time algorithm for this case, compared to the corresponding algorithm for envy-freeness, which either constructs a wj -envy-free matching or reports that no such matching exists. We also showed that the corresponding existence problem is NP-complete even when the maximum degree of the underlying graph is 3.

We then considered j -envy-freeness. We showed that if preferences are binary but not necessarily symmetric, a j -envy-free matching must exist and can be found in polynomial time. We then considered two restrictions of 3DR-AS, in which valuations are ternary but not symmetric, and non-binary and symmetric. In both restrictions, we showed that a given instance of 3DR-AS may not contain a j -envy-free matching and the associated existence problem is NP-complete.

We summarise our new existence and complexity results in Table 7.1, which also includes the corresponding results for stability from Chapter 6. In the table, for a given solution concept and preference restriction, “must exist?” refers to whether an arbitrary such instance of 3DR-AS must contain a matching that satisfies that solution concept, and “search” refers to the complexity class of the associated construction problem. From this table (and the associated theorems) we can identify a general trend in our results that for successively weaker solution concepts, existence and polynomial-time solvability hold under successively weaker restrictions on the agents' preferences.

We now present some open problems specifically involving envy-freeness, wj -envy-freeness, and j -envy-freeness in 3DR-AS. More general problems, involving solution concepts that do not involve envy and other models of fixed-size coalitions, are discussed in Chapter 9.

input settings		results		
solution concept	preference restriction	must exist?	search	Theorem
stability	binary and symmetric	✓	P	6.1
”	binary	✗	NP-h.	6.4
”	ternary and symmetric	✗	NP-h.	6.5
envy	binary and symmetric, $\Delta = 2$	✗	P	7.1
”	binary and symmetric, $\Delta = 3$	✗	NP-h.	7.2
weakly justified envy	binary and symmetric, $\Delta = 2$	✗	P	7.3
”	binary and symmetric, $\Delta = 3$	✗	NP-h.	7.4
justified envy	binary and symmetric	✓	P	Obs. 7.1
”	binary	✓	P	7.5
”	ternary	✗	NP-h.	7.6
”	symmetric and $0 \leq v_i(j) \leq 6$	✗	NP-h.	7.7

Table 7.1: Our complexity results for 3DR-AS (from Chapters 6 and 7). In restrictions involving binary and symmetric preferences, Δ refers to the maximum degree of the underlying graph.

The immediate open problem relates to j -envy-freeness and preferences that are ternary and symmetric. Specifically, it would be interesting to resolve the computational complexity of the problem of deciding if a given instance 3DR-AS with ternary preferences contains j -envy-free matching. The first step in this direction would be to determine whether every instance of 3DR-AS with ternary preferences contains a j -envy-free matching.

In Theorem 7.7 we showed that there exist instances of 3DR-AS with symmetric preferences that do not contain a j -envy-free matching. The proof involved a gadget H from which such an instance can be directly derived, by adding a single “isolated” agent. This specific instance was discovered by sequential search, using an integer programming [86] model to test candidate solutions (similar techniques have been used in the context of hedonic games [90]). In order to reduce the search space, certain assumptions were made about the design of the instance, such as the existence of an isolated agent (an “undesired guest” [27, 39]). Although this technique was effective, it is hard to provide any intuition as to why this specific instance contains no j -envy-free matching. It is also open whether this instance is minimal, or if a smaller instance of 3DR-AS exists, with fewer than 12 agents, that contains no j -envy-free matching.

As we noted in Chapter 6, another open problem is whether our results apply to a setting involving more general definitions of binary and ternary. For example, whether our results for binary preferences hold in a more general setting in which $v_{\alpha_i}(\alpha_j) \in \{a, b\}$ for any non-negative integers a and b where $a < b$.

As we also noted in Chapter 6, it might be interesting to identify other restrictions of 3DR-AS

in which an envy-free, wj -envy-free or j -envy-free matching can be found in polynomial time. The gadgets used in our reductions are highly regular and it might be that there exist interesting classes of instances that must contain a stable matching. Alternatively, we could study these problems from the perspective of parameterised complexity. For example, in the case of binary and symmetric preferences, one could consider the tree-width [85] of the instance.

It might be also interesting to estimate the probability that a random instance of 3DR-AS contains an envy-free, wj -envy-free, or j -envy-free matching, or to estimate the same probability in a random instance of 3DR-AS with binary or ternary preferences. Our complexity results indicate that, among instances with binary and symmetric preferences and maximum degree 2, the set of instances that contain a j -envy-free matching (i.e. all instances) is larger than the set of instances that contain a wj -envy-free matching, which is in turn larger than the set of instances that contain an envy-free matching. We conjecture that, in a general instance of 3DR-AS, the probability that a given instance contains an envy-free matching is smaller than the probability that it contains a wj -envy-free matching, which is in turn smaller than the probability that it contains a j -envy-free matching. In this direction, it might be possible to apply probabilistic techniques from graph theory, such as the Erdős-Rényi model of a random graph. Of course, the probabilistic events in which agents have envy for other agents are not independent, which complicates the analysis. Alternatively an empirical approach might be informative, for example by formulating the problem as an integer program [86].

Chapter 8

The K_r -packing Problem in bounded degree graphs

8.1 Introduction

8.1.1 Background

In this chapter we consider two problems related to clique packings in undirected graphs. In particular, finding a maximum-cardinality set of r -cliques, for some fixed $r \geq 3$, subject to the cliques in that set being either vertex disjoint or edge disjoint. We refer to such a set as a K_r -packing, and to the two problems as the *Vertex-Disjoint K_r -Packing Problem* (VDK $_r$) and the *Edge-Disjoint K_r -Packing Problem* (EDK $_r$). Note that r is a fixed constant and does not form part of the problem input. If r is not fixed then both problems generalise the well-studied problem of finding a clique of a given size [43].

Most existing research relating to either vertex- or edge-disjoint K_r -packings covers either more restricted or more general cases. Two well-known special cases of K_r -packing are VDK $_2$, also known as two-dimensional matching; and VDK $_3$, for which an associated decision problem is known as *Partition Into Triangles* (PIT, Problem 5.1).

VDK $_2$ is a central problem of graph theory and algorithmics. A classical result is that a maximum-cardinality two-dimensional matching can be found in polynomial time [91, 92]. VDK $_3$ and its associated decision problem have also been the subject of much research. In 1975, Karp [93] noted that PIT, i.e. deciding if a give graph contains a K_3 -packing of cardinality $|V|/3$, was NP-complete. In this chapter we call such a K_r -packing, with cardinality $|V|/r$, *perfect*, although it is sometimes known as a K_r -factor [94].

In 2002, Caprara and Rizzi [9] considered VDK $_3$ and EDK $_3$ in the setting of a fixed maximum degree Δ . They showed that VDK $_3$ is solvable in polynomial time if $\Delta = 3$ and APX-hard

even when $\Delta = 4$, and EDK_3 is solvable in polynomial time if $\Delta = 4$ and APX-hard even when $\Delta = 5$. They also showed that VDK_3 is NP-hard for planar graphs even when $\Delta = 4$ and EDK_3 is NP-hard for planar graphs with $\Delta = 5$. In 2013, van Rooij et al. [95] established an equivalence between VDK_3 when $\Delta = 4$ and *Exact 3-Satisfiability* (X3SAT). They then used this equivalence to devise an $O(1.02220^n)$ -time algorithm for PIT (i.e. the perfect VDK_3 decision problem) when $\Delta = 4$.

The approximability of VDK_3 and EDK_3 has also been of interest. In 1989, Hurkens and Schrijver [96] presented a new result relating to *Systems of Distinct Representatives*, which they used to construct an algorithm, based on “local improvement”, for a class of packing problems that includes VDK_r and EDK_r . Specifically, for either VDK_r or EDK_r and for any fixed constant $\varepsilon > 0$, Hurkens and Schrijver identify a polynomial-time $(r/2 + \varepsilon)$ -approximation algorithm. In 1995, Halldórsson [97] presented an alternative proof of this result and considered algorithms based on local improvement for other types of packing problems. In 2005, Manić and Wakabayashi [98] described approximation algorithms that improve on this approximation ratio for the restricted cases of VDK_3 in which $\Delta = 4$, and EDK_3 in which $\Delta = 5$. They also presented a linear-time algorithm for VDK_3 on so-called indifference graphs.

A generalisation of VDK_r is *Vertex-Disjoint H -Packing* (also called *Vertex-Disjoint G -Packing*), where H is an arbitrary but fixed undirected graph. In 1982, Takamizawa et al. [99] showed that an optimisation problem related to Vertex-Disjoint H -Packing is solvable in polynomial time on a subclass of planar graphs known as *series-parallel* graphs. In 1983, Kirkpatrick and Hell [82] reviewed the literature of Vertex-Disjoint H -Packing and classify the complexity of an array of packing problems, some of which generalise VDK_r . In particular, they showed that if H contain any connected component with three vertices then the perfect vertex-disjoint H -packing decision problem is NP-complete. Pantel’s 1999 thesis provides a comprehensive survey of H -packing [100].

The restriction of H -packing to K_r -packing has received comparatively less attention in the literature. In 1998, Dahlhaus and Karpinski [101] proved that a perfect vertex-disjoint K_r -packing can be found in polynomial time, if it exists, in chordal and strongly chordal graphs. In 2001, Guruswami et al. [94] showed that, for any $r \geq 3$, the VDK_r decision problem is NP-complete for chordal graphs, planar graphs (assuming $r < 5$), line graphs, and total graphs. Their NP-completeness result involving K_r -packing on chordal graphs, for any $r \geq 3$, resolved an open question of Dahlhaus and Karpinski. Guruswami et al. also described polynomial-time algorithms for VDK_3 and the perfect VDK_r decision problem on split graphs, and VDK_r on cographs. Their result for cographs was later extended by Pedrotti and de Mello [102] for so-called P_4 -sparse graphs.

From the converse perspective of graphs with a fixed minimum degree, Hajnal and Szemerédi

[103] proved in 1970 that if the minimum degree of a graph is greater than or equal to $(1 - 1/r)|V|$ then a perfect vertex-disjoint K_r -packing must exist. Kierstead and Kostochka [104] later generalised this result to show that, in this case, such a packing can be constructed in polynomial time. A growing area of research work studies the existence of perfect K_r - and H -packings with respect to conditions involving vertex degree [105, 106].

8.1.2 Our contribution

In 2002, Caprara and Rizzi [9] showed that VDK_3 is solvable in polynomial time if $\Delta = 3$ and APX-hard if $\Delta = 4$; and EDK_3 is solvable in polynomial time if $\Delta = 4$ and APX-hard if $\Delta = 5$. In this chapter we extend some of their techniques in order to generalise their results and fully classify the complexity of both VDK_r and EDK_r for any $\Delta \geq 1$ and any fixed $r \geq 3$. We summarise this classification in Table 8.1.

		is solvable in	
	linear time if	polynomial time if	is APX-hard if
VDK_r	$\Delta < 3r/2 - 1$	$\Delta < 5r/3 - 1$	$\Delta \geq \lceil 5r/3 \rceil - 1$
EDK_r	$\Delta < 3r/2 - 1$	$\begin{cases} \Delta \leq 2r - 2 \text{ if } r \leq 5 \\ \Delta < 5r/3 - 1 \text{ otherwise} \end{cases}$	$\begin{cases} \Delta > 2r - 2 \text{ if } r \leq 5 \\ \Delta \geq \lceil 5r/3 \rceil - 1 \text{ otherwise} \end{cases}$

Table 8.1: Our complexity results for VDK_r and EDK_r

In the next section, Section 8.1.3, we define some additional notation and make an observation on the coincidence of vertex- and edge-disjoint K_r -packings.

In Section 8.2, we consider the case when $\Delta < 3r/2 - 1$. We show that in this case, any maximal vertex- or edge-disjoint K_r -packing is also maximum (Theorem 8.1), and devise a linear-time algorithm for VDK_r and EDK_r in this setting (Theorem 8.2 and Corollary 8.1).

In Section 8.3, we present our solvability results, showing that VDK_r can be solved in polynomial time if $\Delta < 5r/3 - 1$ (Theorem 8.3); and EDK_r can be solved in polynomial time if either $r \leq 5$ and $\Delta \leq 2r - 2$ (Theorem 8.5), or $r \geq 6$ and $\Delta < 5r/3 - 1$ (Theorem 8.4). Our proof uses a similar technique to that of Caprara and Rizzi's [9], which involves finding a maximum independent set in a graph that represents the intersection of K_r s.

In Section 8.4, we show that our solvability results are in a sense best possible, unless $\text{P} \neq \text{NP}$. Specifically, we show that VDK_r is APX-hard if $\Delta \geq \lceil 5r/3 \rceil - 1$ (Theorem 8.6); and EDK_r is APX-hard if either $r \leq 5$ and $\Delta > 2r - 2$ (Theorem 8.8), or $r \geq 6$ and $\Delta \geq \lceil 5r/3 \rceil - 1$ (Theorem 8.7). In other words, we prove that there exists some fixed constants $\varepsilon > 1$ and $\varepsilon' > 1$ such that no polynomial-time ε -approximation algorithm exists for VDK_r if $\Delta \geq \lceil 5r/3 \rceil - 1$; and no polynomial-time ε' -approximation algorithm exists for EDK_r if either $r \leq 5$ and $\Delta > 2r - 2$, or $r \geq 6$ and $\Delta \geq \lceil 5r/3 \rceil - 1$.

In Section 8.5 we recap on our results and consider directions for future work.

8.1.3 Preliminaries

In this section we first clarify our terminology and notation and then make a preliminary observation on the coincidence of vertex- and edge-disjoint K_r -packings.

Let $G = (V, E)$ be a simple undirected graph. We denote the *closed neighbourhood* of some vertex $v \in V$ as $N_G[v] = N_G(v) \cup \{v\}$. We denote by $\deg_G(v) = |N_G[v]| - 1$ the *degree* of v in G and by $\Delta(G) = \max_{v \in V} \deg_G(v)$ the *maximum degree* of G . If the graph in question is clear from context then we just write Δ . If $\deg_G(v) = \delta$ for all $v \in V$ then we say that G is δ -*regular*. For any subset of vertices $U \subseteq V$, we denote by $G[U]$ the subgraph of G induced by U . Let K_r denote a clique of size r , for some integer $r \geq 1$, and K_r^G be the set of K_r s in G . We say that T is a K_r -*packing* in G if $T \subseteq K_r^G$. The *cardinality* of a K_r -packing is the number of K_r s that it contains. We say that a K_r -packing T is *vertex disjoint* if any two K_r s in T have no vertex in common and *edge disjoint* if any two K_r s in T intersect by at most one vertex. The *Vertex-Disjoint K_r -Packing Problem* (VDK_r) is the following optimisation problem: given a simple undirected graph G , find a vertex-disjoint K_r -packing of maximum cardinality. The *Edge-Disjoint K_r -Packing Problem* (EDK_r) is defined analogously.

If G contains four vertices $v_i, v_{j_1}, v_{j_2}, v_{j_3}$ where $\{v_i, v_{j_a}\} \in E$ for each $a \in \{1, 2, 3\}$, $\{v_{j_1}, v_{j_2}\} \notin E$, $\{v_{j_1}, v_{j_3}\} \notin E$, and $\{v_{j_2}, v_{j_3}\} \notin E$, then we say that these four vertices form a *claw*. Otherwise, we say that G is *claw-free*. For example, line graphs are claw-free [107].

For any maximisation problem P , instance I of P , and feasible solution S of I , let $m_P(I, S)$ denote the measure of S . Let $\text{opt}_P(I) = \max_{S \in \mathcal{F}(I)} m_P(I, S)$, where $\mathcal{F}(I)$ is the set of feasible solutions of I .

For technical purposes we define the K_r -*vertex intersection graph* $\mathcal{K}_r^G = (K_r^G, E_{\mathcal{K}_r^G})$ of G , in which $\{U, W\} \in E_{\mathcal{K}_r^G}$ if $|U \cap W| \geq 1$ for any $U, W \in K_r^G$. Similarly, we define the K_r -*edge intersection graph* $\mathcal{K}'_r^G = (K_r^G, E_{\mathcal{K}'_r^G})$ of G in which $\{U, W\} \in E_{\mathcal{K}'_r^G}$ if $|U \cap W| \geq 2$ for any $U, W \in K_r^G$. We now make a preliminary observation.

Observation 8.1. *If $\Delta < 2r - 2$ then any edge-disjoint K_r -packing is also vertex disjoint.*

Proof. Any two K_r s in G that share at least one vertex must in fact share at least two vertices, otherwise that vertex has degree at least $2r - 2$ in G . \square

8.2 Linear-time solvability

In this section we present Algorithm `greedyCliques`, which can solve VDK_r and EDK_r in linear time if $\Delta < 3r/2 - 1$. This algorithm generalises an algorithm of van Rooij et al. [95]

that can solve VDK_3 in linear time.

The key insight is that if $\Delta < 3r/2 - 1$ then any maximal vertex-disjoint K_r -packing is also a maximum vertex-disjoint K_r -packing. The proof of this is stated in Theorem 8.1, which we prove using a sequence of lemmas. In what follows, suppose G is a simple undirected graph in which $\Delta(G) < 3r/2 - 1$.

Lemma 8.1. *For any $U_1, U_2 \in K_r^G$, if $\{U_1, U_2\} \in E_{\mathcal{K}_r^G}$ then $|U_1 \cap U_2| > r/2$.*

Proof. Consider any $U_1, U_2 \in K_r^G$ where $\{U_1, U_2\} \in E_{\mathcal{K}_r^G}$. By definition of \mathcal{K}_r^G , there exists at least one $u \in V$ where $u \in U_1 \cap U_2$. Since $3r/2 - 1 > \Delta \geq \deg_G(u) \geq |U_1 \cup U_2| - 1 = |U_1| + |U_2| - |U_1 \cap U_2| - 1 = 2r - |U_1 \cap U_2| - 1$ it follows that $|U_1 \cap U_2| > 2r - 3r/2 = r/2$. \square

Lemma 8.2. *\mathcal{K}_r^G is a disjoint union of cliques (i.e. a cluster graph [108]).*

Proof. It suffices to show that for any three vertices $U_i, U_j, U_k \in K_r^G$, if $\{U_i, U_j\} \in E_{\mathcal{K}_r^G}$ and $\{U_j, U_k\} \in E_{\mathcal{K}_r^G}$ then $\{U_i, U_k\} \in E_{\mathcal{K}_r^G}$. Consider some such U_i, U_j, U_k . If $\{U_i, U_j\} \in E_{\mathcal{K}_r^G}$ and $\{U_j, U_k\} \in E_{\mathcal{K}_r^G}$ then by Lemma 8.1 it must be that $|U_i \cap U_j| > r/2$ and $|U_j \cap U_k| > r/2$. Since $|U_j| = r$ it follows that $|U_i \cap U_k| > 0$ and thus that $\{U_i, U_k\} \in E_{\mathcal{K}_r^G}$. \square

Theorem 8.1. *If T is a maximal vertex-disjoint K_r -packing then T is a maximum vertex-disjoint K_r -packing.*

Proof. Suppose T is a maximal vertex-disjoint K_r -packing in G , which by definition corresponds to a maximal independent set in \mathcal{K}_r^G . Since \mathcal{K}_r^G is the disjoint union of cliques (by Lemma 8.2), any two maximal independent sets in \mathcal{K}_r^G have the same cardinality so T is also maximum. \square

We have shown in Theorem 8.1 that any maximal vertex-disjoint K_r -packing is also a maximum vertex-disjoint K_r -packing. It follows immediately that VDK_r can be solved in $O(|V|')$ time by constructing the K_r -vertex intersection graph \mathcal{K}_r^G and greedily selecting an independent set. In fact, the explicit construction of \mathcal{K}_r^G can be avoided by exploring G and greedily selecting K_r s. We present Algorithm `greedyCliques`, shown in Algorithm 8.1, and show that it requires $O(|V|)$ time.

Lemma 8.3. *Algorithm `greedyCliques` requires $O(|V|)$ time.*

Proof. In any iteration of the outermost while loop, either v and its incident edges are removed from G or a set of vertices K where $v \in K$ and incident edges are removed. It follows that the algorithm terminates after at most $|V|$ iterations of this loop. It remains to show that one iteration of the loop can be performed in constant time.

Algorithm 8.1 Algorithm greedyCliques**Input:** a fixed $r \geq 3$, a simple undirected graph $G = (V, E)$ where $\Delta(G) < 3r/2 - 1$ **Output:** a maximum K_r -packing T

```

 $T \leftarrow \emptyset$ 
while  $|V| > 0$  do
   $v \leftarrow$  any element of  $V$ 
  if  $\deg_G(v) \geq r - 1$  then
     $K \leftarrow \emptyset$ 
    for each subset  $W$  of  $N_G(v)$  of size  $r - 1$  do
      if  $G[W]$  has  $\binom{r-1}{2}$  edges then
         $\triangleright$  so  $W$  is a clique of size  $r - 1$  in  $G$ 
         $K \leftarrow W \cup \{v\}$ 
      end if
    end for
    if  $K \neq \emptyset$  then
       $G \leftarrow G[V \setminus K]$ 
       $T \leftarrow T \cup \{K\}$ 
    else
       $G \leftarrow G[V \setminus \{v\}]$ 
    end if
  else
     $G \leftarrow G[V \setminus \{v\}]$ 
  end if
end while
return  $T$ 

```

In each iteration, either $\deg_G(v) \geq r - 1$ or $\deg_G(v) < r - 1$. Computing $\deg_G(v)$ requires $O(r)$ time, since $\Delta < 3r/2 - 1$. Consider the first branch of the outermost if statement. There are $\binom{\Delta}{r-1} < \binom{3r/2-1}{r-1} = O(2^r)$ iterations of the for loop. In each iteration, the algorithm tests if $G[W]$ contains $\binom{r-1}{2}$ edges. This can be performed in $O(r^2)$ time. Removing K from G and adding K to T , if $K \neq \emptyset$, can be done in $O(r^2)$ time. In both the else branch in which $K = \emptyset$ and the second branch of the outermost if statement, v can be removed from G in $O(r)$ time. \square

Theorem 8.2. *If $\Delta < 3r/2 - 1$ then VDK_r can be solved in linear time.*

Proof. By Lemma 8.3, Algorithm greedyCliques terminates in $O(2^r|V|)$ time. By Theorem 8.1, it suffices to show that it returns a maximal vertex-disjoint K_r -packing T in G . Suppose K' is an arbitrary K_r in G . We show that either K' is added to T or at least one vertex in K' belongs to some other K_r in T . By the pseudocode, the algorithm removes at least one vertex in each iteration of the while loop, which ends once there are no remaining vertices. Consider the first iteration of the while loop in which a vertex v in K' is removed. Let G' be the subgraph of G at the beginning of this iteration. It must be that every vertex of

K' is present in G' , including v . Moreover, it must be that $\deg_{G'}(v) \geq r - 1$. It follows that v was not deleted from G' by the second branch of the outermost if statement. Similarly, v cannot have been deleted from G' by the second branch of the innermost if statement, since v belongs to K' , which is a clique of size r in G' . The only possibility is thus that v was deleted from G' as a result of v being part of a clique K'' of size r in G' , where K'' was then added to T . \square

Corollary 8.1. *If $\Delta < 3r/2 - 1$ then Algorithm greedyCliques can solve EDK_r in linear time.*

Proof. Since $\Delta < 3r/2 - 1 < 2r - 2$, by Observation 8.1 any edge-disjoint K_r -packing is also vertex-disjoint. It follows that any vertex-disjoint K_r -packing by Algorithm greedyCliques is also a maximum edge-disjoint K_r -packing. \square

8.3 Polynomial-time solvability

8.3.1 Vertex-disjoint packing

In this section we show that VDK_r is solvable in polynomial time if $\Delta < 5r/3 - 1$. The proof involves finding an independent set in the K_r -vertex intersection graph \mathcal{K}_r^G . We build on the technique of Caprara and Rizzi [9] and show that if $\Delta(G) < 5r/3 - 1$ then \mathcal{K}_r^G is claw-free. It follows that a maximum independent set in \mathcal{K}_r^G can be found in polynomial time (a result of Minty [107] and Sbihi [109]), which corresponds directly to a maximum vertex-disjoint K_r -packing.

In his paper on claw-free graphs, Minty [107] remarked that an algorithm to find a maximum cardinality matching, or vertex-disjoint K_2 -packing, can be used to find a maximum independent set in a line graph (the K_2 vertex-intersection graph). Here, like Caprara and Rizzi [9], we make use of the converse relationship, that if the corresponding intersection graph is claw-free then VDK_r and EDK_r can be solved in polynomial time. A general relationship between packing problems and independent sets in intersection graphs is also noted by Kann [110].

In what follows, suppose $\Delta(G) < 5r/3 - 1$. In Lemma 8.4 we place a lower bound on the size of the intersection of two non-vertex intersecting K_r s in G .

Lemma 8.4. *$|U_i \cap U_j| > r/3$ for any $\{U_i, U_j\} \in E_{\mathcal{K}_r^G}$.*

Proof. Consider an arbitrary vertex $u_l \in |U_i \cap U_j|$. By definition, u_l is adjacent to each vertex in $U_i \cup U_j$. By the principle of inclusion-exclusion, $|U_i \cup U_j| = 2r - |U_i \cap U_j|$ so $\deg_G(u_l) \geq 2r - |U_i \cap U_j| - 1$. Since $\deg_G(u_l) \leq \Delta(G) < 5r/3 - 1$, it must be that $2r - |U_i \cap U_j| - 1 < 5r/3 - 1$. Rearranging gives $|U_i \cap U_j| > r/3$. \square

Lemma 8.5. \mathcal{K}_r^G is claw-free.

Proof. Consider some $U_i, U_{j_1}, U_{j_2}, U_{j_3} \in K_r^G$ exist where $\{U_i, U_{j_a}\} \in E_{\mathcal{K}_r^G}$ for each $a \in \{1, 2, 3\}$. By Lemma 8.4, it must be that $|U_i \cap U_{j_1}| > r/3$, $|U_i \cap U_{j_2}| > r/3$, and $|U_i \cap U_{j_3}| > r/3$. Since $|U_i| = r$ it follows by the pigeonhole principle that either $U_{j_1} \cap U_{j_2} \neq \emptyset$, $U_{j_2} \cap U_{j_3} \neq \emptyset$, or $U_{j_1} \cap U_{j_3} \neq \emptyset$ and thus $\{U_i, U_{j_1}, U_{j_2}, U_{j_3}\}$ is not a claw in \mathcal{K}_r^G . \square

Theorem 8.3. If $\Delta(G) < 5r/3 - 1$ then VDK_r can be solved in polynomial time.

Proof. First, construct the K_r -vertex intersection graph $\mathcal{K}_r^G = (K_r^G, E_{\mathcal{K}_r^G})$. The set K_r^G can be constructed in $O(\binom{|V|}{r}) = O(|V|^r)$ by considering every possible set of r vertices in V . The set $E_{\mathcal{K}_r^G}$ can then be constructed in $O(|V|^{2r})$ time. Next, find a maximum independent set in \mathcal{K}_r^G , which can be accomplished using Minty's algorithm (other algorithms have since been developed with improved worst-case time complexity [111]). \square

We remark that Minty's algorithm also extends to the problem in which the vertices are weighted and the goal is to find an independent set of maximum total weight [112]. It might be interesting to study a weighted generalisation of VDK_r , in which a weight function exists on either the vertices or edges of the graph, and see if this approach can be used to derive other polynomial-time solvability results.

8.3.2 Edge-disjoint packing

In this section we consider EDK_r . Using Theorem 8.3 and Observation 8.1 it is straightforward to show by that if $r \geq 4$ then EDK_r can be solved in polynomial time. We state this result as Theorem 8.4.

Theorem 8.4. If $r \geq 4$ and $\Delta(G) < 5r/3 - 1$ then EDK_r can be solved in polynomial time.

Proof. If $r > 3$ and $\Delta(G) < 5r/3 - 1$ then we can find a maximum vertex-disjoint K_r -packing in polynomial time by Theorem 8.3. Such a packing is also a maximum edge-disjoint K_r -packing, by Observation 8.1. \square

We now show that this upper bound on $\Delta(G)$ can be improved if $r \in \{4, 5\}$. The key insight in this case is that if $r \in \{4, 5\}$ and $\Delta \leq 2r - 2$ then the K_r -edge intersection graph \mathcal{K}'_r^G is claw-free.

Lemma 8.6. If $r \in \{4, 5\}$ and $\Delta(G) \leq 2r - 2$ then the K_r -edge intersection graph \mathcal{K}'_r^G is claw-free.

Proof. Suppose $U_i, U_{j_1}, U_{j_2}, U_{j_3} \in K_r^G$ exist where $\{U_i, U_{j_a}\} \in E_{\mathcal{K}_r^G}$ for each $a \in \{1, 2, 3\}$. Suppose for a contradiction that $U_i, U_{j_1}, U_{j_2}, U_{j_3}$ is a claw in \mathcal{K}_r^G , i.e. $|U_{j_a} \cap U_{j_b}| < 2$ for each $a, b \in \{1, 2, 3\}$. By the definition of $E_{\mathcal{K}_r^G}$, it follows that $|U_i \cap U_{j_a}| \geq 2$ for each $a \in \{1, 2, 3\}$. Since $|U_i| = r \leq 5$, we assume without loss of generality that $|U_i \cap U_{j_1} \cap U_{j_2}| \geq 1$. Furthermore, it must be that $|U_i \cap U_{j_1} \cap U_{j_2}| = 1$, otherwise $|U_{j_1} \cap U_{j_2}| > 1$ which is a contradiction. Let v_r be the single vertex in $U_i \cap U_{j_1} \cap U_{j_2}$. Since U_{j_1} and U_{j_2} are K_r s in G and $\deg_G(v_r) \leq 2r - 2$ it must be that $N_G[v_r] = U_{j_1} \cup U_{j_2}$. Since U_i is also a K_r it follows that $U_i \subset U_{j_1} \cup U_{j_2}$.

Now consider $U_i \cap U_{j_1}$ and $U_i \cap U_{j_2}$. If $|U_i \cap U_{j_1}| + |U_i \cap U_{j_2}| \geq r + 2$ then since $|U_i| = r$ it follows that $|U_{j_1} \cap U_{j_2}| \geq 2$ which is a contradiction. It follows that $|U_i \cap U_{j_1}| + |U_i \cap U_{j_2}| \leq r + 1$ and either $|U_i \cap U_{j_1}| \leq (r + 1)/2$ or $|U_i \cap U_{j_2}| \leq (r + 1)/2$. Assume without loss of generality that $|U_i \cap U_{j_1}| \leq (r + 1)/2$.

Now consider U_{j_3} . Since $U_i, U_{j_1}, U_{j_2}, U_{j_3}$ is a claw in \mathcal{K}_r^G , it must be that $|U_{j_3} \cap U_i| \geq 2$, $|U_{j_3} \cap U_{j_1}| \leq 1$, and $|U_{j_3} \cap U_{j_2}| \leq 1$. Since $U_i \subset U_{j_1} \cup U_{j_2}$, the only possibility is that $|U_{j_3} \cap U_i| = 2$, $|U_{j_3} \cap U_i \cap U_{j_2}| = 1$ and $|U_{j_3} \cap U_i \cap U_{j_1}| = 1$.

Let v_s be the single vertex in $U_i \cap U_{j_1} \cap U_{j_3}$. Since U_i, U_{j_1} , and U_{j_3} are K_r s in G it follows that v_s is adjacent to every other vertex in $U_{j_3} \cup U_i \cup U_{j_1}$ so

$$\begin{aligned} \deg_G(v_s) &\geq |U_i \cup U_{j_1} \cup U_{j_3}| - 1 \\ &= 3r - |U_i \cap U_{j_1}| - |U_i \cap U_{j_3}| - |U_{j_1} \cap U_{j_3}| + |U_i \cap U_{j_1} \cap U_{j_3}| - 1. \end{aligned}$$

Recall that since $U_i, U_{j_1}, U_{j_2}, U_{j_3}$ is a claw in \mathcal{K}_r^G , $|U_{j_1} \cap U_{j_3}| \leq 1$. We deduced earlier in this proof that $|U_i \cap U_{j_1}| \leq (r + 1)/2$, $|U_i \cap U_{j_3}| = 2$, and $|U_{j_3} \cap U_i \cap U_{j_1}| = 1$. Since $r \geq 4$ it follows that $\deg_G(v_s) \geq (5r - 7)/2 > 2r - 2$, which contradicts the fact that $\Delta(G) \leq 2r - 2$. \square

Theorem 8.5. *If $r \leq 5$ and $\Delta \leq 2r - 2$ then EDK_r can be solved in polynomial time.*

Proof. Caprara and Rizzi [9] prove the case when $r = 3$ and $\Delta \leq 4$. If $r \in \{4, 5\}$ and $\Delta(G) \leq 2r - 2$ then Lemma 8.6 shows that the K_r -edge intersection graph \mathcal{K}_r^G is claw-free. It follows that a maximum edge-disjoint K_r -packing can be found in polynomial time by constructing \mathcal{K}_r^G , in $O(\binom{|V|}{2r})$ time, and finding in it a maximum independent set, which can also be accomplished in time polynomial in the size of \mathcal{K}_r^G [111]. \square

8.4 APX-hardness

8.4.1 Vertex-disjoint packing

We show that if $\Delta \geq \lceil 5r/3 \rceil - 1$ then VDK_r is APX-hard. In other words, there exists some fixed constant $\varepsilon > 0$ such that no polynomial-time ε -approximation algorithm exists for VDK_r , unless $\text{P} = \text{NP}$. To do this, we use an *L-reduction* [113], which is a type of approximability-preserving reduction. An *L-reduction* from an optimisation problem A to another optimisation problem B implies that if B admits a polynomial-time approximation scheme, then so does A .

We reduce from the problem of finding a Maximum Independent Set (MIS) in graphs that are 3-regular and are triangle-free, which we refer to as MIS-3-TF (Problem 8.1).

Problem 8.1. Maximum Independent Set in 3-regular Triangle-Free graphs (MIS-3-TF)

Instance: an undirected graph $G = (V, E)$ that is 3-regular and triangle-free

Solution: a set $S \subseteq V$ such that $\{v_i, v_j\} \notin E$ for any $v_i, v_j \in S$

Measure: $|S|$

Berman and Karpinski [114] show that MIS-3-TF is APX-hard, providing an explicit lower bound on the approximation ratio (specifically, they showed that it is NP-hard to approximate MIS-3-TF within $140/139 - \varepsilon$, for any $\varepsilon > 0$).

The reduction from MIS-3-TF is as follows. Our goal is to construct a new graph $G' = (V', E')$ where each K_r in G' corresponds to exactly one vertex in V and each vertex in V corresponds to exactly one K_r in G' . For any two adjacent vertices in G , the intersection of the two corresponding K_r s in G' will contain exactly $\lfloor r/3 \rfloor$ vertices.

To do this, first construct a set of $|V|$ disjoint K_r s in G' , labelled $\mathcal{U} = \{U_1, U_2, \dots, U_{|V|}\}$ where $U_i = \{u_i^1, u_i^2, \dots, u_i^r\}$. Next, consider each edge $\{v_i, v_j\} \in E$. let $U'_i = \{u_i^{a_1}, u_i^{a_2}, \dots, u_i^{a_{\lfloor r/3 \rfloor}}\}$ be any set of $\lfloor r/3 \rfloor$ vertices in U_i with degree $r - 1$ and $U'_j = \{u_j^{b_1}, u_j^{b_2}, \dots, u_j^{b_{\lfloor r/3 \rfloor}}\}$ be any set of $\lfloor r/3 \rfloor$ vertices in U_j with degree $r - 1$. For each q from 1 to $\lfloor r/3 \rfloor$ inclusive, identify $u_i^{a_q}$ and $u_j^{b_q}$ to create a single vertex labelled $u_{ij}^{a_q}$. Label $U'_j = U'_i$ as W_{ij} .

Finally, for each vertex $v_i \in V$ let X_i be the set of (at least $r \bmod 3$) vertices in U_i with degree $r - 1$. Note that any vertex in G' either belongs to some set W_{ij} where $\{v_i, v_j\} \in E$ or some set X_i where $v_i \in V$.

We first show that the set of K_r s in G' is \mathcal{U} .

Lemma 8.7. $\mathcal{U} = K_r^{G'}$.

Proof. By definition, $\mathcal{U} \subseteq K_r^{G'}$ so it remains to show that each K_r in G' belongs to \mathcal{U} . Suppose K is an arbitrary K_r in G' . By definition, any vertex in any set X_i has degree $r - 1$ in G' and thus belongs to exactly one K_r in G' , namely U_i , which belongs to $K_r^{G'}$. The only other possibility is that each vertex in K belongs to some set W_{ij} where $1 \leq i, j \leq |V|$. Since $|W_{ij}| = \lfloor r/3 \rfloor$ it must be that either there exist three sets $W_{i_1, j_1}, W_{i_2, j_2}, W_{i_3, j_3}$ where $1 \leq i_1, i_2, \dots, j_3 \leq |V|$ and $K \subseteq W_{i_1, j_1} \cup W_{i_2, j_2} \cup W_{i_3, j_3}$, or there exist four sets $W_{i_1, j_1}, W_{i_2, j_2}, W_{i_3, j_3}, W_{i_4, j_4}$ where $1 \leq i_1, i_2, \dots, j_4 \leq |V|$ and $K \subseteq W_{i_1, j_1} \cup W_{i_2, j_2} \cup W_{i_3, j_3} \cup W_{i_4, j_4}$. In the latter case, we may assume without loss of generality that $\{i_1, j_1\} \cap \{i_2, j_2\} = \emptyset$. By the construction of G' it follows that no edge exists between any vertex in W_{i_1, j_1} and any vertex in W_{i_2, j_2} which contradicts the supposition that K is a K_r in G' . It remains that there exist three sets $W_{i_1, j_1}, W_{i_2, j_2}, W_{i_3, j_3}$ where $K \subseteq W_{i_1, j_1} \cup W_{i_2, j_2} \cup W_{i_3, j_3}$ and $1 \leq i_1, i_2, \dots, j_3 \leq |V|$.

By construction, the closed neighbourhood of any vertex in W_{i_1, j_1} is $U_{i_1} \cup U_{j_1}$ so since K is a K_r , without loss of generality assume that $i_1 \in \{i_2, j_2\}$ and $j_1 \in \{i_3, j_3\}$. A symmetric argument shows that $i_2 \in \{i_1, j_1\}$ and $j_2 \in \{i_3, j_3\}$, and $i_3 \in \{i_1, j_1\}$ and $j_3 \in \{i_2, j_2\}$. By symmetry, we need only consider the two cases, in which $i_1 = i_2 = i_3$ and in which $K = W_{i_1, i_2} \cup W_{i_2, i_3} \cup W_{i_3, i_1}$. In the former case, K must be labelled U_{i_1} and thus belongs to $K_r^{G'}$. In the latter case, by the construction of G' the three vertices $\{v_{i_1}, v_{i_2}, v_{i_3}\}$ in G form a triangle, which is a contradiction. \square

Lemma 8.8. $\Delta(G') = \lceil 5r/3 \rceil - 1$.

Proof. By definition, any vertex in X_i has degree $r - 1$. Any vertex in W_{ij} has degree $|U_i| + |U_j| - |W_{ij}| - 1 = 2r - \lfloor r/3 \rfloor - 1$, for any $1 \leq i, j \leq |V|$. \square

Theorem 8.6. For any simple undirected graph G' , if $\Delta(G') \geq \lceil 5r/3 \rceil - 1$ then VDK_r is APX-hard.

Proof. We first show that a vertex-disjoint K_r -packing of size B exists in G' if and only if an independent set of size B exists in G . By Lemma 8.7, any $U_i, U_j \in K_r^{G'}$ that are not vertex disjoint in G' correspond to two vertices $v_i, v_j \in V$, which by the design of the reduction must be adjacent. Conversely, for any two vertices $v_i, v_j \in V$ where $\{v_i, v_j\} \in E$, by the design of the reduction it must be that the two corresponding K_r s, $U_i, U_j \in K_r^{G'}$ are not vertex disjoint in G' . It follows that, for any graph G , $\text{opt}_{\text{MIS-3-TF}}(G) = \text{opt}_{\text{VDK}_r}(G')$. Moreover, for any vertex-disjoint K_r -packing T in G' , there exists a corresponding independent set S in G where $|S| = |T|$, and thus that $\text{m}_{\text{MIS-3-TF}}(G, S) = \text{m}_{\text{VDK}_r}(G', T)$.

It follows that reduction from MIS-3-TF to VDK_r is an L -reduction with $\alpha = \beta = 1$ (also called a *strict reduction* [113]) and thus that VDK_r is APX-hard even when $\Delta(G) = \lceil 5r/3 \rceil - 1$ (shown in Lemma 8.8). To show that VDK_r is APX-hard even when $\Delta(G) \geq \lceil 5r/3 \rceil - 1$, one can add to G' a disconnected star. \square

8.4.2 Edge-disjoint packing

8.4.2.1 Edge-disjoint K_r -packing when $r \geq 6$

If $r \geq 6$ and $\Delta = \lceil 5r/3 \rceil - 1$ then it must be that $\Delta < 2r - 2$, so by Observation 8.1, any edge-disjoint K_r -packing is also vertex disjoint. It follows by Theorem 8.6 that EDK_r is APX-hard for any $r \geq 6$ even when $\Delta = \lceil 5r/3 \rceil - 1$. We generalise this result in Theorem 8.7.

Theorem 8.7. *If $r \geq 6$ and $\Delta \geq \lceil 5r/3 \rceil - 1$ then EDK_r is APX-hard.*

Proof. Suppose $\Delta = \lceil 5r/3 \rceil - 1$. By definition, any vertex-disjoint K_r -packing is also edge disjoint. Since $r \geq 6$ it follows that $\Delta = \lceil 5r/3 \rceil - 1 < 2r - 2$ so by Observation 8.1, any edge-disjoint K_r -packing is also vertex disjoint. We have shown that an edge-disjoint K_r -packing of size B exists in G if and only if a vertex-disjoint K_r -packing of size B exists in G . This fact constitutes an L -reduction with $\alpha = \beta = 1$ from the restricted case of VDK_r in which $\Delta = \lceil 5r/3 \rceil - 1$. The lemma follows by Theorem 8.6. As in the proof of Theorem 8.6, to show that EDK_r is APX-hard if $r \geq 6$ even when $\Delta(G) \geq \lceil 5r/3 \rceil - 1$, one can add to G' a disconnected star. \square

8.4.2.2 Edge-disjoint K_4 -packing

We present an L -reduction from a variant of the *Maximum Satisfiability* problem to EDK_4 when $\Delta = 7$, by extending the L -reduction of Caprara and Rizzi [9] for EDK_3 when $\Delta = 5$.

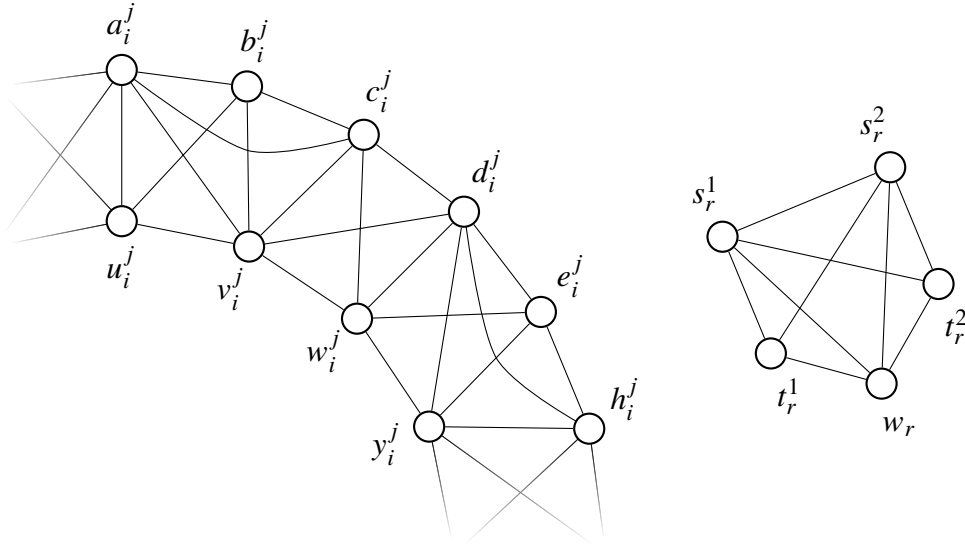
An instance of Maximum Satisfiability is a boolean formula ϕ in conjunctive normal form with *clauses* C and variable set X . Each clause contains a set of *literals*. Each literal is formed by either a variable or its negation. A *truth assignment* \mathfrak{f} is a function $\mathfrak{f} : X \mapsto \{\text{true}, \text{false}\}$. A clause is *satisfied* by \mathfrak{f} if any of its literals are true. The goal is to find a truth assignment that satisfies the maximum number of clauses. We reduce from the special case of Maximum Satisfiability in which each clause contains at most two literals and each variable occurs in at most three clauses. We shall refer to this special case as $\text{Max 2SAT}^{\leq 3}$ (Problem 8.2).

Problem 8.2. $\text{Max 2SAT}^{\leq 3}$

Instance: a boolean formula ϕ in conjunctive normal form, represented as a set of clauses $C = \{c_1, c_2, \dots, c_{|C|}\}$ and a set of variables $X = \{x_1, x_2, \dots, x_{|X|}\}$, in which each clause contains at most two literals and each variable occurs in at most three clauses

Solution: a truth assignment $\mathfrak{f} : X \mapsto \{\text{true}, \text{false}\}$

Measure: the number of clauses in ϕ satisfied by \mathfrak{f}

Figure 8.1: The reduction from $\text{Max } 2\text{SAT}^{\leq 3}$ to EDK_4

Let m_i be the number of occurrences of variable x_i in ϕ for each variable $x_i \in X$. We assume that $2 \leq m_i \leq 3$ for each $x_i \in X$, since if some variable x_i occurs in exactly one clause it can be set to the value satisfying that clause. $\text{Max } 2\text{SAT}^{\leq 3}$ is APX-hard [66].

Given an instance ϕ of $\text{Max } 2\text{SAT}^{\leq 3}$, we construct a graph G such that a truth assignment for ϕ exists that satisfies at least k clauses if and only if there exists an edge-disjoint K_4 -packing of size at least $\sum_{i=1}^{|X|} 3m_i + k$. As in the case of the reduction presented for EDK_3 by Caprara and Rizzi, the reduction here is one of local replacement [43]. As they remark, the construction and connection of variable and clause gadgets is a standard technique when reducing from variants of Maximum Satisfiability. The reduction, shown in Figure 8.1, is as follows.

For each variable x_i , construct a variable gadget of $10m_i$ vertices, labelled $R_i = \{a_i^j, b_i^j, c_i^j, d_i^j, e_i^j, h_i^j, u_i^j, v_i^j, w_i^j, y_i^j\}$ for each j where $1 \leq j \leq m_i$. For each j where $1 \leq j \leq m_i$, add an edge (if it does not exist already) between each pair of vertices in $\{a_i^j, b_i^j, u_i^j, v_i^j\}$; $\{a_i^j, b_i^j, c_i^j, v_i^j\}$; $\{c_i^j, v_i^j, d_i^j, w_i^j\}$; $\{d_i^j, w_i^j, e_i^j, y_i^j\}$; $\{d_i^j, e_i^j, h_i^j, y_i^j\}$; and finally $\{h_i^j, a_i^{j+1}, y_i^j, u_i^{j+1}\}$ if $j < m_i$ and otherwise $\{h_i^j, a_i^1, y_i^j, u_i^1\}$.

We shall refer to $\{a_i^j, b_i^j, u_i^j, v_i^j\}$, $\{c_i^j, v_i^j, d_i^j, w_i^j\}$, and $\{d_i^j, e_i^j, h_i^j, y_i^j\}$ as the *even K_4 s* in R_i , and $\{a_i^j, b_i^j, c_i^j, v_i^j\}$, $\{d_i^j, w_i^j, e_i^j, y_i^j\}$, and $\{h_i^j, a_i^{j+1}, y_i^j, u_i^{j+1}\}$ (and $\{h_i^j, a_i^1, y_i^j, u_i^1\}$) as the *odd K_4 s* in R_i . Note that at this point in construction, $\deg_G(a_i^j) = \deg_G(v_i^j) = \deg_G(d_i^j) = \deg_G(y_i^j) = 6$, $\deg_G(u_i^j) = \deg_G(c_i^j) = \deg_G(w_i^j) = \deg_G(h_i^j) = 5$, and $\deg_G(b_i^j) = \deg_G(e_i^j) = 4$ for each j where $1 \leq j \leq m_i$.

We shall now construct the clause gadgets. For each clause c_r , construct a clause gadget of 5 vertices labelled $S_r = \{s_r^1, t_r^1, s_r^2, t_r^2, w_r\}$. Add an edge (if it does not exist already) between each pair of vertices in $\{s_r^1, t_r^1, s_r^2, w_r\}$ and $\{s_r^1, s_r^2, t_r^2, w_r\}$. We shall refer to $\{s_r^1, t_r^1, s_r^2, w_r\}$ and $\{s_r^1, s_r^2, t_r^2, w_r\}$ as P_i^r and P_j^r supposing the variables of the first and second literals in c_r

are x_i and x_j . Note that at this point in construction, $\deg_G(s_r^1) = \deg_G(s_r^2) = \deg_G(w_r) = 4$ and $\deg_G(t_r^1) = \deg_G(t_r^2) = 3$.

We shall now connect the variable and clause gadgets. For each clause c_r , suppose x_i is the variable of some literal in c_r where c_r contains the j^{th} occurrence of x_i in ϕ . If x_i is the first literal in c_r and occurs positively in c_r then identify b_i^j and s_r^1 , and c_i^j and t_r^1 . We shall hereafter refer to the first identified vertex as either b_i^j or s_r^1 and the second identified vertex as either c_i^j or t_r^1 . Note that now $\deg_G(b_i^j) = \deg_G(c_i^j) = 7$. Similarly, if x_i is the first literal in c_r and occurs negatively in c_r then identify e_i^j and s_r^1 , and h_i^j and t_r^1 . In this case $\deg_G(e_i^j) = \deg_G(h_i^j) = 7$. If x_i is the second literal in c_r and occurs positively in c_r then identify b_i^j and s_r^2 , and c_i^j and t_r^2 . Similarly, if x_i is the second literal in c_r and occurs negatively in c_r then identify e_i^j and s_r^2 , and h_i^j and t_r^2 . This completes the construction of G . Observe that $\Delta = 7$.

It is straightforward that the reduction can be performed in polynomial time. We now prove that the reduction is correct in the first direction. By construction, no K_4 exists in G that contains at least one vertex in a variable gadget and at least one vertex in a clause gadget. Thus, we shall say that some K_4 is *in* a variable or clause gadget if it is a strict subset of that gadget.

Lemma 8.9. *If a truth assignment \mathbf{f} for ϕ satisfies at least k clauses then an edge-disjoint K_4 -packing T exists in G where $|T| \geq \sum_{i=1}^{|X|} 3m_i + k$.*

Proof. Suppose \mathbf{f} is a truth assignment for ϕ that satisfies at least k clauses. We shall construct an edge-disjoint K_4 -packing T where $|T| \geq \sum_{i=1}^{|X|} 3m_i + k$.

For each variable x_i , if $\mathbf{f}(x_i)$ is true then add the set of even K_4 s in R_i to T . Similarly, if $\mathbf{f}(x_i)$ is false then add the set of odd K_4 s in R_i to T . Now $|T| = \sum_{i=1}^{|X|} 3m_i$. For each clause gadget c_r that is satisfied by \mathbf{f} , it must be that there exists some variable x_i where either $\mathbf{f}(x_i)$ is true and x_i occurs positively in c_r or $\mathbf{f}(x_i)$ is false and x_i occurs negatively in c_r . In either case, add P_i^r to T . Now, T contains exactly $\sum_{i=1}^{|X|} 3m_i$ K_4 s in variable gadgets and at least k K_4 s in clause gadgets. It remains to show that T is edge disjoint. By the construction of G , any two K_4 s in T in the same variable gadget are edge disjoint. Consider an arbitrary P_i^r in some clause gadget c_r that belongs to T . It must be that either $\mathbf{f}(x_i)$ is true and x_i occurs positively in c_r or $\mathbf{f}(x_i)$ is false and x_i occurs negatively in c_r . In the former case, T contains the set of even K_4 s in R_i so since $P_i^r \cap R_i = \{b_i^j, c_i^j\}$ where $1 \leq j \leq 3$ it follows that T is edge disjoint. In the latter case, T contains the set of odd K_4 s in R_i so since $P_i^r \cap R_i = \{e_i^j, h_i^j\}$ where $1 \leq j \leq 3$ it also follows that T is edge disjoint. \square

We now prove that the reduction is correct in the second direction. We say that some edge-disjoint K_4 -packing T in G is *canonical* if for any variable gadget R_i , T contains either the

set of even K_4 s in R_i or the set of odd K_4 s in R_i . By the construction of G , no edge-disjoint K_4 -packing can contain all even K_4 s and all odd K_4 s.

We first show that for any variable gadget R_i and edge-disjoint K_4 -packing T , if T neither contains all even K_4 s in R_i nor all odd K_4 s in R_i then the number of K_4 s in T is at most $3m_i - 1$.

Proposition 8.1. *Suppose T is an arbitrary edge-disjoint K_4 -packing in G . For any variable gadget R_i , if T neither contains all even K_4 s in R_i nor all odd K_4 s in R_i then the number of K_4 s in T is at most $3m_i - 1$.*

Proof. By the construction of G , each even K_4 in R_i intersects exactly two odd K_4 s in R_i by at least two vertices and each odd K_4 in R_i intersects exactly two even K_4 s in R_i by at least two vertices.

It follows that the K_4 -edge intersection graph \mathcal{K}_r^G contains a cycle of $6m_i$ vertices corresponding to the $6m_i$ K_4 s in R_i . It then follows that any edge-disjoint K_4 -packing that contains $3m_i$ K_4 s in R_i corresponds to an independent set of size $3m_i$ in \mathcal{K}_r^G , and thus is either the set of even K_4 s in R_i or the set of odd K_4 s in R_i . Since T neither contains all even K_4 s in R_i nor all odd K_4 s in R_i it follows that $|T| < 3m_i$. \square

We can now prove that for any edge-disjoint K_4 -packing in G that is not canonical, there exists a canonical edge-disjoint K_4 -packing in G of at least the same cardinality.

Lemma 8.10. *If T is an edge-disjoint K_4 -packing then there exists a canonical edge-disjoint K_4 -packing T' where $|T'| \geq |T|$.*

Proof. If T is already canonical then let $T' = T$. Otherwise, by the definition of canonical, there must exist at least one variable gadget i such that T neither contains all even K_4 s in R_i nor all odd K_4 s in R_i . For any such i where $1 \leq i \leq |X|$, we show how to modify T to ensure that it either contains the set of even K_4 s in R_i or the set of odd K_4 s in R_i and the cardinality of T does not decrease. It follows that there exists a canonical edge-disjoint K_4 -packing T' where $|T'| \geq |T|$.

By Proposition 8.1, the number of K_4 s in R_i in T is at most $3m_i - 1$.

Suppose the variable x_i corresponding to R_i occurs in clauses $c_{r_1}, c_{r_2}, \dots, c_{r_{m_i}}$, corresponding to the sets $P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}$. It must be that either at most one K_4 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ exists in T where the corresponding occurrence of x_i is positive; or at most one K_4 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ exists in T where the corresponding occurrence of x_i is negative. Suppose the former case is true. Remove the K_4 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ in T where the corresponding occurrence of x_i is positive. Next, remove any even K_4 s in R_i in T and add the set of odd K_4 s in R_i not already in T . The number of K_4 s in R_i in T is now $3m_i$ so since at most one

K_4 was removed, which was not in R_i , it follows that the cardinality of T has not decreased. To see that T is still edge-disjoint, observe that any K_4 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ in T intersects any odd K_4 in R_i by at most one vertex. The construction and proof in the latter case are symmetric. \square

Lemma 8.11. *If T is an edge-disjoint K_4 -packing where $|T| = \sum_{i=1}^{|X|} 3m_i + k$ for some integer $k \geq 1$ then exists a truth assignment \mathbf{f} that satisfies at least k clauses.*

Proof. Assume by Lemma 8.10 that T is canonical. It follows that T contains exactly $\sum_{i=1}^{|X|} 3m_i$ K_4 s in variable gadgets and at least k K_4 s in clause gadgets. For each variable x_i , set $\mathbf{f}(x_i)$ to be true if T contains all even K_4 s in R_i and false otherwise. Now consider each clause gadget c_r where S_r contains some K_4 in T , denoted P_i^r . Suppose x_i occurs positively in c_r . It follows that P_i^r contains b_i^j, c_i^j for some j where $1 \leq j \leq 3$. Since T is canonical and edge-disjoint it follows that T contains the set of even K_4 s in R_i . By the construction of \mathbf{f} it follows that $\mathbf{f}(x_i)$ is true and thus c_r is satisfied. The proof for when x_i occurs negatively in c_r is symmetric. It follows that at least k clauses are satisfied by \mathbf{f} . \square

Lemma 8.12. *If $r = 4$ and $\Delta = 7$ then EDK_r is APX-hard.*

Proof. We shall describe an L -reduction from $\text{Max } 2\text{SAT}^{\leq 3}$ (which is APX-hard [66]) to EDK_4 when $\Delta = 7$, using the definition of Crescenzi [113]. An L -reduction from optimisation problem Q to an optimisation problem P shows that if there exists a $(1 + \delta)$ -approximation algorithm for P then there exists a $(1 + \alpha\beta\delta)$ -approximation algorithm for Q . For compactness we abbreviate $\text{Max } 2\text{SAT}^{\leq 3}$ when appearing in a subscript to M2S3 .

An L -reduction is characterised by a pair (f, g) of functions that can be computed in polynomial time. Here, f is the reduction described at the start of the start of this section (Section 8.4.2.2) in which an instance G of EDK_4 is constructed from an arbitrary instance ϕ of $\text{Max } 2\text{SAT}^{\leq 3}$. It is straightforward to show that f can be computed in polynomial time.

The function g is described by Lemma 8.11. For any instance ϕ of $\text{Max } 2\text{SAT}^{\leq 3}$ and edge-disjoint K_4 -packing in $f(\phi)$, g computes a truth assignment \mathbf{f} for ϕ . It is also straightforward to show that g can be computed in polynomial time.

To show that f, g constitute a valid L -reduction, we must show that there exists fixed constants α, β such that for any instance ϕ of $\text{Max } 2\text{SAT}^{\leq 3}$,

$$\text{opt}_{EDK_4}(f(\phi)) \leq \alpha \text{opt}_{\text{M2S3}}(\phi)$$

and that for any instance ϕ and any edge-disjoint K_4 -packing T in $f(\phi)$,

$$\text{opt}_{\text{M2S3}}(\phi) - \text{m}_{\text{M2S3}}(\phi, g(\phi, T)) \leq \beta(\text{opt}_{EDK_4}(f(\phi)) - \text{m}_{EDK_4}(f(\phi), T)) .$$

We shall now demonstrate the existence of some such α and β . Recall that in the instance of $\text{Max } 2\text{SAT}^{\leq 3}$, X is the set of variables, C is the set of clauses, and m_i is the number of occurrences of each variable x_i . Note that by the definition of $\text{Max } 2\text{SAT}^{\leq 3}$, $\sum_{i=1}^{|X|} m_i$ is the total number of literals, which must be at most $2|C|$. Note also that for any instance ϕ of $\text{Max } 2\text{SAT}^{\leq 3}$, it must be that $\text{opt}_{\text{M2S3}}(\phi) \geq |C|/2$. This is because a truth assignment satisfying $|C|/2$ clauses can be found using a greedy algorithm that in each step assigns a truth value to a variable occurring in the maximum number of clauses [115]. We can now show that

$$\begin{aligned}
 \text{opt}_{\text{EDK}_4}(f(\phi)) &\leq \sum_{i=1}^{|X|} 3m_i + \text{opt}_{\text{M2S3}}(\phi) && \text{by Lemma 8.11} \\
 &= 3 \sum_{i=1}^{|X|} m_i + \text{opt}_{\text{M2S3}}(\phi) \\
 &\leq 6|C| + \text{opt}_{\text{M2S3}}(\phi) && \text{since } 2|C| \geq \sum_{i=1}^{|X|} m_i \\
 &\leq 13\text{opt}_{\text{M2S3}}(\phi) && \text{since } \text{opt}_{\text{M2S3}}(\phi) \geq |C|/2
 \end{aligned}$$

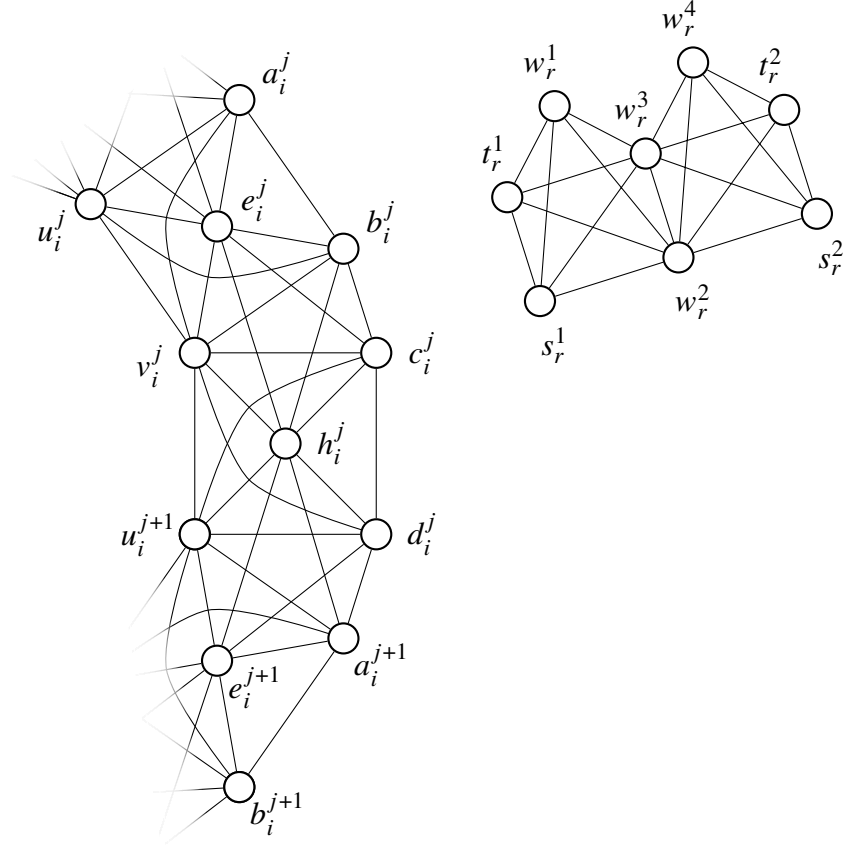
so $\alpha = 13$. We can also show that for any instance ϕ and any edge-disjoint K_4 -packing T in $f(\phi)$,

$$\begin{aligned}
 \text{opt}_{\text{M2S3}}(\phi) - \text{m}_{\text{M2S3}}(\phi, g(\phi, T)) &\leq \text{opt}_{\text{M2S3}}(\phi) - \left(|T| - \sum_{i=1}^{|X|} 3m_i \right) && \text{by Lemma 8.11} \\
 &= \sum_{i=1}^{|X|} 3m_i + \text{opt}_{\text{M2S3}}(\phi) - |T| \\
 &\leq \text{opt}_{\text{EDK}_4}(f(\phi)) - |T| && \text{by Lemma 8.9} \\
 &= \text{opt}_{\text{EDK}_4}(f(\phi)) - \text{m}_{\text{EDK}_4}(f(\phi), T)
 \end{aligned}$$

since $\text{m}_{\text{EDK}_4}(f(\phi), T) = |T|$, which shows that $\beta = 1$. □

8.4.2.3 Edge-disjoint K_5 -packing

In this section we show that EDK_5 is APX-hard even when $\Delta = 9$. The proof uses an L -reduction that follows the same pattern as the one shown in Section 8.4.2.2 for EDK_4 , extending the L -reduction of Caprara and Rizzi [9]. The reduction, shown in Figure 8.2, is as follows. As before, we reduce from $\text{Max } 2\text{SAT}^{\leq 3}$ (Problem 8.2) and construct a set of variable and clause gadgets. For each variable x_i , construct a variable gadget of $8m_i$ vertices, labelled $R_i = \{a_i^j, b_i^j, c_i^j, d_i^j, e_i^j, h_i^j, u_i^j, v_i^j\}$ for each j where $1 \leq j \leq m_i$. For each j where $1 \leq j \leq m_i$, add an edge (if it does not exist already) between each pair

Figure 8.2: The reduction from $\text{Max } 2\text{SAT}^{\leq 3}$ to EDK_5

of vertices in $\{a_i^j, b_i^j, e_i^j, u_i^j, v_i^j\}$; $\{b_i^j, c_i^j, e_i^j, h_i^j, v_i^j\}$; and finally $\{c_i^j, d_i^j, h_i^j, v_i^j, u_i^{j+1}\}$ and $\{d_i^j, a_i^{j+1}, h_i^j, e_i^{j+1}, u_i^{j+1}\}$ if $j < m_i$, otherwise $\{c_i^j, d_i^j, h_i^j, v_i^j, u_i^1\}$ and $\{d_i^j, a_i^1, h_i^j, e_i^1, u_i^1\}$. We shall refer to $\{a_i^j, b_i^j, e_i^j, u_i^j, v_i^j\}$ and $\{c_i^j, d_i^j, h_i^j, v_i^j, u_i^{j+1}\}$ (and $\{c_i^j, d_i^j, h_i^j, v_i^j, u_i^1\}$) as *odd* K_5 s in R_i , and $\{b_i^j, c_i^j, e_i^j, h_i^j, v_i^j\}$ and $\{d_i^j, a_i^{j+1}, h_i^j, e_i^{j+1}, u_i^{j+1}\}$ (and $\{d_i^j, a_i^1, h_i^j, e_i^1, u_i^1\}$) as *even* K_5 s in R_i . At this point $\deg_G(a_i^j) = \deg_G(b_i^j) = \deg_G(c_i^j) = \deg_G(d_i^j) = 6$ and $\deg_G(e_i^j) = \deg_G(h_i^j) = \deg_G(u_i^j) = \deg_G(v_i^j) = 8$ for any j where $1 \leq j \leq m_i$.

We shall now construct the clause gadgets. For each clause c_r , construct a clause gadget of 7 vertices labelled $S_r = \{s_r^1, t_r^1, s_r^2, t_r^2, w_r^1, w_r^2, w_r^3, w_r^4\}$. Add an edge (if it does not exist already) between each pair of vertices in $\{s_r^1, t_r^1, w_r^1, w_r^2, w_r^3\}$ and $\{s_r^2, t_r^2, w_r^2, w_r^3, w_r^4\}$. Label $\{s_r^1, t_r^1, w_r^1, w_r^2, w_r^3\}$ and $\{s_r^2, t_r^2, w_r^2, w_r^3, w_r^4\}$ as P_i^r and P_j^r , where the variables of the literals in c_r are x_i and x_j .

The connection of variable and clause gadgets follows the same pattern as for EDK_4 . For each clause c_r , suppose x_i is the variable of some literal in c_r where c_r contains the j^{th} occurrence of x_i in ϕ . If x_i is the first literal in c_r and occurs positively in c_r then identify a_i^j and s_r^1 , and b_i^j and t_r^1 . Now $\deg_G(a_i^j) = \deg_G(b_i^j) = 9$. Similarly, if x_i is the first literal in c_r and occurs

negatively in c_r then identify b_i^j and s_r^1 , and c_i^j and t_r^1 . If x_i is the second literal in c_r and occurs positively in c_r then identify a_i^j and s_r^2 , and b_i^j and t_r^2 . If x_i is the second literal in c_r and occurs negatively in c_r then identify b_i^j and s_r^2 , and c_i^j and t_r^2 . Now $\Delta = 9$.

As before, the reduction can be performed in polynomial time. We now prove correctness in the first direction.

Lemma 8.13. *If a truth assignment \mathbf{f} for ϕ satisfies at least k clauses then an edge-disjoint K_5 -packing T exists in G where $|T| \geq \sum_{i=1}^{|X|} 2m_i + k$.*

Proof. Suppose \mathbf{f} is a truth assignment for ϕ that satisfies at least k clauses. We shall construct an edge-disjoint K_5 -packing T where $|T| \geq \sum_{i=1}^{|X|} 2m_i + k$. For each variable x_i , add to T the set of even K_5 s in R_i if $\mathbf{f}(x_i)$ is true and otherwise the set of odd K_5 s in R_i . Now $|T| = \sum_{i=1}^{|X|} 3m_i$. For each clause c_r satisfied by \mathbf{f} , it must be that there exists some variable x_i where $\mathbf{f}(x_i)$ is true and x_i occurs positively in c_r , or there exists some variable x_i where $\mathbf{f}(x_i)$ is false and x_i occurs negatively in c_r . As before, in either case add P_i^r to T . Now $|T| = \sum_{i=1}^{|X|} 2m_i + k$. The proof that T is edge disjoint is analogous to the proof in Lemma 8.9. \square

We now prove the second direction. Like before, we say that some edge-disjoint K_5 -packing T in G is *canonical* if for any R_i , T contains either the set of even K_5 s in R_i or the set of odd K_5 s in R_i .

Lemma 8.14. *If T is an edge-disjoint K_5 -packing then there exists a canonical edge-disjoint K_5 -packing T' where $|T'| \geq |T|$.*

Proof. The proof is analogous to the proof of Lemma 8.10. Here we describe the modification of a single variable gadget R_i where T neither contains all even K_5 s nor all odd K_5 s in R_i . It must be that the number of K_5 s in R_i in T is at most $2m_i - 1$.

Suppose x_i occurs in clauses $c_{r_1}, c_{r_2}, \dots, c_{r_{m_i}}$, corresponding to the sets $P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}$. It must be that either at most one K_5 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ exists in T where the corresponding occurrence of x_i is positive, or at most one K_5 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ exists in T where the corresponding occurrence of x_i is negative. In the former case, remove the K_5 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ where the corresponding occurrence of x_i is positive as well as any even K_5 s in R_i in T , then add the set of odd K_5 s not already in T . The number of K_5 s in R_i is now $2m_i$ so since at most one K_5 was removed, which was not in R_i , it follows that the cardinality of T has not decreased. To see that T is still edge-disjoint, observe that any K_5 in $\{P_i^{r_1}, P_i^{r_2}, \dots, P_i^{r_{m_i}}\}$ in T intersects any odd K_5 by at most one vertex. The construction and proof in the latter case is symmetric. \square

Lemma 8.15. *If T is an edge-disjoint K_5 -packing where $|T| = \sum_{i=1}^{|X|} 2m_i + k$ for some integer $k \geq 1$ then exists a truth assignment \mathbf{f} that satisfies at least k clauses.*

Proof. Assume by Lemma 8.14 that T is canonical. It follows that T contains exactly $\sum_{i=1}^{|X|} 2m_i$ K_5 s in variable gadgets and at least k K_5 s in clause gadgets. For each variable x_i , set $\mathbf{f}(x_i)$ to be true if T contains all even K_5 s in R_i and false otherwise. Now consider each clause gadget c_r where S_r contains some K_5 in T , which we label P_i^r . Suppose x_i occurs positively in c_r . It follows that P_i^r contains a_i^j, b_i^j for some j where $1 \leq j \leq 3$. Since T is edge disjoint it follows that T contains the even K_5 s in R_i . By the construction of \mathbf{f} it follows that $\mathbf{f}(x_i)$ is true and thus c_r is satisfied. The proof when x_i occurs negatively in c_r is symmetric. It follows thus that at least k clauses are satisfied by \mathbf{f} . \square

Lemma 8.16. *If $r = 5$ and $\Delta = 9$ then EDK_r is APX-hard.*

Proof. The reduction described runs in polynomial time, and Lemma 8.15 shows how to construct a truth assignment \mathbf{f} that satisfies k clauses given an edge-disjoint K_5 -packing of cardinality $\sum_{i=1}^{|X|} 3m_i + k$ where $k \geq 1$. By Lemmas 8.13 and 8.15, in the reduction a truth assignment \mathbf{f} for ϕ exists that satisfies at least k clauses if and only if there exists an edge-disjoint K_5 -packing of size at least $\sum_{i=1}^{|X|} 3m_i + k$. This reduction is thus an L -reduction with $\alpha = 9$ and $\beta = 1$. \square

We now combine Lemmas 8.12 and 8.12 with the existing result of Caprara and Rizzi [9] in Theorem 8.8.

Theorem 8.8. *If $r \leq 5$ and $\Delta > 2r - 2$ then EDK_r is APX-hard.*

Proof. Caprara and Rizzi [9] prove the case when $r = 3$ and $\Delta = 5$. In Lemma 8.12 we prove the case when $r = 4$ and $\Delta = 7$. In Lemma 8.16 we prove the case when $r = 5$ and $\Delta = 9$. \square

8.5 Summary and future work

To recap, we considered the problem of finding a maximum-cardinality K_r -packing in an undirected graph of fixed maximum degree Δ , subject to the set of selected K_r s being either vertex disjoint (VDK_r) or edge disjoint (EDK_r). It is known that VDK_3 is solvable in linear time if $\Delta = 3$ but APX-hard if $\Delta \geq 4$, and EDK_3 is solvable in linear time if $\Delta = 4$ but APX-hard if $\Delta \geq 5$ [9]. We generalised these results and presented a full complexity classification for both VDK_r and EDK_r . We first showed that VDK_r is solvable in linear time if $\Delta < 3r/2 - 1$ (Theorem 8.2), solvable in polynomial time if $\Delta < 5r/3 - 1$ (Theorem 8.3), and APX-hard if $\Delta \geq \lceil 5r/3 \rceil - 1$ (Theorem 8.6).

We also showed that if $r \geq 6$ then EDK_r is also solvable in linear time if $\Delta < 3r/2 - 1$, solvable in polynomial time if $\Delta < 5r/3 - 1$, and APX-hard if $\Delta \geq \lceil 5r/3 \rceil - 1$. If $r \leq 5$, then

EDK_r is solvable in linear time if $\Delta < 3r/2 - 1$, solvable in polynomial time if $\Delta \leq 2r - 2$, and APX-hard if $\Delta > 2r - 2$.

Some of our polynomial-time algorithms involved finding a maximum independent set in a corresponding intersection graph. In each case, we showed that this intersection graph was claw-free, from which it follows that a maximum independent set in the intersection graph can be found in polynomial time [107, 109]. As we noted in Section 8.3.1, in a more general setting in which graph vertices have weights, it is possible to find an independent set of maximum weight [107, 112]. It might be interesting to use this result to derive polynomial-time algorithms for weighted versions of VDK_r and EDK_r .

Another direction for future work is to generalise other known results for VDK_3 and EDK_3 to VDK_r and EDK_r where $r \geq 3$. For example, Manić and Wakabayashi [98] showed that the known approximation ratio of $(3/2 + \varepsilon)$ for VDK_3 and EDK_3 can be improved upon in the restricted settings where $\Delta = 4$ and $\Delta = 5$, respectively. It might be possible to show a similar improvement of the corresponding approximation ratio for VDK_r and EDK_r (of $r/2 + \varepsilon$, as discussed in Section 8.1) in the setting of an arbitrary fixed maximum degree.

Chapter 9

Conclusion

In this chapter we recap on the contribution of this thesis and discuss some future work relating to 3DR as well as more general problems of coalition formation.

Our main contribution related to *Three-Dimensional Roommates* (3DR). For two models involving \mathcal{B} - and \mathcal{W} -preferences (3DR-B and 3DR-W), we considered the existence of matchings that are stable. We first showed that both associated existence problems are NP-complete. Next, in each model we considered the optimisation problem in which the objective is to construct a matching with the maximum number of non-blocking triples. We showed that an existing result led to a $9/4$ -approximation algorithm in both models and a simple algorithm based on serial dictatorship led to a $3/2$ -approximation in 3DR-B.

In a model of 3DR with additively separable preferences (3DR-AS), we studied stable and envy-free matchings, for three successively weaker definitions of envy-freeness. We considered various restrictions on the agents' valuations and gave a comprehensive complexity classification based on these restrictions. Interestingly, we identified a general trend that shows, for successively weaker solution concepts, either existence or polynomial-time solvability holds under successively weaker preference restrictions. Building on our new result that any instance of 3DR-AS with binary and symmetric preferences must contain a stable matching, we also developed a 2-approximation algorithm for the problem of finding a stable matching with maximum utilitarian welfare in such an instance.

We also presented new results relating to Three-Dimensional Stable Matching with Cyclic Preferences (3-DSM-CYC). In particular, we considered the optimisation problem of finding a matching with the maximum number of non-blocking families. We first presented two different approximation algorithms for this problem in the general case. We then considered a situation in which the preferences of some agents are sufficiently similar to some master list, and showed that the approximation ratio of one algorithm can be improved in relation to a particular similarity measure (specifically the Kendall tau distance [6]).

Finally, we considered a general problem in graph theory that generalises the notion of assigning agents to coalitions of a fixed size, known as the K_r -packing problem. In particular, we studied the restricted case of this problem in which the graph has a fixed maximum degree Δ . It is known for $r = 3$ that the vertex-disjoint (edge-disjoint) variant is solvable in linear time if $\Delta = 3$ ($\Delta = 4$) but APX-hard if $\Delta \geq 4$ ($\Delta \geq 5$). We generalised these results to an arbitrary but fixed $r \geq 3$, and provided a full complexity classification for both the vertex- and edge-disjoint variants in graphs of maximum degree Δ , for all $r \geq 3$.

At the end of each chapter of this thesis we summarised our new results in detail and discussed some closely related angles of possible future work. We shall now discuss some more general directions for future work related to both 3DR and the wider topic of coalition formation.

An immediate open question is to what extent our results relating to 3DR generalise to problems of multidimensional roommates and more general models of coalition formation. We conjecture that our NP-completeness reductions relating to 3DR-B, 3DR-W, and 3DR-AS can all be generalised to a model of k -dimensional roommates (k DR) where $k \geq 3$. We also conjecture that our approximation algorithms for 3-DSM-CYC and 3DR-B can also be generalised, without too much extra work, to k -DSM-CYC and k DR-B, where $k \geq 3$, with the same approximation ratios. In our opinion the most interesting question here concerns our polynomial-time algorithm for the restriction of 3DR-AS in which preferences are binary and symmetric. It is unclear if either a similar algorithm exists for the same restriction in 4DR-AS or if instances of 4DR-AS exist that do not contain a stable matching.

In Chapter 3, we devised a $9/4$ -approximation algorithm for 3-DSM-CYC-MSM, which involved first constructing a corresponding instance of 3GSM and then using Rosenbaum's [75] $9/4$ -approximation algorithm for 3GSM-MSM to find a matching with at least $9n^3/4$ non-blocking families. We also proved similar results in Chapters 4 and 5 for 3DR-B-MSM and 3DR-W-MSM respectively, making use of Rosenbaum's algorithm for 3PSA-MSM. We believe that this approach can be easily generalised to other optimisation problems in variants of either 3GSM or 3PSA. Specifically, we believe that this approach can be generalised for any such variant in which each agent's preference over triples can be expressed as a poset. It follows that a linear extension of each agent's preferences exists, which was the central component of our proofs for 3-DSM-CYC-MSM, 3DR-B-MSM, and 3DR-W-MSM. For example, we believe that it will be straightforward to identify, along these lines, a $9/4$ -approximation algorithm for the corresponding problem in the model of 3DR proposed by Iwama et al. [57] in 2007.

We saw in Chapters 4 and 5 that deciding if a given instance of 3DR-B or 3DR-W contains a stable matching is NP-complete, which contrasts with the analogous models in which coalitions need not have a fixed size, wherein a stable matching is bound to exist and can be found in polynomial time [7, 8]. It seems intuitive that the added restriction of fixed coalition

size makes both problems somehow harder to solve. It would be interesting to identify other problem models that exhibit a similar behaviour, or identify models that counter this intuition. In this direction, one could also explore other restrictions of coalition size, such as flatmate games [69] or lower and upper bounds.

Many existing works relating to fixed-size coalition formation, and in particular those involving multidimensional roommates, propose new models and study related problems in a relatively ad-hoc way. In this thesis we used the common framework of 3DR to formalise three related models of fixed-size coalition formation, and in each one studied the existence of, and complexity of finding, feasible matchings. This approach allowed us to compare analogous results between problems that relate to different systems of preference representation and different solution concepts. For example, we noted in Chapter 5 that it seems difficult to construct an approximation algorithm for 3DR-W-MSM with the same performance guarantee as the algorithm for 3DR-B-MSM. We believe that such a systematic approach helps us explore the interplay between the system of preference representation, solution concept, and coalition size, both in the setting of 3DR as well as in more general models of coalition formation.

As we saw in Chapter 2 (and in Figure 2.1) a multitude of solution concepts and systems of preference representation have been explored in the setting of hedonic games (in which coalitions generally need not have a fixed size) [11]. It remains open to what extent many of these systems and concepts can be transposed either to 3DR or other models involving coalitions of a restricted size. For example, as noted by Bilò et al. [73], it seems unclear whether Nash stability, which involves the individual deviation of agents, can be meaningfully defined in some models of fixed-size coalitions. More generally, it would be interesting to see to what extent the hierarchy of solution concepts defined in the setting of hedonic games [11, 15, 16] (which we discussed in Chapter 2) can be redefined in a model involving fixed-size coalitions.

Glossary of abbreviations

3DR	Three-Dimensional Roommates	2
3DR-AS	Three-Dimensional Roommates with Additively Separable preferences . . .	63
3DR-AS-SMUW	3DR-AS Stable Maximum Utilitarian Welfare problem	97
3DR-B	Three-Dimensional Roommates with \mathcal{B} -preferences	38
3DR-B-MSM	3DR-B Maximally Stable Matching problem	44
3DR-W	Three-Dimensional Roommates with \mathcal{W} -preferences	55
3DR-W-MSM	3DR-W Maximally Stable Matching problem	60
3-DSM-CYC	Three-Dimensional Stable Matching with Cyclic preferences	24
3-DSM-CYC-MSM	3-DSM-CYC Maximally Stable Matching problem	24
3GSM	Three-Gender Stable Marriage problem	16
3PSA	Three-Person Stable Assignment problem	19
ASHG	Additively Separable Hedonic Game	11
DTP	Directed Triangle Packing	11
EDK_r	Edge-Disjoint K_r -packing problem	160
IRLC	Individually Rational Lists of Coalitions	9
LC	Lists of Coalitions	9
Max 2SAT^{≤3}	a restriction of Maximum Satisfiability	168
MIS-3-TF	Maximum Independent Set in 3-regular Triangle-Free graphs	166
PIT	Partition into Triangles	56
SDR	System of Distinct Representatives	56
SM	Stable Marriage problem	14
SR	Stable Roommates problem	14
VDK_r	Vertex-Disjoint K_r -packing problem	160
X3SAT₊⁼³	a restriction of Exact 3-Satisfiability	105

Bibliography

- [1] Jang I, Shin H, Tsourdos A. Anonymous Hedonic Game for Task Allocation in a Large-Scale Multiple Agent System. *IEEE Transactions on Robotics*. 2018;34(6):1534-48. (cited on pages 1 and 9)
- [2] Aumann RJ, Maschler M. The bargaining set for cooperative games. In: *Advances in Game Theory*. vol. 52 of *Annals of Mathematics Studies*. Princeton University Press; 1961. p. 443-72. (cited on page 1)
- [3] Peters H. *Game Theory: A Multi-Leveled Approach*. Springer; 2008. (cited on page 1)
- [4] Sless L, Hazon N, Kraus S, Wooldridge M. Forming k coalitions and facilitating relationships in social networks. *Artificial Intelligence*. 2018;259:217-45. (cited on pages 2, 22, and 63)
- [5] Plaxton CG, Lam CK. On the Existence of Three-Dimensional Stable Matchings with Cyclic Preferences. *Theory of Computing Systems*. 2022;66(3):679-95. Preliminary version appeared as [116]. (cited on pages 3, 18, 24, 36, 39, and 44)
- [6] Kendall MG. *Rank correlation methods*. London, UK: Charles Griffin & Company; 1955. (cited on pages 3, 31, and 178)
- [7] Cechlárová K, Romero-Medina A. Stability in coalition formation games. *International Journal of Game Theory*. 2001;29(4):487-94. (cited on pages 3, 6, 10, 12, 38, 39, and 179)
- [8] Cechlárová K, Hajduková J. Stable partitions with \mathcal{W} -preferences. *Discrete Applied Mathematics*. 2004;138(3):333-47. (cited on pages 3, 10, 55, 61, and 179)
- [9] Caprara A, Rizzi R. Packing triangles in bounded degree graphs. *Information Processing Letters*. 2002;84(4):175-80. (cited on pages 4, 157, 159, 163, 165, 168, 173, and 176)

- [10] Ng C, Hirschberg DS. Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics*. 1991;4(1):245-52. (cited on pages 6, 16, 19, 38, 39, and 53)
- [11] Aziz H, Savani R, Moulin H. Hedonic Games. In: Brandt F, Conitzer V, Endriss U, Lang J, Procaccia AD, editors. *Handbook of Computational Social Choice*. Cambridge University Press; 2016. p. 356-76. (cited on pages 7, 8, 9, 11, 26, 31, and 180)
- [12] Drèze JH, Greenberg J. Hedonic coalitions: optimality and stability. *Econometrica*. 1980;48(4):987-1003. (cited on page 7)
- [13] Aziz H, Brandt F, Harrenstein P. Pareto optimality in coalition formation. In: *Proceedings of SAGT '11: the 4th Symposium on Algorithmic Game Theory*. vol. 6982 of LNCS. Springer; 2011. p. 93-104. (cited on pages 7 and 21)
- [14] Aziz H, Lang J, Monnot J. Computing Pareto Optimal Committees. In: *Proceedings of IJCAI '16: the 25th International Joint Conference on Artificial Intelligence*. AAAI Press; 2016. p. 60-6. (cited on pages 8 and 10)
- [15] Aziz H, Brandt F, Seedig HG. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*. 2013;195:316-34. (cited on pages 8, 11, 12, 13, 63, and 180)
- [16] Barrot N, Yokoo M. Stable and Envy-free Partitions in Hedonic Games. In: *Proceedings of IJCAI '19: the 28th International Joint Conference on Artificial Intelligence*. IJCAI Organization; 2019. p. 67-73. (cited on pages 8, 13, 101, and 180)
- [17] Elkind E, Wooldridge M. Hedonic Coalition Nets. In: *Proceedings of AAMAS '09: the 8th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems; 2009. p. 417-24. (cited on page 9)
- [18] Manlove DF. *Algorithmics of Matching Under Preferences*. Singapore: World Scientific; 2013. (cited on pages 9, 14, 15, 16, and 31)
- [19] Cechlárová K, Hajduková J. Computational complexity of stable partitions with \mathcal{B} -preferences. *International Journal of Game Theory*. 2002;31:353-64. (cited on pages 10 and 38)
- [20] Barberà S, Bossert W, Pattanaik PK. Ranking sets of objects. In: Barberà S, Hammond PJ, Seidl C, editors. *Handbook of Utility Theory*. vol. 2. Springer; 2004. p. 893-977. (cited on pages 10 and 16)
- [21] Hajduková J. Coalition formation games: a survey. *International Game Theory Review*. 2006;8(4):613-41. (cited on pages 10 and 11)

- [22] Kannai Y, Peleg B. A note on the extension of an order on a set to the power set. *Journal of Economic Theory*. 1984;32(1):172-5. (cited on pages 10 and 16)
- [23] Banerjee S, Konishi H, Sönmez T. Core in a simple coalition formation game. *Social Choice and Welfare*. 2001;18:135-53. (cited on page 11)
- [24] Olsen M. Nash Stability in Additively Separable Hedonic Games Is NP-Hard. In: *Proceedings of CiE '07: The 3rd Conference on Computability in Europe*. vol. 4497 of LNCS. Springer; 2007. p. 598-605. (cited on page 11)
- [25] Deng X, Papadimitriou CH. On the Complexity of Cooperative Solution Concepts. *Mathematics of Operations Research*. 1994;19(2):257-66. (cited on page 11)
- [26] Aziz H, Brandl F, Brandt F, Harrenstein P, Olsen M, Peters D. Fractional Hedonic Games. *ACM Transactions on Economics and Computation*. 2019;7(2):1-29. (cited on pages 11 and 13)
- [27] Bogomolnaia A, Jackson MO. The stability of hedonic coalition structures. *Games and Economic Behavior*. 2002;38:201-30. (cited on pages 11 and 155)
- [28] Brandt F, Bullinger M, Wilczynski A. Reaching Individually Stable Coalition Structures in Hedonic Games. In: *Proceedings of AAAI '21: the 35th AAAI Conference on Artificial Intelligence*. AAAI Press; 2021. p. 5211-8. (cited on page 11)
- [29] Huang CC. Two's Company, Three's a Crowd: Stable Family and Threesome Roommates Problems. In: *Proceedings of ESA '07: the 15th European Symposium on Algorithms*. vol. 4698 of LNCS. Springer; 2007. p. 558-69. (cited on pages 11, 16, 19, and 63)
- [30] Ballester C. NP-completeness in hedonic games. *Games and Economic Behaviour*. 2004;49(1):1-30. (cited on page 12)
- [31] Cseh Á, Fleiner T, Harján P. Pareto optimal coalitions of fixed size. *Journal of Mechanism and Institution Design*. 2019;4(1):87-108. (cited on pages 12, 22, and 26)
- [32] Woeginger GJ. A hardness result for core stability in additive hedonic games. *Mathematical Social Sciences*. 2013;65(2):101-4. (cited on pages 12 and 13)
- [33] Ohta K, Barrot N, Ismaili A, Sakurai Y, Yokoo M. Core Stability in Hedonic Games among Friends and Enemies: Impact of Neutrals. In: *Proceedings of IJCAI '17: the 26th International Joint Conference on Artificial Intelligence*. IJCAI Organization; 2017. p. 359-65. (cited on page 12)

- [34] Sung SC, Dimitrov D. Computational complexity in additive hedonic games. *European Journal of Operational Research*. 2010;203(3):635-9. (cited on pages 12 and 63)
- [35] Gairing M, Savani R. Computing Stable Outcomes in Symmetric Additively Separable Hedonic Games. *Mathematics of Operations Research*. 2008;44(3):1101-21. (cited on page 13)
- [36] Johnson DS, Papadimitriou CH, Yannakakis M. How easy is local search? *Journal of Computer and System Sciences*. 1988;37(1):79-100. (cited on page 13)
- [37] Ueda S. Envy based fairness in hedonic games. In: *Proceedings of GECCO '18: the Genetic and Evolutionary Computation Conference*. ACM; 2018. p. 1852-3. (cited on pages 13 and 101)
- [38] West DB. *Combinatorial Mathematics*. Cambridge University Press; 2020. (cited on pages 14 and 61)
- [39] Gale D, Shapley LS. College admissions and the stability of marriage. *American Mathematical Monthly*. 1962;69(1):9-15. (cited on pages 14 and 155)
- [40] Irving RW. An Efficient Algorithm for the “Stable Roommates” Problem. *Journal of Algorithms*. 1985;6(4):577-95. (cited on pages 14 and 61)
- [41] Knuth DE. *Stable Marriage and Its relation to Other Combinatorial Problems*. vol. 10 of CRM Proceedings and Lecture Notes. American Mathematical Society; 1997. English translation of [117]. (cited on pages 14 and 15)
- [42] The Prize in Economic Sciences 2012. NobelPrize.org, Nobel Media AB; 2020. Available from: <https://web.archive.org/web/20220815060610/https://www.nobelprize.org/prizes/economic-sciences/2012/summary/>. (cited on page 14)
- [43] Garey MR, Johnson DS. *Computers and Intractability*. San Francisco, CA: Freeman; 1979. (cited on pages 15, 56, 93, 97, 104, 130, 135, 153, 157, and 169)
- [44] Alkan A. Non-existence of stable threesome matchings. *Mathematical Social Sciences*. 1988;16(2):207-9. (cited on page 15)
- [45] Subramanian A. A New Approach to Stable Matching Problems. *SIAM Journal on Computing*. 1994;23(4):671-700. (cited on page 16)
- [46] Danilov VI. Existence of stable matchings in some three-sided systems. *Mathematical Social Sciences*. 2003;46(2):145-8. (cited on page 17)

- [47] Boros E, Gurvich V, Jaslar S, Krasner D. Stable matchings in three-sided systems with cyclic preferences. *Discrete Mathematics*. 2004;289(1-3):1-10. (cited on pages 17 and 18)
- [48] Biró P, McDermid E. Three-sided stable matchings with cyclic preferences. *Algorithmica*. 2010;58(1):5-18. (cited on pages 17, 18, and 24)
- [49] Pashkovich K, Poirrier L. Three-dimensional stable matching with cyclic preferences. *Optimization Letters*. 2020;14:2615-23. (cited on pages 17, 18, and 24)
- [50] Eriksson K, Sjöstrand J, Strimling P. Three dimensional stable matching with cyclic preferences. *Mathematical Social Sciences*. 2006;52(1):77-87. (cited on page 18)
- [51] Escamocher G, O'Sullivan B. Three-Dimensional Matching Instances Are Rich in Stable Matchings. In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. vol. 10848 of LNCS. Springer; 2018. p. 182-97. (cited on pages 18, 25, 31, 33, 53, and 62)
- [52] Hofbauer J. d -dimensional stable matching with cyclic preferences. *Mathematical Social Sciences*. 2016;82:72-6. (cited on pages 18 and 27)
- [53] Pittel B. On random stable matchings: Cyclic ones with strict preferences and two-sided ones with partially ordered preferences. *Advances in Applied Mathematics*. 2020;120(102061). (cited on pages 18, 53, and 62)
- [54] Lerner EY. Minimal instances with no weakly stable matching for three-sided problem with cyclic incomplete preferences. *Computing Research Repository, Cornell University Library*; 2022. 2107.10102. Available from: <http://arxiv.org/abs/2107.10102>. (cited on pages 18 and 50)
- [55] Cseh Á, Peters J. Three-Dimensional Popular Matching with Cyclic Preferences. In: *Proceedings of AAMAS '22: the 21st International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems; 2022. p. 309-17. (cited on page 18)
- [56] Cseh Á, Escamocher G, Genç B, Quesada L. A collection of Constraint Programming models for the three-dimensional stable matching problem with cyclic preferences. *Constraints*. 2022;27(3):249-83. (cited on pages 19 and 31)
- [57] 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*; 2007. p. 105-12. Archived as Technical Report COMP2007-41 in the Kyoto University Research Information Repository. (cited on pages 20, 38, and 179)

- [58] Bossong U, Schweigert D. Minimal paths on ordered graphs. *Mathematica Slovaca*. 2006;56(1):23-31. (cited on page 20)
- [59] Delort C, Spanjaard O, Weng P. Committee Selection with a Weight Constraint Based on a Pairwise Dominance Relation. In: *Proceedings of ADT '11: the 2nd International Conference on Algorithmic Decision Theory*. vol. 6992 of LNCS. Springer; 2011. p. 28-41. (cited on page 20)
- [60] Lang J, Rey A, Rothe J, Schadrack H, Schend L. Representing and Solving Hedonic Games with Ordinal Preferences and Thresholds. In: *Proceedings of AAMAS '15: the 14th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems; 2015. p. 1229-37. (cited on page 20)
- [61] Iwama K, Miyazaki S, Okamoto K. Inapproximability of stable roommates problem with triple rooms. In: *Proceedings of AAAC '08: the 1st Annual Meeting of the Asian Association for Algorithms and Computation*; 2008. p. 31. (cited on pages 20, 54, and 61)
- [62] Arkin EM, Bae SW, Efrat A, Mitchell JSB, Okamoto K, Polishchuk V. Geometric stable roommates. *Information Processing Letters*. 2009;109(4):219-24. (cited on pages 20, 54, and 63)
- [63] Deineko VG, Woeginger GJ. Two hardness results for core stability in hedonic coalition formation games. *Discrete Applied Mathematics*. 2013;161(13-14). (cited on pages 20 and 93)
- [64] Chen J, Roy S. Multi-Dimensional Stable Roommates in 2-Dimensional Euclidean Space. In: *Proceedings of ESA' 22: the 30th Annual European Symposium on Algorithms*. vol. 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik; 2022. p. 36:1-36:16. (cited on page 20)
- [65] Ostrovsky R, Rosenbaum W. It's Not Easy Being Three: The Approximability of Three-Dimensional Stable Matching Problems. *Computing Research Repository, Cornell University Library*; 2014. 1412.1130. Available from: <http://arxiv.org/abs/1412.1130>. (cited on page 21)
- [66] Ausiello G, Crescenzi P, Gambosi G, Kann V, Marchetti-Spaccamela A, Protasi M. *Complexity and Approximation*. Springer; 1999. (cited on pages 21, 54, 169, and 172)

- [67] Boehmer N, Elkind E. Stable Roommate Problem with Diversity Preferences. In: Proceedings of IJCAI '20: the 29th International Joint Conference on Artificial Intelligence. IJCAI Organization; 2020. p. 96-102. (cited on page 21)
- [68] 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. vol. 12495 of LNCS. Springer; 2020. p. 59-73. (cited on pages 21, 31, and 53)
- [69] 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. International Foundation for Autonomous Agents and Multiagent Systems; 2020. p. 195-203. (cited on pages 22 and 180)
- [70] Wright M, Vorobeychik Y. Mechanism Design for Team Formation. In: Proceedings of AAAI '15: the 29th AAAI Conference on Artificial Intelligence. AAAI Press; 2015. p. 1050-6. (cited on page 22)
- [71] Yekta HA, Bergman D, Day R. On Finding Stable and Efficient Solutions for the Team Formation Problem. Computing Research Repository, Cornell University Library; 2018. 1804.00309. Available from: <http://arxiv.org/abs/1804.00309>. (cited on page 22)
- [72] Li L, Micha E, Nikolov A, Shah N. Partitioning Friends Fairly; 2022. Unpublished manuscript. Available from: <https://web.archive.org/web/20220912204204/https://www.cs.toronto.edu/~nisarg/papers/fair-partition.pdf>. (cited on pages 22 and 23)
- [73] Bilò V, Monaco G, Moscardelli L. Hedonic Games with Fixed-Size Coalitions. In: Proceedings of AAAI '22: the 36th AAAI Conference on Artificial Intelligence. AAAI Press; 2022. p. 9287-95. (cited on pages 23, 98, 101, and 180)
- [74] Alcalde J. Exchange-proofness or divorce-proofness? Stability in one-sided matching markets. *Economic design*. 1994;1:275-87. (cited on page 23)
- [75] Rosenbaum WB. Distributed Almost Stable Matchings [Ph.D. thesis]. University of California, Los Angeles, USA; 2016. (cited on pages 26, 36, 37, 45, 46, 53, 54, 61, 62, and 179)
- [76] Aziz H, Brandt F, Seedig HG. Optimal Partitions in Additively Separable Hedonic Games. In: Proceedings of IJCAI '11: the 22nd International Joint Conference on Artificial Intelligence. AAAI Press; 2011. p. 43-8. (cited on page 26)

- [77] Biró P, Manlove DF, Mittal S. Size versus stability in the Marriage problem. *Theoretical Computer Science*. 2010;411(16-18):1828-41. (cited on page 36)
- [78] Abraham DJ, Biró P, Manlove DF. “Almost stable” matchings in the Roommates problem. In: *Proceedings of WAOA '05: the 3rd Workshop on Approximation and Online Algorithms*. vol. 3879 of *Lecture Notes in Computer Science*. Springer; 2006. p. 1-14. (cited on pages 36 and 54)
- [79] Hamada K, Iwama K, Miyazaki S. The Hospitals/Residents Problem with Lower Quotas. *Algorithmica*. 2016;74:440-65. (cited on page 36)
- [80] Gusfield D, Irving RW. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press; 1989. (cited on page 40)
- [81] Pittel BG, Irving RW. An upper bound for the solvability probability of a random stable roommates instance. *Random Structures and Algorithms*. 1994;5(3):465-86. (cited on pages 53 and 62)
- [82] Kirkpatrick DG, Hell P. On the complexity of general graph factor problems. *SIAM Journal on Computing*. 1983;12(3):601-9. (cited on pages 66 and 158)
- [83] Chataigner F, Manić G, Wakabayashi Y, Yuster R. Approximation algorithms and hardness results for the clique packing problem. *Discrete Applied Mathematics*. 2009;157(7):1396-406. (cited on page 66)
- [84] McDonald J, Puleo GJ, Tennenhouse C. Packing and Covering Directed Triangles. *Graphs and Combinatorics*. 2020;36:1059-63. (cited on page 98)
- [85] Robertson N, Seymour PD. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*. 1984;36(1):49-64. (cited on pages 99 and 156)
- [86] Conforti M, Cornuéjols G, Zambelli G. *Integer Programming*. vol. 271 of *Graduate Texts in Mathematics*. Springer; 2014. (cited on pages 99, 155, and 156)
- [87] Schaefer TJ. The Complexity of Satisfiability Problems. In: *Proceedings of STOC '78: the 10th Annual ACM Symposium on Theory of Computing*; 1978. p. 216-26. (cited on page 104)
- [88] Porschen S, Schmidt T, Speckenmeyer E, Wotzlaw A. XSAT and NAE-SAT of linear CNF classes. *Discrete Applied Mathematics*. 2014;167:1-14. (cited on pages 105, 109, and 127)
- [89] Cechlárová K, Fleiner T, Manlove D. The kidney exchange game. In: *Proceedings of SOR '05: the 8th International Symposium on Operations Research in Slovenia*; 2005.

- p. 77-83. IM Preprint series A, no. 5/2005, PJ Šafárik University, Faculty of Science, Institute of Mathematics. (cited on pages 130 and 135)
- [90] Bullinger M. Single-Agent Dynamics in Hedonic Games; 2021. Talk presented on 17th August 2021 at Dagstuhl Seminar 21331 “Coalition Formation Games” in Dagstuhl, Germany. Available from: http://web.archive.org/web/20220913192100/https://drops.dagstuhl.de/opus/volltexte/2021/15588/pdf/dagrep_v011_i007_p001_21331.pdf. (cited on page 155)
- [91] Edmonds J. Paths, trees and flowers. *Canadian Journal of Mathematics*. 1965;17:449-67. (cited on page 157)
- [92] Micali S, Vazirani VV. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In: *Proceedings of FOCS '80: the 21st Annual IEEE Symposium on Foundations of Computer Science*; 1980. p. 17-27. (cited on page 157)
- [93] Karp RM. On the Computational Complexity of Combinatorial Problems. *Networks*. 1975;5(1):45-68. (cited on page 157)
- [94] Guruswami V, Pandu Rangan C, Chang MS, Chang GJ, Wong CK. The K_r -Packing Problem. *Computing*. 2001;66:79-89. (cited on pages 157 and 158)
- [95] van Rooij JMM, van Kooten-Niekerk ME, Bodlaender HL. Partition Into Triangles on Bounded Degree Graphs. *Theory of Computing Systems*. 2013;52(4):687-718. (cited on pages 158 and 160)
- [96] Hurkens CAJ, Schrijver A. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*. 1989;2(1):68-72. (cited on page 158)
- [97] Halldórsson MM. Approximating Discrete Collections via Local Improvements. In: *Proceedings of SODA '95: the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*; 1995. p. 160-9. (cited on page 158)
- [98] Manić G, Wakabayashi Y. Packing triangles in low degree graphs and indifference graphs. *Discrete Mathematics*. 2008;308(8):1455-71. (cited on pages 158 and 177)
- [99] Takamizawa K, Nishizeki T, Saito N. Linear-time computability of combinatorial problems on series-parallel graphs. *Journal of the ACM*. 1982;29(3):623-41. (cited on page 158)
- [100] Pantel S. Graph Packing Problems [Master's thesis]. Simon Fraser University, Canada; 1999. (cited on page 158)

- [101] Dahlhaus E, Karpinski M. Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Applied Mathematics*. 1998;84(1-3):79-91. (cited on page 158)
- [102] Pedrotti V, de Mello CP. K_r -packing of P_4 -sparse graphs. Institute of Computing, University of Campinas; 2008. IC-08-025. Available from: <https://web.archive.org/web/20220912204041/https://www.ic.unicamp.br/~wtd/2012/Anais/AnaisWTD2008.pdf>. (cited on page 158)
- [103] Hajnal A, Szemerédi E. Proof of a conjecture of P. Erdős. *Combinatorial Theory and Its Applications*. 1970;2:601-23. (cited on page 159)
- [104] Kierstead HA, Kostochka AV. An Ore-type theorem on equitable coloring. *Journal of Combinatorial Theory, Series B*. 2008;98(1):226-34. (cited on page 159)
- [105] Treglown A. The Regularity Lemma and applications to packings in graphs [Ph.D. thesis]. University of Birmingham, UK; 2007. (cited on page 159)
- [106] Kühn D, Osthus D. The Minimum Degree Threshold for Perfect Graph Packings. *Combinatorica*. 2009;29(1):65-107. (cited on page 159)
- [107] Minty GJ. On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*. 1980;28(3):284-304. (cited on pages 160, 163, and 177)
- [108] HN de Ridder et al. Information System on Graph Classes and their Inclusions (ISGCI) [database on the Internet]. 2022 [updated 2022 Feb 22; cited 2022 Sep 11]. Graphclass: cluster. Available from: https://www.graphclasses.org/classes/gc_1237.html. (cited on page 161)
- [109] Sbihi N. Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics*. 1980;29(1):53-76. (cited on pages 163 and 177)
- [110] Kann V. Maximum bounded 3-dimensional matching is MaxSNP-complete. *Information Processing Letters*. 1991;37(1):27-35. (cited on page 163)
- [111] Faenza Y, Oriolo G, Stauffer G. An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$ -algorithm for the weighted stable set problem. In: *Proceedings of SODA '11: the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*; 2011. p. 630-46. (cited on pages 164 and 165)

- [112] Nakamura D, Tamura A. A revision of Minty's algorithm for finding a maximum weight stable set of a claw-free graph. *Journal of the Operations Research Society of Japan*. 2001;44(2):194-204. (cited on pages 164 and 177)
- [113] Crescenzi P. A Short Guide To Approximation Preserving Reductions. In: *Proceedings of CCC '97: The 12th Annual IEEE Conference on Computational Complexity*; 1997. p. 262-73. (cited on pages 166, 167, and 172)
- [114] Berman P, Karpinski M. On Some Tighter Inapproximability Results. In: *Proceedings of ICALP '99: the 26th International Colloquium on Automata, Languages and Programming*. vol. 1644 of *Lecture Notes in Computer Science*. Springer; 1999. p. 200-9. (cited on page 166)
- [115] Vazirani VV. *Approximation Algorithms*. Springer; 2003. (cited on page 173)
- [116] Lam CK, Plaxton CG. On the Existence of Three-Dimensional Stable Matchings with Cyclic Preferences. In: *Proceedings of SAGT '19: the 12th International Symposium on Algorithmic Game Theory*. vol. 11801 of *LNCS*. Springer; 2019. p. 329-42. (cited on page 182)
- [117] Knuth DE. *Mariages Stables et leurs relations avec d'autres problèmes combinatoires*. Presses de L'Université de Montréal; 1976. (cited on page 185)