FAIR DIVISION OF RESOURCES: FROM INDIVISIBLE GOODS TO CAKES

YUEN SHEUNG MAN
(MComp, NUS; B.Sc.(Hons), NUS)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE

2024

Supervisor:
Assistant Professor Warut Suksompong

Examiners:
Associate Professor Jonathan Mark Scarlett
Associate Professor Arnab Bhattacharyya

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Yuen Sheung Man

25 November 2024

Summary

In fair division, the goal is to divide a set of resources amongst several agents so that every agent receives what they perceive to be a fair share. This is often a challenging task. This thesis aims to provide more insights into fair division by studying theories, structures, algorithms, and complexities related to fair allocations in various settings, thereby revealing possibilities and impossibilities within the field.

In Part I, we study instances wherein the goods are indivisible. In the allocation of indivisible goods, the maximum Nash welfare rule has been characterized as the only rule within the class of additive welfarist rules that guarantees envy-freeness up to one good (EF1). We extend this characterization to the class of all welfarist rules. We then examine the structure of EF1 allocations by studying their reachability when agents are allowed to exchange goods sequentially. We investigate whether it is always possible to reach an EF1 allocation from another EF1 allocation via a sequence of exchanges such that every intermediate allocation is also EF1. In circumstances where this can be done, we investigate whether there is also an optimal sequence of such exchanges. Another problem that we study is the reformation of an unfair allocation into an EF1 allocation via such sequences. We investigate the complexity of deciding whether this reformation process is possible and the complexity of computing the number of exchanges needed whenever this is possible. Furthermore, we provide bounds to the number of exchanges required in the reformation process in the worst case.

In Part II, we study instances wherein the goods are divisible. We characterize the existence of a connected strongly-proportional allocation of an interval cake. We devise algorithms to determine this condition and to compute such an allocation if it exists. This problem is investigated along different axes, including whether the agents are hungry, whether the agents have different entitlements, and whether the agents are required to have a small positive value more than their entitlements. We then study the problem wherein the resource is in the form of a graph, also known as a graphical cake. Unlike for the interval cake, a connected envy-free allocation is not guaranteed to exist for a graphical cake. We devise efficient algorithms to compute connected allocations with low envy in a graphical cake. We also derive guarantees when each agent can receive more than one connected piece.

Acknowledgements

I would firstly like to express my deepest gratitude to my advisor, Warut Suksompong, for his guidance throughout my PhD journey. Warut has outperformed his role as an advisor. Whenever I had run out of research ideas, he was always able to offer me slices of his expertise and guide me towards appropriate research topics. Yet, he encouraged me to collaborate with others and gave me the opportunity to pursue my own research interests. His thorough feedback and high standards constantly challenged me to make precise cuts in refining arguments in our research work, while his regular check-ins and unwavering belief in my potential kept me focused and motivated. I have learnt so much about the craft of research from his excellent mentoring and I will always be grateful for his patience and dedication.

I would next like to thank my collaborators: Ayumi Igarashi, Naoyuki Kamiyama, Erel Segal-Halevi, Zsuzsanna Jankó, and Attila Joó. A significant portion of this thesis is based on collaborations with them, and this thesis would not have been possible without their contributions. In addition to enriching my research and academic journey, their desire to participate in thought-provoking debates has promoted a deeper understanding of the field. I particularly enjoyed the mutual exchange of ideas and the collective effort to make each piece of work possible, making research seem like a piece of cake. They have taught me a lot, and I am grateful for the opportunity to have worked with these talented experts.

I would like to extend my appreciation to several others who have also contributed significantly to my PhD journey. Jonathan Scarlett and Arnab Bhattacharyya, members of my Thesis Advisory Committee, sharpened my thoughts through useful feedback for earlier drafts of this thesis, thereby allowing me to enhance the clarity and coherence of this work. I had the opportunity to be mentored by Bryan Kian Hsiang Low in the early part of my PhD programme and by Arnab and Limsoon Wong during their lab rotations in my first semester—these opportunities equipped me with the recipe for success as a beginning researcher. Nicholas Teh played a huge role in my PhD journey by connecting me with other researchers at various conferences; I would also like to thank him for proofreading my thesis. I benefited from enriching academic discussions with other PhD students on various projects, including Flint Xiaofeng Fan, Ruixi Zhang, Rachael Hwee Ling Sim, and Karen Frilya Celine. As the icing on the cake, Yuen Jien Soo ("Uncle Soo") offered constant encouragement and guidance on teaching-related matters when I was a Graduate Tutor, and Keng Hwee Ng made me feel welcome from the very first day of my PhD journey. I also appreciate the support from members in Bryan's lab group and the people in the i^3 and COM1 offices who made my journey more enjoyable and fulfilling.

I am thankful for the financial support provided by the NUS School of Computing via the Graduate Tutorship programme during the first three years of my PhD studies and the Ministry of Education via a research grant subsequently.

Most importantly, I would like to thank my family for their unwavering support throughout my PhD studies. They have been extremely supportive of my decision to pursue my interests. In particular, thank you to Wen Yi for the countless sacrifices made, from adjusting to my schedule to providing emotional support throughout; I would certainly not have made the cut without you!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fair division is the problem of dividing a set of resources amongst several competing parties, each of whom has an interest in it, such that every party receives a share of the resources which they think is fair. This problem is ubiquitous in society, with applications ranging from small-scale ones like divorce settlement, division of inheritance, and university course allocation, to large-scale ones like airspace management, frequency allocation, and international dispute resolution (Brams and Taylor, 1996; Moulin, 2003; Thomson, 2016). The parties amongst whom the resources are to be divided could be individuals, groups of individuals, organizations, or even sovereignties, depending on the application. For simplicity, we shall henceforth refer to them as *agents*.

In fair division, agents agree on mechanisms to reconcile their interests; such mechanisms produce allocations with certain fairness properties. A common example is the "divide-and-choose" procedure for the allocation of a cake between two agents: one agent cuts the cake into two equal parts and allows the other agent to choose a part first, thereby ensuring that it is fair for both agents.[1] This is in contrast to the use of *arbitrators*, who may decide on allocations without taking the agents' interests into account. For example, the division of matrimonial assets in a contested divorce is often left to the court to decide, and may sometimes not be fair to one of the parties due to legal reasons.

Fair division is often challenging for a variety of reasons. The value of the resources in contention may not be uniformly distributed among the resources, making some parts more valuable to some agents than other parts. For example, the airspace around the locality of an airshow is more valuable to the aircrafts exhibited at the airshow during that time and should definitely be prioritized for these aircrafts, and a challenge is to still allocate enough airspace to other non-participating aircrafts in the region. Additionally, the nature of the resources may hinder the parties from receiving fair shares. For example, if there are two children sharing a toy, then only one child can play with the toy at a time, making it unfair to the other child. Besides, there are many different notions of fairness, making it difficult to ascertain whether an allocation is *really* fair.

While its formal study has a long and storied history dating back to the work of Steinhaus

[1] One may argue that it is not as fair for the first agent since they only get exactly 50%, while the second agent may potentially get more than 50%. We refer the reader to Brams and Taylor (1996) for a more extensive discussion on this.

(1948), fair division remains a highly active research area at the intersection of mathematics, economics, and computer science, and has drawn great interest from researchers in various disciplines over the past few decades (Dubins and Spanier, 1961; Stromquist, 1980; Brams and Taylor, 1996; Walsh, 2020). In particular, researchers have recently drawn connections between fair division and various other fields such as graph theory (Bei et al., 2022; Bilò et al., 2022), extremal combinatorics (Berendsohn et al., 2022; Akrami et al., 2023), two-sided matching (Freeman et al., 2021; Igarashi et al., 2023), and differential privacy (Manurangsi and Suksompong, 2023), to name but a few. In practice, the theory developed in this area has been applied to several tools for fairly allocating resources, including Spliddit (Goldman and Procaccia, 2014), Course Match (Budish et al., 2017), Kajibuntan (Igarashi and Yokoyama, 2023), and Fast & Fair (Han and Suksompong, 2024).

In this thesis, we focus on the mathematical and the computational aspects of fair division. We develop *theories* related to fair allocations, such as the properties of welfarist rules that guarantee allocations satisfying a fairness notion called envy-freeness up to one good (EF1) in Chapter 3. We examine *structures* of fair allocations, such as the structure of EF1 allocations and how these allocations can be reached via sequential exchanging of goods in Chapters 4 and 5. We design *algorithms* to compute examples of fair allocations, such as strongly-proportional allocations of cakes in Chapter 6 and approximate envy-free allocations of graphical cakes in Chapter 7. We also study *complexities* in various fair division settings, such as the reachability of EF1 allocations in Chapter 4, the reformability of allocations in Chapter 5, and the existence of strongly-proportional allocations of cakes in Chapter 6.

1.1 Resources

While the nature of the resources depends on the application, these resources can broadly be classified along a few dimensions. Along one of the dimensions is the *desirability* of the resources. Resources that are desirable by every agent are known as *goods*, while resources that are undesirable by every agent are known as *chores*. A deceased's assets are desired by the beneficiaries, and fairly allocating these assets to the beneficiaries may be a challenging problem. On the other hand, household chores are tasks that are undesirable, but yet these need to be distributed amongst the family members fairly. Suksompong (2021) and Guo et al. (2023) provide surveys on fair division of goods and of chores respectively. In this thesis, we focus on *goods*, which are resources that agents either prefer having, or are indifferent towards having.

Resources can also be classified according to their *structure*. Goods that must be allocated wholly to an agent are called *indivisible* goods. An example of an indivisible good is a car, which must be allocated as a whole to an agent, and it would not be feasible to break the car up into smaller parts to be allocated to multiple agents without losing the utility of the car. Indivisible goods are the focus of Part I of this thesis. On the other hand, goods that can be divided into smaller parts are called *divisible* goods. For example, when agents dispute over a piece of land, the land can possibly be divided such that each agent receives a fair share of the land. A *cake*—usually modeled as the unit interval $[0, 1]$—is an example of a divisible good, and the problem of fairly allocating it is known as *cake-cutting*. Divisible goods are the focus of Part II of this thesis. In particular, we focus on dividing a cake in Chapter 6 and dividing a *graphical cake*—wherein the goods are represented by a connected graph—in Chapter 7.

Beyond the scope of this thesis, a line of work studies the division of a mix of indivisible and/or divisible goods and/or chores (Bei et al., 2021; Bhaskar et al., 2021; Aziz et al., 2022; Liu et al., 2024).

1.2 Measure of Value

In order to quantify fairness, agents must be able to assign value to the goods that they receive. Could there be a way for the agents to agree on the value assigned to each good? According to the subjective theory of value, the value of a good cannot be fully determined by the inherent properties of the good, and different agents may assign different values to the same good (Menger, 1871). Indeed, suppose that a group of friends rents a house and wishes to assign the rooms in the house to each of them fairly, perhaps by getting the person with the best room to pay the largest proportion of the rent. One of them may prefer a room at the highest floor due to its good view, while another may desire a room at the middle floor with the easiest access to the bathroom, and yet a third person may like a room at the basement due to its large area. Different people have different preferences, and it would be difficult for them to agree on the value of each room. Since there is no objective measure of value, each agent must therefore have their own subjective measure of value, called a *utility function*. A utility function assigns a value, or utility, to each set of goods, and this value increases as more goods are added to the set.

An agent's value for a good may also depend on the interaction between the goods that the agent already has. A person receiving a tank of fuel may not have much use for it, but its value greatly increases if the agent also has a car, since the fuel can be used to power the car. In this case, cars and fuel are *complementary goods*, which are goods that are typically used together and enhance each other's value. On the other hand, the value of a car to a person decreases if that person already has a motorcycle, since that person would typically only use one of them as a mode of transport. In this case, cars and motorcycles are *substitute goods*, which are goods that have similar functions and decrease each other's value. The presence of complementary or substitute goods makes the utility function of an agent more complicated. Instead, we shall focus only on *independent goods*—goods whose values are not affected by the presence of other goods. As a result, an agent's utility of a set of goods is simply the sum of the utility of each of the goods in the set; this property of the utility function is known as *additivity*.[2]

We also assume that agents' measures of value are not influenced by other agents. In other words, every agent's utility function does not depend on the presence of other agents or on other agents' utility functions.[3]

[2]Non-additive utilities have also been studied in fair division (Stromquist, 1980; Su, 1999; Oh et al., 2021; Ghodsi et al., 2022; Amanatidis et al., 2023b).

[3]By contrast, some works assume the presence of *externalities* (Brânzei et al., 2013; Aziz et al., 2023).

1.3 Fairness

Given an allocation of the goods amongst the agents, we use the agents' utility functions to determine if the allocation is fair. Some fairness notions involve comparing an agent's bundle with the entire set of goods. An example is *maximin-share-fairness*. An agent is said to receive their *maximin share* if the utility of their assigned bundle is at least the utility of some bundle in another allocation where the agent divides all the goods as evenly as possible with respect to their own utility function. Another fairness notion is *proportionality*, which requires each of the n agents to receive a bundle with utility at least $1/n$ of the utility of all the goods. Proportionality is a stronger fairness notion than maximin-share-fairness, since an agent who receives a bundle worth at least $1/n$ of all the goods is guaranteed to receive at least their maximin share.

At first glance, it may seem that maximin-share-fairness and proportionality are reasonable fairness notions. Indeed, in cake-cutting, a proportional (and hence maximin-share-fair) allocation always exists, and there is a simple algorithm to get such an allocation (Steinhaus, 1948). However, a proportional allocation may not always exist when allocating indivisible goods, even for two agents. In the example of dividing a toy between two children, the toy, being an indivisible good, can only be allocated to one of the children, making the allocation *not* proportional for the other child. In fact, computing whether a proportional allocation exists for two agents is NP-hard.[4] Even a maximin-share-fair allocation may not exist for three or more agents (Kurokawa et al., 2018), and the best result so far only guarantees each agent slightly more than 3/4 of their maximin share (Akrami and Garg, 2024).

	good 1	good 2	good 3	good 4	good 5
agent 1	100	400	0	0	0
agent 2	0	100	400	0	0
agent 3	0	0	100	400	0
agent 4	0	0	0	100	400
agent 5	400	0	0	0	100

Figure 1.1: Illustration of the division of five goods amongst five agents. Each number represents the utility of the good to that particular agent. An allocation that gives good i to agent i is proportional, but not envy-free.

Another issue is that these two fairness notions only compare each agent's bundle with all the goods, rather than with other agents' bundles. This gives a myopic view on fairness. For example, suppose that there are five agents and five goods, and that each agent i (for i from 1 to 5) deems that good i has utility 100 and good $i+1$ has utility 400 (good 6 is defined to be the same as good 1); the rest of the goods have zero utility to agent i—see Figure 1.1 for an illustration. An allocation that gives good i to agent i is proportional (and hence maximin-share-fair) since agent i receives a good with utility 100, which is 1/5 of 500, the utility of all the goods. However, from agent i's perspective, agent $i+1$ receives a utility of 400, making the allocation extremely unbalanced against agent i's favour. In this case, agent i *envies* agent $i+1$. A better allocation would be to give good $i+1$ to agent i (for

[4] Via a reduction from PARTITION.

every i) so that every agent receives a utility of 400, and thus envies nobody. We can see that while the aforementioned fairness notions are reasonable, they may induce high envy amongst agents.

In order to mitigate this problem, we consider a fairness notion, called *envy-freeness*, that compares agents' bundles with each other. An allocation is *envy-free* if every agent would rather have their own allocated bundle than another agent's bundle based on their own measure of value. Envy-freeness remains one of the strongest notions of fairness—it can be shown that envy-freeness implies both proportionality and maximin-share-fairness.

However, envy-freeness may not always be possible to achieve. For two agents, proportionality and envy-freeness are equivalent, and we saw in an earlier example that a proportional allocation of one indivisible good cannot be attained, so envy-freeness also cannot be attained in this same example. For a cake, it turns out that an envy-free allocation wherein each agent receives a contiguous piece always exists (Stromquist, 1980), but no finite algorithm can guarantee to find this allocation for three or more agents (Stromquist, 2008). Even without the contiguity requirement, the best known algorithm requires $O(n \uparrow\uparrow 6)$ queries[5] for n agents (Aziz and Mackenzie, 2016), which is a huge number for large n.

Having discussed some of the issues with common fairness notions, we consider variants of these notions in this thesis. The variants depend on the structure of the goods. For indivisible goods, a prominent fairness notion in the literature is *EF1*. In an EF1 allocation of the goods, an agent is allowed to envy another agent only if there exists a good in the latter agent's bundle whose removal would eliminate this envy. A simple mechanism to get an EF1 allocation is the *round-robin* protocol, whereby agents take turns choosing their most valuable good among all the goods that have not been chosen yet. Even for non-additive utility functions, it is well-known that an EF1 allocation always exists, and can moreover be computed in polynomial time (Lipton et al., 2004; Budish, 2011). The simplicity, guaranteed existence, and efficient computation makes EF1 a particularly attractive fairness notion. For further discussion of EF1, we refer to the survey by Amanatidis et al. (2023a). We focus on the fairness notion of EF1 in Chapters 3 to 5.

When allocating a divisible good, EF1 would be too extreme since it would involve eliminating the entire good when comparing envy, when the good can be divided into smaller parts to be allocated to different agents instead. As such, we consider approximations of the envy-freeness notion. We consider, separately, envy-freeness up to a multiplicative factor of a parameter α (*α-EF*) and envy-freeness up to a constant value of a parameter α (*α-additive-EF*). In an α-EF allocation, an agent would not envy another agent up to a *factor* of α based on the former agent's utility, whereas in an α-additive-EF allocation, an agent would not envy another agent up to *difference* of α. We focus on α-EF and α-additive-EF allocations in Chapter 7. Besides envy-freeness, we also consider a stronger version of proportionality, known as *strong-proportionality*, where each of the n agents receives a bundle with utility *strictly more* than $1/n$ of the utility of all the goods. Strong-proportionality is the focus of Chapter 6.

[5] We use Knuth's up-arrow notation here.

1.4 Roadmap of This Thesis

This thesis is divided into two main parts. **Part I (Chapters 3 to 5)** is on fair division of *indivisible* goods, while **Part II (Chapters 6 and 7)** is on fair division of *divisible* goods. **Chapter 2 (Preliminaries)** introduces some notations, definitions, and tools that are used across multiple chapters.

In **Chapter 3 (Extending the Characterization of Maximum Nash Welfare)**, we characterize the class of welfarist rules that guarantee EF1 allocations. In a prior work, Caragiannis et al. (2019) showed that every allocation output by the maximum Nash welfare (MNW) rule satisfies EF1. Suksompong (2023) later showed that MNW is the unique additive welfarist rule that satisfies EF1. We show that the MNW rule is in fact the only (not necessarily additive) welfarist rule that guarantees EF1.

In **Chapter 4 (Reachability of Fair Allocations via Sequential Exchanges)**, we examine the structure of EF1 allocations by studying their reachability when agents are allowed to exchange goods sequentially. Given an initial allocation and a target allocation, both of which are EF1, we are interested in whether the target allocation can be reached from the initial allocation via a sequence of operations such that every intermediate allocation is also EF1; each operation consists of two agents exchanging a pair of goods. We show that this is not always possible even for two agents, and deciding their reachability is PSPACE-complete in general. On the other hand, we prove that reachability is guaranteed for two agents with identical or binary utilities as well as for any number of agents with identical binary utilities. We also examine whether there is an EF1 exchange sequence that is optimal in the number of exchanges required. We show that this is always possible for two agents with identical or binary utilities, and that deciding whether such an optimal sequence exists is NP-hard even for four agents with identical utilities.

In **Chapter 5 (Reforming an Unfair Allocation by Exchanging Goods)**, we continue the study from Chapter 4 on the structure of EF1 allocations. This time, we are interested in whether a given initial allocation, which is now not EF1, can be reformed into an EF1 allocation via a sequence of operations. Similar to the previous chapter, we shall allow agents to exchange a pair of goods in each operation. We investigate the complexity of deciding whether this is possible, and show that it is in P for (a) two agents with identical utilities, (b) a constant number of agents with binary utilities, and (c) any number of agents with identical binary utilities. In these cases, finding the optimal number of such exchanges is also in P. We show that, however, the respective problems are NP-hard for other cases. We also examine the number of exchanges required in the reformation process in the worst case, and show that almost all goods need to be exchanged in the worst case for general utilities, while only half of the goods need to be exchanged in the worst case for identical binary utilities.

In **Chapter 6 (On Connected Strongly-Proportional Cake-Cutting)**, we characterize the existence of a connected strongly-proportional allocation of a cake and study the complexity to decide its existence. We present a simple characterization for hungry agents with equal entitlements, and show that deciding the existence of such an allocation requires

$\Theta(n^2)$ queries. For non-hungry agents or for generic entitlements, deciding the existence of such an allocation, however, requires $\Theta(n \cdot 2^n)$ queries. We also investigate the existence of a connected allocation of a cake where each agent must receive more than a small fixed amount greater than their proportional share, and show that the number of queries is in $\Theta(n \cdot 2^n)$ even for hungry agents with equal entitlements. We also show that no finite algorithm can decide the existence of a connected strongly-proportional allocation of a pie.

In **Chapter 7 (Approximate Envy-Freeness in Graphical Cake Cutting)**, we consider another setting in dividing goods that are divisible—the graphical cake cutting model—wherein the cake lies on the edges of a connected graph. The goal is to investigate the existence of connected allocations of a graphical cake with minimal envy. We show that there always exists a 1/2-additive-EF allocation of a graphical cake, and a $(3+\epsilon)$-EF allocation of a graphical cake in the form of a star graph for any $\epsilon > 0$. In the case where agents have identical utilities, we show that an essentially 2-EF allocation exists. These existence results are accompanied by efficient algorithms to compute such allocations. We also study the existence of allocations with minimal envy when agents are allowed to have a small number of connected pieces, and provide bounds to the maximum number of connected pieces each agent is allowed to have.

1.5 Bibliographic Notes

The material presented in this thesis is based on joint works with other collaborators during my PhD studies. In all the works listed below, I am the only student author.

- Chapter 3 is based on a joint work with Warut Suksompong (2023).

  Sheung Man Yuen and Warut Suksompong. Extending the characterization of maximum Nash welfare. *Economics Letters*, 224:111030, 2023.

- Chapter 4 is based on a joint work with Ayumi Igarashi, Naoyuki Kamiyama, and Warut Suksompong (2024). A preliminary version of it appeared in *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

  Ayumi Igarashi, Naoyuki Kamiyama, Warut Suksompong, and Sheung Man Yuen. Reachability of fair allocations via sequential exchanges. *Algorithmica*, 86(12):3653–3683, 2024.

- Chapter 5 is based on a joint work with Ayumi Igarashi, Naoyuki Kamiyama, and Warut Suksompong (2024).

  Sheung Man Yuen, Ayumi Igarashi, Naoyuki Kamiyama, and Warut Suksompong. Reforming an unfair allocation by exchanging goods. Under submission, 2024.

- Chapter 6 is based on a joint work with Zsuzsanna Jankó, Attila Joó, and Erel Segal-Halevi (2024).

  Zsuzsanna Jankó, Attila Joó, Erel Segal-Halevi, and Sheung Man Yuen. On connected strongly-proportional cake-cutting. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pages 3356–3363, 2024.

- Chapter 7 is based on a joint work with Warut Suksompong (2024). A preliminary version of it appeared in *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

  Sheung Man Yuen and Warut Suksompong. Approximate envy-freeness in graphical cake cutting. *Discrete Applied Mathematics*, 357:112–131, 2024.

Chapter 2

Preliminaries

This chapter introduces some definitions, notation, and tools that we will use across multiple chapters. Preliminaries specific to a single chapter are presented in the chapter itself.

2.1 General Setting

In fair division, there is a set G of goods in contention among a finite set N of agents. The structure of G depends on whether the goods are *indivisible* or *divisible*, which will be elaborated in Sections 2.2 and 2.3 respectively. A *bundle* or *share* is a subset of G. The number of agents is denoted by n. We usually use $1, \dots, n$ to denote the agents and $A_1, \dots, A_n$ to denote the bundles allocated to the respective agents. An *allocation* $\mathcal{A} = (A_1, \dots, A_n)$ is an ordered list of n bundles such that A_i and A_j are disjoint for all distinct $i, j \in N$ and $\bigcup_{i \in N} A_i = G$. The meaning of "disjoint" differs slightly from the usual definition in the case of *divisible goods*—this will be elaborated in Section 2.3 later. When we have $\bigcup_{i \in N} A_i \subsetneq G$ instead, then $\mathcal{A}$ is called a *partial allocation*.

Each agent i has a *utility function* $u_i : 2^G \to \mathbb{R}_{\geq 0}$ that assigns a utility to each bundle. Each agent assigns a utility of 0 to the empty bundle, i.e., $u_i(\varnothing) = 0$. The utility functions satisfy the additivity property, i.e., $u_i(G' \cup G'') = u_i(G') + u_i(G'')$ for all disjoint bundles $G', G'' \subseteq G$. When the agents' utility functions are identical, i.e., $u_i = u_j$ for all $i, j \in N$, we simply use u to denote the common utility function.

A *(fair division) instance* consists of the set of agents N, the set of goods G, and the agents' utility functions $(u_i)_{i \in N}$.

We now give the definitions of the fairness notions on which this thesis is based.

Definition 2.1 (Proportionality)**.** An allocation $\mathcal{A}$ is *proportional* if $u_i(A_i) \geq u_i(G)/n$ for all $i \in N$.

Definition 2.2 (Envy-freeness)**.** An allocation $\mathcal{A}$ is *envy-free* if $u_i(A_i) \geq u_i(A_j)$ for all $i, j \in N$.

Common to both fairness notions, every agent receives a bundle that they are satisfied with in a fair allocation. The satisfaction of an agent i is based on their utility function u_i, and they are satisfied if their utility of their bundle, $u_i(A_i)$, is at least a certain threshold. In the case of a proportional allocation, the threshold is based on their utility of the entire set G—agent i's utility of their bundle must be at least their utility of G divided by the number of agents. In contrast, in the case of an envy-free allocation, the threshold is based on their utility of other agents' bundles—agent i's utility of their bundle must be the highest one among the bundles allocated to all agents. The comparison for envy-freeness is between agent i's utility of their own bundle and *agent i's utility* of other agents' bundles, not between agent i's utility of their own bundle and *other agents' utilities* of their own respective bundles. In other words, in an envy-free allocation, each agent does not care about other agents' preferences as long as the former agent receives a bundle they perceive to be at least as valuable as a bundle received by another agent.

Variants of the fairness notions depend on the type of goods in contention. We shall now discuss each type separately.

2.2 Indivisible Goods

When the goods are indivisible, each of these goods must be allocated wholly to an agent and cannot be shared among multiple agents or divided into parts to be allocated to more than one agent. In this case, G is a *finite* set of discrete goods. The number of goods in G is denoted by m, and the goods are usually denoted by $g_1, \ldots, g_m$. An *(allocation) size vector* $\vec{s} = (s_1, \ldots, s_n)$ is a list of non-negative integers such that $s_i = |A_i|$. This implies that $\sum_{i \in N} s_i = m$, the number of goods in G.

For ease of notation, we write $u_i(g)$ instead of $u_i(\{g\})$ for agent i's utility of a single good g. A utility function u_i is *binary* if $u_i(g) \in \{0, 1\}$ for all $g \in G$. Binary utility functions are relevant in settings where each agent only has two levels of preference for each good: either the good is valuable to them or not.

We now discuss fairness in dividing indivisible goods. Since an allocation satisfying envy-freeness may not always exist, we relax this fairness notion and consider envy-freeness *up to one good* instead.

Definition 2.3 (EF1)**.** An allocation $\mathcal{A}$ is *envy-free up to one good (EF1)* if $u_i(A_i) \geq u_i(A_j)$ or there exists $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{g\})$ for all $i, j \in N$.

We say that agent i is *EF1 towards* agent j if $u_i(A_i) \geq u_i(A_j)$ or there exists $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{g\})$. When agent i is EF1 towards agent j, agent i either does not envy agent j, or agent i envies agent j such that the envy can be eliminated by having agent i choose a good (i.e., a most valuable good from agent i's perspective) to remove from agent j's bundle. In the latter case, the good is not actually removed from agent j's bundle in the allocation, but merely excluded for the purpose of comparing agent i's utility of their own bundle with that of agent j's bundle. In an EF1 allocation, every agent is EF1 towards every other agent.

An EF1 allocation can always be found efficiently (Lipton et al., 2004), which demonstrates the feasibility of EF1. Therefore, EF1 is the main fairness notion considered in Part I of this thesis. In particular, Chapters 3 to 5 are all related to EF1 allocations.

In Chapters 4 and 5, we consider the setting where agents can exchange goods with each other. Define the *exchange graph* $\mathcal{G} = \mathcal{G}(N, G)$ as a simple undirected graph with the following properties: the set of vertices consists of all allocations $\mathcal{A}$ in the instance, and the set of edges consists of all pairs $\{\mathcal{A}, \mathcal{B}\}$ of allocations such that $\mathcal{B} = (B_1, \ldots, B_n)$ can be obtained from $\mathcal{A} = (A_1, \ldots, A_n)$ by having two agents exchange one pair of goods with each other—that is, there exist distinct agents $i, i' \in N$ and goods $g \in A_i$ and $g' \in A_{i'}$ such that $B_i = (A_i \cup \{g'\}) \setminus \{g\}$, $B_{i'} = (A_{i'} \cup \{g\}) \setminus \{g'\}$, and $B_j = A_j$ for all $j \in N \setminus \{i, i'\}$. A path from one allocation to another on the graph is called an *exchange path*. The *distance* between two allocations is the length of a shortest exchange path between them—if such a path does not exist, then the distance is defined to be ∞.

The following result shows that the exchange graph $\mathcal{G}$ consists of many connected components, each represented by a unique size vector $\vec{s} = (s_1, \ldots, s_n)$ such that $\sum_{i \in N} s_i = m$.

Proposition 2.2.1. *Let $\mathcal{A}$ and $\mathcal{B}$ be allocations in an instance. Then, there exists an exchange path from $\mathcal{A}$ to $\mathcal{B}$ if and only if $\mathcal{A}$ and $\mathcal{B}$ have the same size vector.*

Proof. Note that every exchange preserves the size vector of the allocation, since each agent involved in the exchange gives away one good and receives one good in return, while other agents retain their bundles.

($\Rightarrow$) If there exists an exchange path from $\mathcal{A}$ to $\mathcal{B}$, then there exists a non-negative integer T and a list of allocations $(\mathcal{A}^0, \mathcal{A}^1, \ldots, \mathcal{A}^T)$ such that $\mathcal{A}^0 = \mathcal{A}$, $\mathcal{A}^T = \mathcal{B}$, and for each $t \in \{0, \ldots, T-1\}$, $\mathcal{A}^t$ and $\mathcal{A}^{t+1}$ are adjacent on the exchange graph. For each $t \in \{0, \ldots, T-1\}$, $\mathcal{A}^t$ and $\mathcal{A}^{t+1}$ have the same size vector. Therefore, the whole list of allocations, including $\mathcal{A}$ and $\mathcal{B}$, have the same size vector.

($\Leftarrow$) Assume that $\mathcal{A}$ and $\mathcal{B}$ have the same size vector; we shall show the existence of an exchange path from $\mathcal{A}$ to $\mathcal{B}$. We first remedy the goods in agent 1's bundle. If $A_1 = B_1$, then all the goods in agent 1's bundle are correct and we are done. Otherwise, since $|A_1| = |B_1|$, we must have $|A_1 \setminus B_1| = |B_1 \setminus A_1| > 0$. Perform an exchange between a good $g \in A_1 \setminus B_1$ and a good $g' \in B_1 \setminus A_1$. This creates a new allocation where the number of wrong goods in agent 1's bundle decreases by one. By repeating this procedure, we eventually arrive at an allocation with agent 1's bundle remedied. This allocation can be reached from $\mathcal{A}$ via an exchange path. We then remedy the goods in the bundles of agents $2, 3, \ldots, n$ in the same manner until every agent has their own bundle in $\mathcal{B}$. Note that when the goods in agent i's bundle are remedied, there is no exchange of goods involving agents 1 to $i-1$ anymore, and so the bundles of agents 1 to $i-1$ remain correct. By concatenating all these exchange paths, we get an exchange path from $\mathcal{A}$ to $\mathcal{B}$. □

2.3 Divisible Goods

We now consider the setting in which a good may be broken up into smaller parts to be allocated to different agents. The simplest representation of G is by the unit interval $C = [0, 1]$, also known as a *cake*. Each utility function u_i is divisible on the cake, i.e., for every interval $[x_1, x_2] \subseteq C$ and $\lambda \in [0, 1]$, there exists a point $y_i \in [x_1, x_2]$ such that $u_i([x_1, y_i]) =$

$\lambda \cdot u_i([x_1, x_2])$. This implies that the set containing a single point $\{x\}$ is worth zero utility to every agent.

Each agent's bundle is restricted to a finite union of intervals. Since a single point is worth zero utility to every agent, we may assume without loss of generality that intervals are *closed*, and that two agents' bundles are considered *disjoint* if they intersect at a finite set of points (notably, the endpoints of the intervals).

In Chapter 6, we consider the simple case wherein G consists of a single piece of cake, and each agent receives a single (closed) interval in an allocation. The setting in Chapter 7, however, is more complex. We assume that G is in the form of a graph, and that each edge of the graph is isomorphic to a single piece of cake. Each agent receives a finite union of (closed) intervals—where the intervals may belong to different edges—such that it is connected, i.e., for any two points in the agent's bundle, there exists a path between the two points that only traverses the bundle. Vertices, as well as the endpoints of the respective intervals, may be shared between multiple agents. In this chapter, G is known as a *graphical cake*. For simplicity, in both Chapters 6 and 7, we normalize the utility functions to 1, i.e., $u_i(G) = 1$ for all $i \in N$.

We now introduce variants of the common fairness notions that were discussed earlier. We start with two variants of proportionality.

Definition 2.4 (Strong-proportionality)**.** An allocation $\mathcal{A}$ is *strongly-proportional* if $u_i(A_i) > u_i(G)/n$ for all $i \in N$.

Definition 2.5 (α-proportionality)**.** For $\alpha \geq 1$, an allocation $\mathcal{A}$ is *α-proportional* if $u_i(A_i) \geq u_i(G)/(\alpha n)$ for all $i \in N$.

Strong-proportionality is a stronger variant of proportionality in that it requires every agent i to receive *more than* their proportional share from G, rather than *at least* their proportional share. On the other hand, for $\alpha > 1$, α-proportionality is a weaker variant of proportionality in that every agent only needs to receive at least $1/\alpha$ of their proportional share. Note that 1-proportionality and proportionality are equivalent.

Next, we consider two variants of envy-freeness.

Definition 2.6 (α-EF)**.** For $\alpha \geq 1$, an allocation $\mathcal{A}$ is *α-EF* if $u_i(A_i) \geq u_i(A_j)/\alpha$ for all $i, j \in N$.

Definition 2.7 (α-additive-EF)**.** For $\alpha \geq 0$, an allocation $\mathcal{A}$ is *α-additive-EF* if $u_i(A_i) \geq u_i(A_j) - \alpha$ for all $i, j \in N$.

An α-EF allocation allows agents to envy each other, as long as the envy does not exceed a *factor* of α. Analogously, an α-additive-EF allocation requires that the envy does not exceed an *amount* of α. Note that 1-EF and 0-additive-EF are both equivalent to envy-freeness.

Strong-proportionality is the main fairness notion considered in Chapter 6. In the same chapter, we also consider a more general case where each agent's proportional share need not be $u_i(G)/n$. On the other hand, α-EF and α-additive-EF are the main fairness notions considered in Chapter 7. The notion of α-proportionality is also used in Chapter 7 to illustrate certain concepts.

We state the relationships between the different fairness notions.

Proposition 2.3.1. *Let $\mathcal{A}$ be an allocation in an instance for $n \geq 2$ agents, and let $\alpha \geq 1$.*

- *If $\mathcal{A}$ is α-EF, then it is $\left(\alpha - \frac{\alpha-1}{n}\right)$-proportional.*
- *If $\mathcal{A}$ is α-EF, then it is $\left(\frac{\alpha-1}{\alpha+1}\right)$-additive-EF.*
- *If $\mathcal{A}$ is α-proportional, then it is $\left(1 - \frac{2}{\alpha n}\right)$-additive-EF.*

Proof. Let $i \in N$. We prove the three statements in turn.

- Suppose that $\mathcal{A}$ is α-EF, and let $\xi = u_i(A_i)$. The utility of the bundle of every other agent $j \in N$ is at most $\alpha\xi$ to agent i, so
$$1 = \sum_{j \in N} u_i(A_j) \leq \xi + (n-1)\alpha\xi = \xi(\alpha n - \alpha + 1).$$
This gives
$$\xi \geq \frac{1}{\alpha n - \alpha + 1} = \frac{1}{\left(\alpha - \frac{\alpha-1}{n}\right) n},$$
which establishes $\left(\alpha - \frac{\alpha-1}{n}\right)$-proportionality.

- Suppose that $\mathcal{A}$ is α-EF, and let $j \in N$. If $u_i(A_j) > \frac{\alpha}{\alpha+1}$, then $u_i(A_i) < 1 - \frac{\alpha}{\alpha+1} = \frac{1}{\alpha}\left(\frac{\alpha}{\alpha+1}\right) < \frac{u_i(A_j)}{\alpha}$, and the allocation is not α-EF, a contradiction. Hence, $u_i(A_j) \leq \frac{\alpha}{\alpha+1}$. By definition of α-EF we have $u_i(A_i) \geq u_i(A_j)/\alpha$, and so
$$u_i(A_j) - u_i(A_i) \leq u_i(A_j) - \frac{u_i(A_j)}{\alpha} = \left(1 - \frac{1}{\alpha}\right) u_i(A_j) \leq \left(1 - \frac{1}{\alpha}\right)\left(\frac{\alpha}{\alpha+1}\right) = \frac{\alpha-1}{\alpha+1};$$
this shows $\left(\frac{\alpha-1}{\alpha+1}\right)$-additive-EF.

- Suppose that $\mathcal{A}$ is α-proportional, and let $j \in N$. Then $u_i(A_i) \geq \frac{1}{\alpha n}$ and $u_i(A_j) \leq 1 - u_i(A_i) \leq 1 - \frac{1}{\alpha n}$, so
$$u_i(A_j) - u_i(A_i) \leq 1 - \frac{2}{\alpha n},$$
proving $\left(1 - \frac{2}{\alpha n}\right)$-additive-EF. □

In order for algorithms to access the utilities of the agents' bundles, we assume that eval and mark queries are available as in the standard model of Robertson and Webb (1998). More specifically, for any $x, y \in C$ with $x \leq y$, $\textsc{Eval}_i(x, y)$ returns the value of $u_i([x, y])$, and for any $x \in C$ and $r \in [0, u_i([x, 1])]$, $\textsc{Mark}_i(x, r)$ returns the smallest value $z \in [x, 1]$ such that $u_i([x, z]) = r$. For convenience, in Chapter 6, we let $\textsc{Mark}_i(x, r)$ return the *largest* value z instead, and show that the results in the chapter still hold.

Part I

Indivisible Goods

Chapter 3

Extending the Characterization of Maximum Nash Welfare

3.1 Introduction

The fair allocation of indivisible goods—be it artwork, furniture, school supplies, or electronic devices—is a ubiquitous problem in society and has attracted significant interest in economics (Moulin, 2019). Among the plethora of methods that one may use to allocate indivisible goods fairly, the method that has arguably received the most attention in recent years is the *maximum Nash welfare (MNW)* rule. For instance, MNW is used to allocate goods on the popular fair division website Spliddit (Goldman and Procaccia, 2014), which has served hundreds of thousands of users since its launch in 2014.

MNW selects from each profile an allocation that maximizes the product of the agents' utilities, or equivalently, the sum of their logarithms. In an influential work, Caragiannis et al. (2019) showed that every allocation output by MNW satisfies *envy-freeness up to one good (EF1)*: given any two agents, if the first agent envies the second agent, then this envy can be eliminated by removing some good in the second agent's bundle. Recently, Suksompong (2023) provided the first characterization of MNW by showing that it is the unique *additive welfarist rule* that guarantees EF1—an additive welfarist rule selects an allocation maximizing a welfare notion that can be expressed as the sum of some function of the agents' utilities. Suksompong's characterization raises an obvious question: Is MNW also the unique (not necessarily additive) *welfarist rule* that guarantees EF1, where a welfarist rule selects an allocation maximizing a welfare notion that can be expressed as some function of the agents' utilities?

In this chapter, we answer the above question in the affirmative, by extending the characterization of Suksompong (2023) to the class of *all* welfarist rules (whether additive or not). This further solidifies the "unreasonable fairness" of MNW established by Caragiannis et al. (2019).

3.2 Preliminaries

Refer to the preliminaries in Sections 2.1 and 2.2. We now describe other preliminaries specific to this chapter.

A *profile* consists of the set of agents N, the set of goods G, and the agents' utility functions $(u_i)_{i \in N}$.[1] A *rule* maps any given profile to an allocation. Given $n \geq 2$, a *welfare function* is a non-decreasing function $f_n : [0, \infty)^n \to [-\infty, \infty)$. The *welfarist rule with (welfare) function* f_n chooses from each profile an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ that maximizes the *welfare* $f_n(u_1(A_1), \ldots, u_n(A_n))$; if there are multiple such allocations, the rule may choose one arbitrarily.

3.3 Result for Continuous Welfare Functions

In this section, we consider only welfare functions that are continuous—in the prior characterization of *additive* welfarist rules, Suksompong (2023) made the stronger assumption that the welfare function is differentiable. We focus only on profiles that admit an allocation where every agent receives positive utility—for profiles that do not admit an allocation where every agent receives positive utility, MNW requires an additional tie-breaking specification in order to ensure EF1 (Caragiannis et al., 2019).

We now state our characterization. Recall from Section 3.2 that a welfare function is assumed to be non-decreasing on $[0, \infty)^n$.

Theorem 3.3.1. *Fix $n \geq 2$. Let f_n be a welfare function that is continuous and strictly increasing*[2] *on $(0, \infty)^n$. Then, the following three statements are equivalent:*

- *(a) For every profile that admits an allocation where every agent receives positive utility, every allocation that can be chosen by the welfarist rule with function f_n is EF1.*
- *(b) For every profile that admits an allocation where every agent receives positive utility, there exists an EF1 allocation that can be chosen by the welfarist rule with function f_n.*
- *(c) The following two statements hold for f_n:*
  - *(i) There exists a strictly increasing and continuous function $q : (0, \infty) \to (-\infty, \infty)$ such that $f_n(x_1, x_2, \ldots, x_n) = q(x_1 x_2 \cdots x_n)$ for all $x_1, \ldots, x_n > 0$.*
  - *(ii) The inequality $f_n(x_1, x_2, \ldots, x_n) > f_n(y_1, y_2, \ldots, y_n)$ holds for all $x_1, \ldots, x_n > 0$ and $y_1, \ldots, y_n \geq 0$ satisfying $\prod_{i=1}^n y_i = 0$.*

Note that if f_n satisfies (c), then given any profile that admits an allocation where every agent receives positive utility, an allocation can be chosen by the welfarist rule with function f_n if and only if it can be chosen by MNW, so the two rules are effectively equivalent. Hence, Theorem 3.3.1 provides a characterization of MNW among all welfarist rules.

Before proceeding to the proof of our characterization, we first establish a technical lemma.

[1] This was defined as an *instance* in Chapter 2, but we use *profile* in this chapter.

[2] The theorem does not hold without the assumption that f_n is strictly increasing on $(0, \infty)^n$: for example, if f_n is a constant function, then statement (b) holds but (a) does not.

Lemma 3.3.2. *Fix $n \geq 2$. Let $f_n : [0,\infty)^n \to [-\infty,\infty)$ be a function that is continuous on $(0,\infty)^n$. Suppose that*

$$f_n((k+1)x_1, x_2, \ldots, x_{i-1}, kx_i, x_{i+1}, \ldots, x_n) \\ = f_n(kx_1, x_2, \ldots, x_{i-1}, (k+1)x_i, x_{i+1}, \ldots, x_n) \tag{3.1}$$

for all $x_1, \ldots, x_n > 0$, positive integers k, and $i \in N \setminus \{1\}$. Then, there exists a continuous function $q : (0,\infty) \to [-\infty,\infty)$ such that $f_n(x_1, x_2, \ldots, x_n) = q(x_1 x_2 \cdots x_n)$ for all $x_1, \ldots, x_n > 0$.

Proof. Suppose that f_n fulfills assumption (3.1). First, we show that f_n satisfies

$$f_n(x, x_2, \ldots, x_{i-1}, z/x, x_{i+1}, \ldots, x_n) = f_n(y, x_2, \ldots, x_{i-1}, z/y, x_{i+1}, \ldots, x_n) \tag{3.2}$$

for all $i \in N \setminus \{1\}$ and $x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, x, y, z > 0$. Assume without loss of generality that $i = 2$; the proof for any other $i \in \{3, \ldots, n\}$ is analogous. Let $x_3, \ldots, x_n, z > 0$ be fixed throughout. Define $p : (0,\infty) \to [-\infty,\infty)$ by $p(x) := f_n(x, z/x, x_3, \ldots, x_n)$ for all $x > 0$. Note that p is continuous due to the continuity of f_n on $(0,\infty)^n$. For any positive integer k and any $x > 0$, we have

$$\begin{aligned} p\left(\frac{k+1}{k} \cdot x\right) &= f_n\left((k+1) \cdot \frac{x}{k},\, k \cdot \frac{z}{(k+1)x},\, x_3, \ldots, x_n\right) \\ &= f_n\left(k \cdot \frac{x}{k},\, (k+1) \cdot \frac{z}{(k+1)x},\, x_3, \ldots, x_n\right) && \text{(by (3.1))} \\ &= f_n(x, z/x, x_3, \ldots, x_n) \\ &= p(x), \end{aligned}$$

so for any rational number $r = a/b > 1$, we have

$$p(rx) = p\left(\frac{a}{a-1} \cdot \frac{a-1}{a-2} \cdot \ \cdots \ \cdot \frac{b+1}{b} \cdot x\right) = p\left(\frac{a-1}{a-2} \cdot \ \cdots \ \cdot \frac{b+1}{b} \cdot x\right) = \cdots = p(x).$$

Similarly, we have $p(rx) = p(x)$ for any rational number $0 < r < 1$, hence the same equation is true for all positive rational numbers r. Since p is continuous and the positive rational numbers are dense in $(0,\infty)$, we can conclude that p is constant, and thus, $f_n(x, z/x, x_3, \ldots, x_n) = f_n(y, z/y, x_3, \ldots, x_n)$ for all $x, y > 0$. Hence, (3.2) is true for all $i \in N \setminus \{1\}$ and $x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, x, y, z > 0$.

Next, we prove by backward induction that for all integers $t \in \{1, \ldots, n\}$, there exists a continuous function $q_t : (0,\infty)^t \to [-\infty,\infty)$ such that

$$f_n(x_1, \ldots, x_n) = q_t(x_1 x_{t+1} \cdots x_n, x_2, \ldots, x_t)$$

for all $x_1, \ldots, x_n > 0$. Then, $q := q_1$ gives the desired conclusion.

For the base case $t = n$, we have $q_n := f_n|_{(0,\infty)^n}$. For the inductive step, let $t \in \{2, \ldots, n\}$ be given, and assume that such a function q_t exists; we shall prove that q_{t-1} exists as well. Define q_{t-1} by $q_{t-1}(y_1, \ldots, y_{t-1}) := q_t(y_1, \ldots, y_{t-1}, 1)$ for all $y_1, \ldots, y_{t-1} > 0$. Note that q_{t-1}

is continuous on $(0,\infty)^{t-1}$ due to the continuity of q_t on $(0,\infty)^t$. Let $x_1,\ldots,x_n > 0$ be given. Then, by setting $x := x_1$ and $y := z := x_1x_t$, we have

$$\begin{aligned}
f_n(x_1,\ldots,x_n) &= f_n(x,x_2,\ldots,x_{t-1},z/x,x_{t+1},\ldots,x_n) \\
&= f_n(y,x_2,\ldots,x_{t-1},z/y,x_{t+1},\ldots,x_n) && \text{(by (3.2))} \\
&= f_n(x_1x_t,x_2,\ldots,x_{t-1},1,x_{t+1},\ldots,x_n) \\
&= q_t(x_1x_tx_{t+1}\cdots x_n,x_2,\ldots,x_{t-1},1) && \text{(by the inductive hypothesis)} \\
&= q_{t-1}(x_1x_t\cdots x_n,x_2,\ldots,x_{t-1}),
\end{aligned}$$

establishing the inductive step and therefore the lemma. □

We are now ready to prove our main result.

Proof of Theorem 3.3.1. The implication (a) ⇒ (b) is trivial. For the implication (c) ⇒ (a), if f_n satisfies (c), then given a profile that admits an allocation where every agent receives positive utility, every allocation that can be chosen by the welfarist rule with function f_n is also an allocation that can be chosen by MNW, which is known to be EF1 (Caragiannis et al., 2019); hence, f_n also satisfies (a). It therefore remains to prove the implication (b) ⇒ (c). Assume that f_n satisfies (b); we will show that both statements (i) and (ii) of (c) hold.

To prove (i), it suffices to show that f_n satisfies (3.1) for all $x_1,\ldots,x_n > 0$, positive integers k, and $i \in N \setminus \{1\}$. Indeed, once this is shown, Lemma 3.3.2 provides a continuous function $q : (0,\infty) \to [-\infty,\infty)$ satisfying $f_n(x_1,x_2,\ldots,x_n) = q(x_1x_2\cdots x_n)$ for all $x_1,\ldots,x_n > 0$. Note that q must be strictly increasing because f_n is strictly increasing on $(0,\infty)^n$, and $-\infty$ cannot be in the range of q since q is strictly increasing and its domain is an open set in $\mathbb{R}$.

To show (3.1), suppose on the contrary that (3.1) is false for some $x_1,\ldots,x_n > 0$, positive integer k, and $i \in N \setminus \{1\}$; assume without loss of generality that $i = 2$, which means that

$$f_n((k+1)x_1,kx_2,x_3,\ldots,x_n) \neq f_n(kx_1,(k+1)x_2,x_3,\ldots,x_n).$$

Suppose that

$$f_n((k+1)x_1,kx_2,x_3,\ldots,x_n) < f_n(kx_1,(k+1)x_2,x_3,\ldots,x_n);$$

the case where the inequality goes in the opposite direction can be handled similarly. By the continuity of f_n, there exists $\epsilon \in (0,x_1)$ such that

$$f_n((k+1)x_1-\epsilon,kx_2,x_3\ldots,x_n) < f_n(kx_1-\epsilon,(k+1)x_2,x_3,\ldots,x_n). \tag{3.3}$$

Consider a profile with $m = kn+1$ goods, where $G' := \{g_1,\ldots,g_{kn}\} = G \setminus \{g_m\}$, such that

- for each $g \in G'$, we have $u_j(g) = x_j$ for $j \in \{1,2\}$ and $u_j(g) = x_j/k$ for $j \in N \setminus \{1,2\}$;
- $u_1(g_m) = x_1 - \epsilon$, and $u_j(g_m) = 0$ for $j \in N \setminus \{1\}$.

Clearly, this profile admits an allocation where every agent receives positive utility. Let $\mathcal{A}$ be an EF1 allocation chosen by the welfarist rule with function f_n on this profile. Regardless of whom g_m is allocated to, each agent receives at most k goods from G' in $\mathcal{A}$—otherwise, if some agent j receives more than k goods from G', then some other agent receives fewer than k goods from G' by the pigeonhole principle and therefore envies j by more than one good, meaning that $\mathcal{A}$ is not EF1. Since $|G'| = kn$, every agent receives exactly k goods from G'. Furthermore, g_m must be allocated to agent 1; otherwise, the allocation where g_m is allocated to agent 1 (and all other goods are allocated as in $\mathcal{A}$) has a higher welfare than $\mathcal{A}$, contradicting the fact that $\mathcal{A}$ is chosen by the welfarist rule with function f_n. The welfare of $\mathcal{A}$ must not be smaller than that of another allocation where agent 1 receives g_m along with $k-1$ goods from G', agent 2 receives $k+1$ goods from G', and every other agent receives k goods from G' each. This means that

$$f_n((k+1)x_1 - \epsilon, kx_2, x_3 \ldots, x_n) \geq f_n(kx_1 - \epsilon, (k+1)x_2, x_3, \ldots, x_n),$$

contradicting (3.3). This establishes (i).

It remains to prove (ii). Consider any $x_1, \ldots, x_n > 0$ and $y_1, \ldots, y_n \geq 0$ satisfying $\prod_{i=1}^n y_i = 0$. Let $X := \prod_{i=1}^n x_i > 0$. Without loss of generality, assume that $y_1 = \cdots = y_k = 0$ and $Y := \prod_{i=k+1}^n y_i > 0$ for some $k \in \{1, \ldots, n\}$ (if $k = n$, the empty product $\prod_{i=k+1}^n y_i$ is taken to be 1). Define $z_1, \ldots, z_n$ by $z_i := (X/2Y)^{1/k}$ for all $i \in \{1, \ldots, k\}$ and $z_i := y_i$ for all $i \in \{k+1, \ldots, n\}$. Then,

$$\begin{aligned}
f_n(y_1, \ldots, y_n) &\leq f_n(z_1, \ldots, z_n) && \text{(since } f_n \text{ is non-decreasing)}\\
&= q(z_1 \cdots z_k \cdot z_{k+1} \cdots z_n) && \text{(by (i) and since all } z_i\text{'s are positive)}\\
&= q((X/2Y) \cdot y_{k+1} \cdots y_n)\\
&= q(X/2)\\
&< q(X) && \text{(since } q \text{ is strictly increasing)}\\
&= q(x_1 \cdots x_n)\\
&= f_n(x_1, \ldots, x_n), && \text{(by (i) and since all } x_i\text{'s are positive)}
\end{aligned}$$

completing the proof of the theorem. □

3.4 Result for Non-Continuous Welfare Functions

We now consider welfare functions that are not necessarily continuous. It turns out that the characterization of the MNW in this case can be expressed more simply than in the continuous case.

Theorem 3.4.1. *Fix $n \geq 2$. Let f_n be a welfare function that is strictly increasing on $(0, \infty)^n$. Then, the following three statements are equivalent:*

(a) For every profile that admits an allocation where every agent receives positive utility, every allocation that can be chosen by the welfarist rule with function f_n is EF1.

(b) For every profile that admits an allocation where every agent receives positive utility, there exists an EF1 allocation that can be chosen by the welfarist rule with function f_n.

(c) *The welfare function* f_n *satisfies* $f_n(x_1, \ldots, x_n) > f_n(y_1, \ldots, y_n)$ *whenever* $\prod_{i=1}^n x_i > \prod_{i=1}^n y_i$.

For a function f_n satisfying (c), the condition does not say whether $f_n(x_1, \ldots, x_n)$ is equal to $f_n(y_1, \ldots, y_n)$ when $\prod_{i=1}^n x_i = \prod_{i=1}^n y_i$, unlike in the continuous case or in MNW. However, despite this ambiguity, the theorem still says that *all* allocations tied for the greatest product of the agents' utilities of their bundles are also EF1. Indeed, suppose that $\mathcal{A} = (A_1, \ldots, A_n)$ has the greatest product $\prod_{i=1}^n u_i(A_i) > 0$ among all allocations in the profile but is not chosen by some f_n that satisfies (c). Suppose that f_n chooses $\mathcal{B} = (B_1, \ldots, B_n)$ instead. We have $\prod_{i=1}^n u_i(B_i) = \prod_{i=1}^n u_i(A_i)$ but $f_n(u_1(B_1), \ldots, u_n(B_n)) > f_n(u_1(A_1), \ldots, u_n(A_n))$. Then, consider another function f'_n such that

- $f'_n(u_1(A_1), \ldots, u_n(A_n)) = f_n(u_1(B_1), \ldots, u_n(B_n))$,
- $f'_n(u_1(B_1), \ldots, u_n(B_n)) = f_n(u_1(A_1), \ldots, u_n(A_n))$, and
- $f'_n(x_1, \ldots, x_n) = f_n(x_1, \ldots, x_n)$ for all other values of $(x_1, \ldots, x_n)$.

It can be easily verified that f'_n satisfies (c) and chooses $\mathcal{A}$, which, by Theorem 3.4.1, is indeed EF1. Therefore, Theorem 3.4.1 says that any allocation with the greatest product of the agents' utilities of their bundles must be EF1, as long as the product is positive. This is exactly what Caragiannis et al. (2019) says about MNW.

We show that condition (c) is still sufficient to satisfy (a) and (b) despite the condition being more general than in the continuous case. Because of the ambiguity, the set of allocations chosen by a specific welfare function f_n that satisfies (c) is a *subset* of allocations chosen by MNW. This, together with the fact that MNW guarantees EF1, proves the implication (c) $\Rightarrow$ (a), just like in the continuous case. It is also clear that (a) implies (b). Thus, it suffices to prove (b) $\Rightarrow$ (c). We shall establish this fact via a series of lemmas.

Lemma 3.4.2. *Let* f_n *be a welfare function that satisfies statement (b) of Theorem 3.4.1. Then, for any positive integer* k*, distinct* $i, j \in \{1, \ldots, n\}$*, and positive numbers* $x_1, \ldots, x_n$*, we have*

$$f_n(x_1, \ldots, x_n) \geq f_n\left(x_1, \ldots, x_{i-1}, \left(1 + \frac{1}{k}\right) x_i, x_{i+1}, \ldots, x_{j-1}, \left(1 - \frac{1}{k}\right) x_j, x_{j+1}, \ldots, x_n\right).$$

Proof. Consider a profile with $m = kn$ goods such that for all $i \in N$ and $g \in G$, we have $u_i(g) = x_i/k$. The only EF1 allocation $\mathcal{A}$ is when every agent has k goods, and the welfare of this allocation is $f_n(x_1, \ldots, x_n)$. The welfare of another allocation $\mathcal{B}$ in which agent i receives $k+1$ goods, agent j receives $k-1$ goods, and every other agent receives k goods, is $f_n(x_1, \ldots, (1+1/k)x_i, \ldots, (1-1/k)x_j, \ldots, x_n)$. Since $\mathcal{A}$ is the only EF1 allocation, the welfare of $\mathcal{A}$ must be at least the welfare of $\mathcal{B}$. This proves the result. □

Lemma 3.4.3. *Let* f_n *be a welfare function that satisfies statement (b) of Theorem 3.4.1. Then, for any* $i \in \{2, \ldots, n\}$ *and positive numbers* $x_1, \ldots, x_n$*, we have*

$$f_n(x_1, \ldots, x_n) > f_n(z, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

whenever $0 < z < x_1 x_i$.

Proof. Assume without loss of generality that $i = n = 2$; the proof for any other $n \geq 3$ and $i \in \{3, \ldots, n\}$ is analogous. The result is clear for $x_2 = 1$, since f_2 is strictly increasing on $(0, \infty)^2$. We first consider the case where $x_2 > 1$.

Let $r = (z/(x_1 x_2))^{1/2} < 1$. Let k_0 be a positive integer such that $k_0 > 1/(1-r)$—this implies that $r < 1 - 1/k_0$. Define $y = (1 + 1/k_0)x_2$; note that $y > x_2$. Then, we have

$$f_2(x_1, x_2) \geq f_2\left(\left(1 - \frac{1}{k_0}\right)x_1, \left(1 + \frac{1}{k_0}\right)x_2\right) > f_2(rx_1, y),$$

where the first inequality follows from Lemma 3.4.2 and the second inequality follows from the fact that f_2 is strictly increasing on $(0, \infty)^2$.

Now, consider the function $h_1(k) = (1 + 1/k)^{\lfloor k \log x_2 \rfloor}$, where log is the natural logarithm. We have $h_1(k) \leq (1 + 1/k)^{k \log x_2} \to \exp(\log x_2) = x_2$ as $k \to \infty$, and

$$h_1(k) \geq (1 + 1/k)^{k \log x_2 - 1} = (1 + 1/k)^{k \log x_2}(1 + 1/k)^{-1} \to \exp(\log x_2) \cdot 1 = x_2$$

as $k \to \infty$. By the squeeze theorem, we have $\lim_{k \to \infty} h_1(k) = x_2$. Let k_1 be a positive integer such that for all $k \geq k_1$, we have $h_1(k) > rx_2$; this is possible since $rx_2 < x_2$.

Consider also the function $h_2(k) = (1 - 1/k)^{\lfloor k \log x_2 \rfloor}$. By a similar reasoning, we have $\lim_{k \to \infty} h_2(k) = 1/x_2$. Let k_2 be a positive integer such that for all $k \geq k_2$, we have $h_2(k) > 1/y$; this is possible since $1/y < 1/x_2$.

Let $k_3 = \max\{k_1, k_2\}$. Then, by applying the inequality from Lemma 3.4.2 repeatedly, we have

$$\begin{aligned}
f_2(rx_1, y) &\geq f_2\left(\left(1 + \frac{1}{k_3}\right)rx_1, \left(1 - \frac{1}{k_3}\right)y\right) \\
&\geq f_2\left(\left(1 + \frac{1}{k_3}\right)^2 rx_1, \left(1 - \frac{1}{k_3}\right)^2 y\right) \\
&\geq \cdots \\
&\geq f_2\left(\left(1 + \frac{1}{k_3}\right)^{\lfloor k_3 \log x_2 \rfloor} rx_1, \left(1 - \frac{1}{k_3}\right)^{\lfloor k_3 \log x_2 \rfloor} y\right) \\
&= f_2(h_1(k_3)rx_1, h_2(k_3)y) \\
&> f_2(r^2 x_1 x_2, 1) \\
&= f_2(z, 1),
\end{aligned}$$

where the last inequality holds because $h_1(k_3) > rx_2$, $h_2(k_3) > 1/y$, and that f_2 is strictly increasing on $(0, \infty)^2$. Combining all the inequalities, we have $f_2(x_1, x_2) > f_2(z, 1)$, proving the result for $x_2 > 1$.

The case for $x_2 < 1$ is proved similarly. $\square$

Lemma 3.4.4. *Let f_n be a welfare function that satisfies statement (b) of Theorem 3.4.1.*

Then, for any positive numbers $x_1, \ldots, x_n$*, we have*

$$f_n(y, 1, \ldots, 1) > f_n(x_1, \ldots, x_n) > f_n(z, 1, \ldots, 1)$$

whenever $0 < z < \prod_{i=1}^n x_i < y$*.*

Proof. We shall prove only the right inequality. The left inequality can be proven by a symmetrical argument by reversing the inequalities in Lemma 3.4.3.

Let $r = (z/\prod_{i=1}^n x_i)^{1/(n-1)} < 1$. Then, by applying the inequality from Lemma 3.4.3 repeatedly, we have

$$\begin{aligned} f_n(x_1, \ldots, x_n) &> f_n(rx_1x_2, 1, x_3, \ldots, x_n) \\ &> f_n(r^2x_1x_2x_3, 1, 1, x_4, \ldots, x_n) \\ &> \cdots \\ &> f_n(r^{n-1}x_1 \cdots x_n, 1, \ldots, 1) \\ &= f_n(z, 1, \ldots, 1). \end{aligned}$$

□

The result from Lemma 3.4.4 allows us to prove our main result.

Proof of Theorem 3.4.1 (b) ⇒ (c). Let f_n satisfy statement (b). Let $x_1, \ldots, x_n, y_1, \ldots, y_n$ be non-negative numbers satisfying $\prod_{i=1}^n x_i > \prod_{i=1}^n y_i$. First, we consider the case where $\prod_{i=1}^n y_i > 0$. Let w be the mean of $\prod_{i=1}^n x_i$ and $\prod_{i=1}^n y_i$—note that $\prod_{i=1}^n x_i > w > \prod_{i=1}^n y_i$. By Lemma 3.4.4, we have $f_n(x_1, \ldots, x_n) > f_n(w, 1, \ldots, 1) > f_n(y_1, \ldots, y_n)$, which shows that f_n satisfies (c).

Next, we consider the case where $\prod_{i=1}^n y_i = 0$. Let $X := \prod_{i=1}^n x_i > 0$. Without loss of generality, assume that $y_1 = \cdots = y_k = 0$ and $Y := \prod_{i=k+1}^n y_i > 0$ for some $k \in \{1, \ldots, n\}$ (if $k = n$, the empty product $\prod_{i=k+1}^n y_i$ is taken to be 1). Define $z_1, \ldots, z_n$ by $z_i := (X/2Y)^{1/k}$ for all $i \in \{1, \ldots, k\}$ and $z_i := y_i$ for all $i \in \{k+1, \ldots, n\}$. Then, we have

$$f_n(y_1, \ldots, y_n) \leq f_n(z_1, \ldots, z_n) < f_n(x_1, \ldots, x_n),$$

where the first inequality follows because f_n is non-decreasing, and the second inequality follows from the first half of this proof since $0 < \prod_{i=1}^n z_i = X/2 < X < \prod_{i=1}^n x_i$. This shows that f_n satisfies (c). □

3.5 Conclusion

In this chapter, we have shown that the MNW is the unique welfarist rule that guarantees EF1, thereby justifying its "unreasonable fairness".

Chapter 4

Reachability of Fair Allocations via Sequential Exchanges

4.1 Introduction

In fair division, the goal is typically to find an allocation of the resource that is “fair” with respect to the agents’ preferences. When allocating indivisible goods—such as books, clothes, and office supplies—a prominent fairness notion in the literature is *envy-freeness up to one good (EF1)*. In an EF1 allocation of the goods, an agent is allowed to envy another agent only if there exists a good in the latter agent’s bundle whose removal would eliminate this envy. The “up to one good” relaxation is necessitated by the fact that full envy-freeness is sometimes infeasible, as can be seen when two agents compete for a single valuable good. It is well-known that an EF1 allocation always exists regardless of the agents’ valuations for the goods and can moreover be computed in polynomial time (Lipton et al., 2004; Budish, 2011). The simplicity, guaranteed existence, and efficient computation makes EF1 a particularly attractive fairness notion.[1]

In this chapter, we take a different perspective by initiating the study of *reachability* in fair division. Given two fair allocations—an initial allocation and a target allocation—we are interested in whether the target allocation can be reached from the initial allocation via a sequence of operations such that every intermediate allocation is also fair. As an application of our problem, consider a company that wants to redistribute some of its employees between its departments. Since performing the entire redistribution at once may excessively disrupt the operation of the departments, the company prefers to gradually adjust the distribution while maintaining fairness among the departments throughout the process. Another example is a museum that plans to reallocate certain exhibits among its branches—performing one small change at a time can help ensure a seamless transition for the visitors. In this chapter, we shall use EF1 as our fairness benchmark and allow any two agents to *exchange* a pair of goods in an operation. The reachability between EF1 allocations, or lack thereof, is an interesting structural property in itself; similar properties have been studied in other collective

[1] By contrast, it remains unknown whether a stronger fairness notion called *envy-freeness up to any good (EFX)* can always be satisfied (Akrami et al., 2023), whereas another well-studied fairness notion, *maximin share fairness*, does not offer guaranteed existence (Kurokawa et al., 2018).

decision-making scenarios such as voting (Obraztsova et al., 2013, 2020).

Closest to the material in this chapter is perhaps a line of work initiated by Gourvès et al. (2017). These authors considered the “housing market” setting, where the number of agents is the same as the number of goods and each agent receives exactly one good. In their model, a pair of agents is allowed to exchange goods if the two agents are neighbors in a given social network and the exchange benefits both agents. Their paper, along with a series of follow-up papers (Huang and Xiao, 2020; Li et al., 2021; Müller and Bentert, 2021; Ito et al., 2023), explored the complexity of determining whether an allocation can be reached from another allocation in this model and its variants. More broadly, reachability problems are also known as *reconfiguration* problems (Nishimura, 2018); examples of such problems that have been studied include minimum spanning tree (Ito et al., 2011), graph coloring (Johnson et al., 2016), and perfect matching (Bonamy et al., 2019).

4.1.1 Our Results

As is often done in fair division, we assume that every agent is equipped with an additive utility function. We consider an “exchange graph” with allocations as vertices. The first question we study is whether it is always possible for agents to reach a target EF1 allocation from an initial EF1 allocation by exchanging goods sequentially with each other while maintaining the EF1 property in all the intermediate allocations; in other words, we ask whether the subgraph of the exchange graph consisting of all EF1 allocations is *connected*. The second question is whether we could perform this exchange process using as few exchanges as would be required if the intermediate allocations need not be EF1; that is, whether there exists an EF1 exchange path which is *optimal* in terms of the number of exchanges required. Note that each agent’s bundle size remains unchanged throughout the process since every operation is an exchange of goods. Our formal model is described in Section 4.2.

In Section 4.3, we investigate the setting where there are only two agents. Perhaps surprisingly, we establish negative results even for this setting: the EF1 exchange graph may not be connected, and even for those instances in which it is connected, optimal EF1 exchange paths may not exist between EF1 allocations. Therefore, we consider restricted classes of utility functions. We show that an optimal EF1 exchange path always exists between any two EF1 allocations if the utilities are identical *or* binary; this implies the connectivity of the EF1 exchange graph in these cases as well.

In Section 4.4, we explore the general setting of three or more agents. Interestingly, we show that finding the smallest number of exchanges between two allocations is NP-hard in this setting even if we disregard the EF1 restriction. In addition, we establish that deciding whether an EF1 exchange path exists between two allocations is PSPACE-complete, and deciding whether an optimal such path exists is NP-hard even for four agents with identical utilities. We also examine restricted utility functions in more detail. We show that while connectivity of the EF1 exchange graph is guaranteed for identical binary utilities, the same holds neither for identical utilities nor for binary utilities separately. Furthermore, the optimality of EF1 exchange paths cannot be guaranteed even for identical binary utilities. Overall, our findings demonstrate that the case of three or more agents is much less tractable than that of two agents in our setting.

With the exception of hardness results (Theorems 4.4.4, 4.4.5 and 4.4.10), our results are summarized in Table 4.1. For the positive results, we also show that the corresponding exchange paths can be found in polynomial time.

utilities		general	identical	binary	identical binary
$n = 2$	connected?	✗ (Th. 4.3.1)	✓ (Th. 4.3.3)	✓ (Th. 4.3.4)	✓ (Th. 4.3.3)
	optimal?	✗ (Th. 4.3.2)	✓ (Th. 4.3.3)	✓ (Th. 4.3.4)	✓ (Th. 4.3.3)
$n \geq 3$	connected?	✗ (Th. 4.4.8)	✗ (Th. 4.4.9)	✗ (Th. 4.4.8)	✓ (Th. 4.4.6)
	optimal?	✗ (Th. 4.4.7)	✗ (Th. 4.4.7)	✗ (Th. 4.4.7)	✗ (Th. 4.4.7)

Table 4.1: Overview of our results. The top row indicates the class of utility functions considered, "connected?" refers to whether the EF1 exchange graph is always connected, and "optimal?" refers to whether there always exists an optimal EF1 exchange path between any two EF1 allocations provided that the EF1 exchange graph is connected.

4.2 Preliminaries

Refer to the preliminaries in Sections 2.1 and 2.2. We now describe other preliminaries specific to this chapter.

An *instance* consists of a set of agents N, a set of goods G, a size vector $\vec{s}$, and agents' utility functions $(u_i)_{i\in N}$.[2] Given an instance, define the *EF1 exchange graph* $\mathcal{G}^{\text{EF1}}$ as the subgraph of the exchange graph $\mathcal{G}$ induced by all EF1 allocations with $\vec{s}$, i.e., $\mathcal{G}^{\text{EF1}}$ contains all vertices in $\mathcal{G}$ that correspond to EF1 allocations with $\vec{s}$ and all edges in $\mathcal{G}$ incident to two such vertices. As we shall see later, EF1 exchange graphs are not always connected, unlike the subgraph induced by all (possibly not EF1) allocations with $\vec{s}$ in the exchange graph (see Proposition 2.2.1). An exchange path using only the edges in $\mathcal{G}^{\text{EF1}}$ is called an *EF1 exchange path*. An EF1 exchange path is *optimal* if its length is equal to the distance between the two corresponding allocations (in $\mathcal{G}$).

4.3 Two Agents

In this section, we examine properties of the EF1 exchange graph when there are only two agents. We remark that this is an important special case in fair division and has been the focus of several prior papers in the area.[3]

We first consider the question of whether the EF1 exchange graph is necessarily connected. One may intuitively think that with only two agents, an EF1 exchange path is guaranteed between any two EF1 allocations because the two agents only need to consider the envy between themselves. An agent may then carefully select a good from her bundle to exchange with the other agent so as to ensure that the subsequent allocation is also EF1. However, this in fact cannot always be done, as our first result shows.

[2] An *instance* as defined in Chapter 2 excludes the size vector, but we include the size vector in this chapter.
[3] Plaut and Roughgarden (2020, Sec. 1.1.1) discussed the significance of the two-agent setting in detail.

Theorem 4.3.1. *There exists an instance with $n = 2$ agents with the same ordinal preferences over the goods such that the EF1 exchange graph is disconnected.*

Proof. Consider the utility of the goods as follows:

g	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
$u_1(g)$	3	3	2	2	2	2	0	0
$u_2(g)$	3	3	1	1	1	1	0	0

Let $\mathcal{A}$ and $\mathcal{B}$ be allocations such that $A_1 = B_2 = \{g_1, g_2, g_7, g_8\}$ and $A_2 = B_1 = \{g_3, g_4, g_5, g_6\}$; it can be verified that both $\mathcal{A}$ and $\mathcal{B}$ are EF1. If there exists an EF1 exchange path between $\mathcal{A}$ and $\mathcal{B}$, then there exists an EF1 allocation $\mathcal{A}'$ adjacent to $\mathcal{A}$ on the exchange path. Without loss of generality, $\mathcal{A}'$ can be reached from $\mathcal{A}$ by exchanging g_3 with either g_1 or g_7. If g_3 is exchanged with g_1, then agent 1 envies agent 2 by more than one good. If g_3 is exchanged with g_7, then agent 2 envies agent 1 by more than one good. Therefore, neither of these exchanges leads to an EF1 allocation, so $\mathcal{A}'$ cannot be EF1. Hence, no EF1 exchange path exists between $\mathcal{A}$ and $\mathcal{B}$. □

Next, we consider the question of whether an *optimal* EF1 exchange path always exists between two EF1 allocations. By Theorem 4.3.1, even an EF1 exchange path may not exist, so an optimal such path does not necessarily exist either. We therefore focus on instances in which the EF1 exchange graph is *connected*. It turns out that even for such instances, an optimal EF1 exchange path still might not exist.

Theorem 4.3.2. *There exists an instance with $n = 2$ agents satisfying the following properties: the EF1 exchange graph is connected, but for some pair of EF1 allocations, no optimal EF1 exchange path exists between them.*

Proof. Consider $\vec{s} = (3, 3)$ and the utility of the goods as follows:

g	g_1	g_2	g_3	g_4	g_5	g_6
$u_1(g)$	5	3	1	0	2	2
$u_2(g)$	0	3	1	5	2	2

Let $\mathcal{B}$ be the allocation such that $B_1 = \{g_1, g_2, g_3\}$ and $B_2 = \{g_4, g_5, g_6\}$—it can be verified that $\mathcal{B}$ is EF1. We first prove that the EF1 exchange graph is connected by constructing an EF1 exchange path between any EF1 allocation $\mathcal{A}$ and the EF1 allocation $\mathcal{B}$. If g_1 is not with agent 1 or g_4 is not with agent 2 in $\mathcal{A}$, perform any exchange involving g_1 and/or g_4 so that g_1 is now with agent 1 and g_4 is now with agent 2. After the exchange, for each $i \in \{1, 2\}$, agent i's bundle is worth at least 5 to her, while any two goods in agent $(3-i)$'s bundle are worth at most 5 to agent i, so the allocation is EF1. Now, we can exchange the goods in $\{g_2, g_3, g_5, g_6\}$ in an arbitrary order to reach $\mathcal{B}$ after at most two more exchanges.

We next prove that an optimal EF1 exchange path between allocations $\mathcal{A}'$ and $\mathcal{B}'$ does not exist, where $A'_1 = \{g_2, g_3, g_4\}$, $A'_2 = \{g_1, g_5, g_6\}$, $B'_1 = \{g_4, g_5, g_6\}$, and $B'_2 = \{g_1, g_2, g_3\}$; it can be verified that both $\mathcal{A}'$ and $\mathcal{B}'$ are EF1, and the distance between $\mathcal{A}'$ and $\mathcal{B}'$ is 2 (through exchanging $g_2 \leftrightarrow g_5$ and $g_3 \leftrightarrow g_6$). Suppose there exists an optimal EF1 exchange path between $\mathcal{A}'$ and $\mathcal{B}'$, and let $\mathcal{A}''$ be the EF1 allocation between $\mathcal{A}'$ and $\mathcal{B}'$ on the exchange path. Since $\mathcal{A}'$ and $\mathcal{A}''$ are adjacent, one good from $\{g_2, g_3\}$ must be exchanged with one good from $\{g_5, g_6\}$ in $\mathcal{A}'$ to reach $\mathcal{A}''$. However, no matter which goods are exchanged with this restriction, there exists $i \in \{1, 2\}$ such that agent i's bundle is worth 3 to her and agent $(3-i)$'s bundle is worth $5+5$ to agent i, contradicting the EF1 property of $\mathcal{A}''$. Therefore, no optimal EF1 exchange path exists between $\mathcal{A}'$ and $\mathcal{B}'$. □

In light of these negative results, we turn our attention to special classes of utility functions: identical utilities and binary utilities. We prove that for these two classes of utility functions, the EF1 exchange graph is always connected, and moreover, an optimal EF1 exchange path exists between every pair of EF1 allocations.

Theorem 4.3.3. *Let an instance with $n = 2$ agents and identical utilities be given. Then, the EF1 exchange graph is connected. Moreover, there exists an optimal EF1 exchange path between any two EF1 allocations, and this path can be computed in polynomial time.*

Theorem 4.3.4. *Let an instance with $n = 2$ agents and binary utilities be given. Then, the EF1 exchange graph is connected. Moreover, there exists an optimal EF1 exchange path between any two EF1 allocations, and this path can be computed in polynomial time.*

To establish these results, we shall prove by induction on T that two EF1 allocations with distance T have an optimal EF1 exchange path between them. For the base case $T = 0$, an optimal EF1 exchange path trivially exists. For the inductive step, let $T \geq 1$ be given, and assume the inductive hypothesis that any two EF1 allocations with distance $T-1$ have an EF1 exchange path of length $T-1$. Now, let $\mathcal{A} = (A_1, A_2)$ and $\mathcal{B} = (B_1, B_2)$ be any two EF1 allocations with distance T; this means that $|A_1 \setminus B_1| = |A_2 \setminus B_2| = T$. Define $X = A_1 \setminus B_1 = \{x_1, \ldots, x_T\}$ and $Y = A_2 \setminus B_2 = \{y_1, \ldots, y_T\}$. We show that there exist goods $x_\alpha \in X$ and $y_\beta \in Y$ such that exchanging them in $\mathcal{A}$ leads to an EF1 allocation $\mathcal{A}' = (A'_1, A'_2)$. If this is possible, then $|A'_1 \setminus B_1| = |A'_2 \setminus B_2| = T - 1$, which implies that the distance between $\mathcal{A}'$ and $\mathcal{B}$ is $T-1$. By the inductive hypothesis, there exists an EF1 exchange path between $\mathcal{A}'$ and $\mathcal{B}$ of length $T-1$. This means that there exists an EF1 exchange path between $\mathcal{A}$ and $\mathcal{B}$ via $\mathcal{A}'$ of length T, which is optimal, hence completing the proof.

For the time complexity, for each pair of goods from $X \times Y$, one can check in polynomial time whether exchanging them leads to an EF1 allocation. Since there are at most t^2 pairs of goods to check at each step, and there are t steps in the path, the running time claim follows.

Proof of Theorem 4.3.3. We follow the notation and inductive outline described before this proof. Assume that the goods in X and Y are arranged in non-increasing order of utilities, i.e., $u(x_i) \geq u(x_j)$ and $u(y_i) \geq u(y_j)$ whenever $i < j$. Denote $\Delta_t = u(y_t) - u(x_t)$ for all $t \in \{1, \ldots, T\}$. Define $A'_1 = (A_1 \cup \{y_1\}) \setminus \{x_1\}$ and $A'_2 = (A_2 \cup \{x_1\}) \setminus \{y_1\}$ to be the bundles after exchanging x_1 and y_1. If (A'_1, A'_2) is EF1, we are done by induction. Otherwise, we assume without loss of generality that in the allocation (A'_1, A'_2), agent 2 envies agent 1 by more than one good. Let x be a highest-utility good in A_1—we may assume that $x \neq x_t$ for all $t \geq 2$. Since (A_1, A_2) is an EF1 allocation, we have $u(x) \geq \gamma := u(A_1) - u(A_2)$.

If both $x = x_1$ and $\Delta_1 < 0$ are true, then

$$u(A'_2) = u(A_2) - \Delta_1 > u(A_2) \geq u(A_1 \setminus \{x\}) = u(A_1 \setminus \{x_1\}) = u(A'_1 \setminus \{y_1\}),$$

which shows that agent 2 does not envy agent 1 by more than one good in (A'_1, A'_2)—a contradiction. Therefore, we must have $x \neq x_1$ or $\Delta_1 \geq 0$. If $x \neq x_1$, then both x and y_1 belong to A'_1. If $x = x_1$ and $\Delta_1 \geq 0$, then y_1 belongs to A'_1 and $u(y_1) \geq u(x)$. Hence, in either case, we have

$$\max\{u(x), u(y_1)\} < u(A'_1) - u(A'_2) = u(A_1) - u(A_2) + 2\Delta_1,$$

which implies

$$\gamma + 2\Delta_1 > \max\{u(x), u(y_1)\}. \tag{4.1}$$

We claim that there exists $t \in \{2, \ldots, T\}$ such that

$$2\Delta_t \leq u(x) - \gamma. \tag{4.2}$$

Suppose on the contrary that $2\Delta_t > u(x) - \gamma$ for all $t \in \{2, \ldots, T\}$. Since every good in A_1 has value at most $u(x)$ and every good in $B_1 \setminus A_1$ has value at most $u(y_1)$, it holds that every good in B_1 has value at most $\max\{u(x), u(y_1)\}$. As (B_1, B_2) is an EF1 allocation, we have

$$\begin{aligned}
\max\{u(x), u(y_1)\} &\geq u(B_1) - u(B_2) \\
&= (u(A_1) - u(A_2)) + \sum_{t=1}^{T} 2\Delta_t \\
&= \gamma + 2\Delta_1 + \sum_{t=2}^{T} 2\Delta_t \\
&\geq \gamma + 2\Delta_1 + \sum_{t=2}^{T} (u(x) - \gamma) \\
&\geq \gamma + 2\Delta_1,
\end{aligned}$$

where the last inequality holds because $u(x) \geq \gamma$ and $T \geq 1$. This contradicts (4.1). Therefore, let $t \in \{2, \ldots, T\}$ be an index that satisfies (4.2). We now claim that we must have

$$2\Delta_t \geq \max\{u(x), u(y_1)\} - 2u(y_1) - \gamma. \tag{4.3}$$

Suppose on the contrary that $2\Delta_t < \max\{u(x), u(y_1)\} - 2u(y_1) - \gamma$. Then we have

$$\begin{aligned}
\max\{u(x), u(y_1)\} - 2u(y_1) - \gamma &> 2\Delta_t && \text{(by assumption)} \\
&\geq -2u(x_t) && \text{(since } u(y_t) \geq 0\text{)} \\
&\geq -2u(x_1), && \text{(since } u(x_t) \leq u(x_1)\text{)}
\end{aligned}$$

which implies

$$\gamma + 2\Delta_1 < \max\{u(x), u(y_1)\},$$

contradicting (4.1). This establishes (4.3).

Combining inequalities (4.2) and (4.3), we have

$$\begin{aligned} -u(y_1) &\le \max\{u(x) - u(y_1), 0\} - u(y_1) \\ &= \max\{u(x), u(y_1)\} - 2u(y_1) \\ &\le \gamma + 2\Delta_t && \text{(by (4.3))} \\ &\le u(x), && \text{(by (4.2))} \end{aligned}$$

which implies $\gamma + 2\Delta_t \in [-u(y_1), u(x)]$. We claim that exchanging x_t and y_t results in an EF1 allocation, i.e., the allocation comprising $A_1'' = (A_1 \cup \{y_t\}) \setminus \{x_t\}$ and $A_2'' = (A_2 \cup \{x_t\}) \setminus \{y_t\}$ is EF1. This is because

$$u(A_1'') - u(A_2'') = u(A_1) - u(A_2) + 2\Delta_t = \gamma + 2\Delta_t \in [-u(y_1), u(x)],$$

where $x \in A_1''$ and $y_1 \in A_2''$—note that x ($\neq x_t$) and y_1 were not exchanged going from $\mathcal{A}$ to $\mathcal{A}''$. This completes the induction and therefore the proof. □

Proof of Theorem 4.3.4. We follow the notation and inductive outline described before the proof of Theorem 4.3.3. Recall that $X = A_1 \setminus B_1 = \{x_1, \ldots, x_T\}$ and $Y = A_2 \setminus B_2 = \{y_1, \ldots, y_T\}$. Let $G_i = \{g \in G \mid u_i(g) = 1\}$ for $i \in \{1, 2\}$. Note that if $|A_i \cap G_i| > |A_{3-i} \cap G_i|$, then agent i does not envy agent $(3 - i)$ by more than one good after the exchange of any pair of goods. Therefore, if $|A_i \cap G_i| > |A_{3-i} \cap G_i|$ is true for both $i \in \{1, 2\}$, then exchanging any pair of goods from X and Y works.

Otherwise, suppose that $|A_i \cap G_i| \le |A_{3-i} \cap G_i|$ is true for some $i \in \{1, 2\}$, and without loss of generality, assume that $i = 1$. We claim that $|X \cap G_1| \le |Y \cap G_1|$. Suppose by way of contradiction that $|X \cap G_1| > |Y \cap G_1|$. Then,

$$\begin{aligned} |B_1 \cap G_1| &= |((A_1 \setminus X) \cup Y) \cap G_1| \\ &= |A_1 \cap G_1| - |X \cap G_1| + |Y \cap G_1| \\ &\le |A_2 \cap G_1| - 1 \\ &= |((B_2 \setminus X) \cup Y) \cap G_1| - 1 \\ &= |B_2 \cap G_1| - |X \cap G_1| + |Y \cap G_1| - 1 \le |B_2 \cap G_1| - 2, \end{aligned}$$

which means that agent 1 envies agent 2 by more than one good in $\mathcal{B}$, contradicting the assumption that $\mathcal{B}$ is an EF1 allocation. Therefore, we must have $|X \cap G_1| \le |Y \cap G_1|$. Thus, there exists a bijection $\phi : X \to Y$ such that $u_1(x) \le u_1(\phi(x))$ for all $x \in X$; this can be obtained by ensuring that goods in $X \cap G_1$ are mapped to goods in $Y \cap G_1$. Exchanging x and $\phi(x)$ in $\mathcal{A}$ for any $x \in X$ will not make agent 1 envy agent 2 by more than one good.

Now, we consider two cases for agent 2. If $|A_2 \cap G_2| > |A_1 \cap G_2|$, then agent 2 does not envy agent 1 by more than one good after the exchange of any pair of goods. In particular, we can exchange x and $\phi(x)$ for any $x \in X$, and we are done by induction. In the other case, we have $|A_2 \cap G_2| \leq |A_1 \cap G_2|$. We claim that there exists some $x \in X$ such that $u_2(\phi(x)) \leq u_2(x)$. Suppose on the contrary that $u_2(\phi(x)) > u_2(x)$ for all $x \in X$. This means that $u_2(x) = 0$ for all $x \in X$ and $u_2(y) = 1$ for all $y \in Y$, and hence $|Y \cap G_2| - |X \cap G_2| = T \geq 1$, where $T = |X| = |Y|$. Then, we have

$$\begin{aligned}
|B_2 \cap G_2| &= |((A_2 \setminus Y) \cup X) \cap G_2| \\
&= |A_2 \cap G_2| - |Y \cap G_2| + |X \cap G_2| \\
&\leq |A_1 \cap G_2| - 1 \\
&= |((B_1 \setminus Y) \cup X) \cap G_2| - 1 \\
&= |B_1 \cap G_2| - |Y \cap G_2| + |X \cap G_2| - 1 \leq |B_1 \cap G_2| - 2,
\end{aligned}$$

which means that agent 2 envies agent 1 by more than one good in $\mathcal{B}$, contradicting the assumption that $\mathcal{B}$ is an EF1 allocation. Thus, $u_2(\phi(x_t)) \leq u_2(x_t)$ for some $x_t \in X$. In particular, exchanging x_t and $\phi(x_t)$ in $\mathcal{A}$ does not make agent 2 envy agent 1 by more than one good. Hence, exchanging $x_t \in X$ and $\phi(x_t) \in Y$ leads to an EF1 allocation, completing the induction and therefore the proof. $\square$

Since the EF1 exchange graph $\mathcal{G}^{\mathrm{EF1}}$ is a subgraph of the exchange graph $\mathcal{G}$, the distance between two allocations (in $\mathcal{G}$) cannot be greater than the length of the shortest EF1 exchange path between the two allocations in $\mathcal{G}^{\mathrm{EF1}}$. In Theorems 4.3.3 and 4.3.4, the polynomial-time algorithms find EF1 exchange paths in $\mathcal{G}^{\mathrm{EF1}}$ that are optimal in the exchange graph $\mathcal{G}$; such exchange paths must also be the shortest possible ones in $\mathcal{G}^{\mathrm{EF1}}$.

4.4 Three or More Agents

In this section, we address the general case where there are more than two agents. We shall see that this case is less tractable, both existentially and computationally.

Before we present our results on the EF1 exchange graph, we provide some insights on finding the distance between two allocations regardless of EF1 considerations. Observe that finding this distance for two agents is simple, as the distance equals the number of goods from each of the two bundles that need to be exchanged. However, this task is not so trivial for more agents—in fact, we shall show that it is NP-hard. To this end, we draw an interesting connection between this distance and the maximum number of disjoint cycles in a graph constructed based on the allocations. We start off by detailing how to construct such a graph.

Let N, G, and $\vec{s}$ be given, and let $\mathcal{A} = (A_1, \ldots, A_n)$ and $\mathcal{B} = (B_1, \ldots, B_n)$ be two allocations with size vector $\vec{s}$. Define $\mathcal{G}_{\mathcal{A},\mathcal{B}} = (V_{\mathcal{A},\mathcal{B}}, E_{\mathcal{A},\mathcal{B}})$ as a directed multigraph consisting of a set of vertices $V_{\mathcal{A},\mathcal{B}} = N$ and a set of (directed) edges $E_{\mathcal{A},\mathcal{B}} = \{e_1, \ldots, e_m\}$. For each $k \in \{1, \ldots, m\}$, the edge e_k represents the good g_k, and $e_k = (i, j)$ if and only if $g_k \in A_i \cap B_j$, i.e., g_k is in agent i's bundle in $\mathcal{A}$ and in agent j's bundle in $\mathcal{B}$ (possibly $i = j$). Note that for every vertex i, its indegree is equal to its outdegree, which is equal to s_i, the number of goods in agent i's bundle. Let $\mathfrak{C}_{\mathcal{A},\mathcal{B}}$ be the collection of partitions of $E_{\mathcal{A},\mathcal{B}}$ into directed circuits;[4] see Figure 4.1 for an illustration of possible partitions into directed circuits.

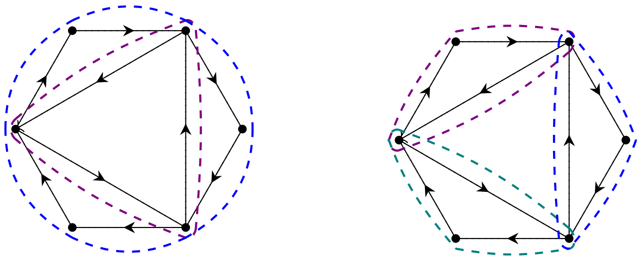


Figure 4.1: Illustration of possible partitions of a graph into directed circuits. The graph has six vertices and nine edges. On the left figure, the partition consists of two directed cycles: one of length 3 and one of length 6. On the right figure, the partition consists of three directed cycles of length 3 each.

Note that $\mathfrak{C}_{\mathcal{A},\mathcal{B}}$ is non-empty—for example, a partition of $E_{\mathcal{A},\mathcal{B}}$ into directed circuits can be constructed in the following way: start with any vertex with an outdegree of at least 1, traverse a path until some vertex v is encountered for the second time, remove the resulting directed cycle from v to itself, and repeat on the remaining graph; the remaining graph still satisfies the condition that every vertex has its indegree equal to its outdegree. Let $c^*_{\mathcal{A},\mathcal{B}} = \max_{C_{\mathcal{A},\mathcal{B}} \in \mathfrak{C}_{\mathcal{A},\mathcal{B}}} |C_{\mathcal{A},\mathcal{B}}|$ be the maximum cardinality of such a partition. Note that a partition with the maximum cardinality must consist only of directed *cycles*; otherwise, if it contains a circuit that passes through a vertex more than once, we can break the circuit into two smaller circuits, contradicting the fact that this partition has the maximum cardinality. We claim that the distance between allocations $\mathcal{A}$ and $\mathcal{B}$ is $m - c^*_{\mathcal{A},\mathcal{B}}$.

Proposition 4.4.1. *Let N, G, and $\vec{s}$ be given, and let $\mathcal{A}$ and $\mathcal{B}$ be two allocations with size vector $\vec{s}$. Then, the distance between $\mathcal{A}$ and $\mathcal{B}$ is $m - c^*_{\mathcal{A},\mathcal{B}}$.*

Proof. We have $m \geq c^*_{\mathcal{A},\mathcal{B}}$ since every partition in $\mathfrak{C}_{\mathcal{A},\mathcal{B}}$ is a partition of a set with cardinality m, so $m - c^*_{\mathcal{A},\mathcal{B}} \geq 0$. We shall prove the result by strong induction on $m - c^*_{\mathcal{A},\mathcal{B}}$. For the base case, let $\mathcal{A}$ and $\mathcal{B}$ be given such that $m - c^*_{\mathcal{A},\mathcal{B}} = 0$. This means that there exists a partition $C_{\mathcal{A},\mathcal{B}} \in \mathfrak{C}_{\mathcal{A},\mathcal{B}}$ such that $|C_{\mathcal{A},\mathcal{B}}| = m = |E_{\mathcal{A},\mathcal{B}}|$. The only way that this is possible is when every edge in $E_{\mathcal{A},\mathcal{B}}$ is a self-loop. Thus, each good appears in the same agent's bundle in $\mathcal{A}$ and $\mathcal{B}$. This means that $\mathcal{A} = \mathcal{B}$, and so the distance between $\mathcal{A}$ and $\mathcal{B}$ is $0 = m - c^*_{\mathcal{A},\mathcal{B}}$.

For the inductive hypothesis, suppose that there exists a non-negative integer T such that for all pairs of allocations $\mathcal{A}$ and $\mathcal{B}$ satisfying $m - c^*_{\mathcal{A},\mathcal{B}} = t$ for any $t \in \{0, \ldots, T\}$, the distance between $\mathcal{A}$ and $\mathcal{B}$ is t. For the inductive step, consider a pair of allocations $\mathcal{A}$ and $\mathcal{B}$ such that $m - c^*_{\mathcal{A},\mathcal{B}} = T + 1$. We shall prove that the distance between $\mathcal{A}$ and $\mathcal{B}$ is $T + 1$.

We first prove that the distance between $\mathcal{A}$ and $\mathcal{B}$ is at most $T + 1$. Let $C_{\mathcal{A},\mathcal{B}} \in \mathfrak{C}_{\mathcal{A},\mathcal{B}}$ be such that $m - |C_{\mathcal{A},\mathcal{B}}| = T + 1 > 0$. Since $|E_{\mathcal{A},\mathcal{B}}| > |C_{\mathcal{A},\mathcal{B}}|$, $C_{\mathcal{A},\mathcal{B}}$ contains at least one directed circuit of length at least two. For notational simplicity, let one such directed circuit be $v_0 \to v_1 \to \cdots \to v_\ell = v_0$ for some $\ell \geq 2$, where $e_k = (v_{k-1}, v_k)$ for $k \in \{1, \ldots, \ell\}$ without loss of generality. From $\mathcal{A}$, exchange $g_{\ell-1}$ (in agent $v_{\ell-2}$'s bundle) with g_ℓ (in agent $v_{\ell-1}$'s

[4] Recall that a directed circuit is a non-empty walk such that the first vertex and the last vertex coincide; we consider a self-loop to be a directed circuit as well.

bundle) to form the allocation $\mathcal{A}'$. This removes the directed circuit $v_0 \to v_1 \to \cdots \to v_\ell$ but introduces two new directed circuits: $v_{\ell-1} \to v_{\ell-1}$ and $v_0 \to v_1 \to \cdots \to v_{\ell-2} \to v_0$. Thus, there exists a partition $C_{\mathcal{A}',\mathcal{B}} \in \mathfrak{C}_{\mathcal{A}',\mathcal{B}}$ such that $|C_{\mathcal{A}',\mathcal{B}}| = |C_{\mathcal{A},\mathcal{B}}| + 1$. This gives $m - |C_{\mathcal{A}',\mathcal{B}}| = T$, and thus $m - c^*_{\mathcal{A}',\mathcal{B}} \leq T$. By the inductive hypothesis, the distance between $\mathcal{A}'$ and $\mathcal{B}$ is at most T. Therefore, the distance between $\mathcal{A}$ and $\mathcal{B}$ (via $\mathcal{A}'$) is at most $T+1$, which is $m - c^*_{\mathcal{A},\mathcal{B}}$.

It remains to prove that the distance between $\mathcal{A}$ and $\mathcal{B}$ is at least $T+1$. Suppose on the contrary that the distance between $\mathcal{A}$ and $\mathcal{B}$ is at most T. Since $c^*_{\mathcal{A},\mathcal{B}} < m$, $\mathcal{A}$ and $\mathcal{B}$ are distinct allocations. Consider a shortest path between $\mathcal{A}$ and $\mathcal{B}$ on the exchange graph, and let $\mathcal{A}'$ be the allocation on this path adjacent to $\mathcal{A}$. By assumption, the distance between $\mathcal{A}'$ and $\mathcal{B}$ is t for some $t < T$. By the inductive hypothesis, $m - c^*_{\mathcal{A}',\mathcal{B}} = t$. Let $C_{\mathcal{A}',\mathcal{B}} \in \mathfrak{C}_{\mathcal{A}',\mathcal{B}}$ be such that $|C_{\mathcal{A}',\mathcal{B}}| = m - t$; by definition of $c^*_{\mathcal{A}',\mathcal{B}}$, $C_{\mathcal{A}',\mathcal{B}}$ must consist only of directed cycles. Now, since $\mathcal{A}$ and $\mathcal{A}'$ are adjacent on the exchange graph, there exist distinct goods g_x and g_y such that exchanging them in allocation $\mathcal{A}$ leads to the allocation $\mathcal{A}'$. Let $i_\mathcal{A}, i_\mathcal{B}, j_\mathcal{A}, j_\mathcal{B} \in N$ be such that $e_x = (i_\mathcal{A}, i_\mathcal{B})$ and $e_y = (j_\mathcal{A}, j_\mathcal{B})$ are edges in $E_{\mathcal{A},\mathcal{B}}$ corresponding to goods g_x and g_y, respectively (some of $i_\mathcal{A}, i_\mathcal{B}, j_\mathcal{A}, j_\mathcal{B}$ may coincide). Accordingly, we must have edges $e'_x = (j_\mathcal{A}, i_\mathcal{B})$ and $e'_y = (i_\mathcal{A}, j_\mathcal{B})$ in $E_{\mathcal{A}',\mathcal{B}}$ (some of these edges may be self-loops). See Figure 4.2 for an illustration.

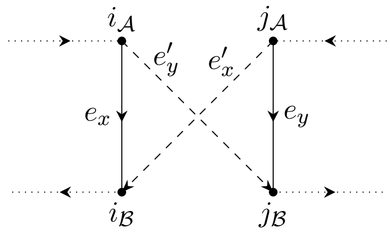


Figure 4.2: The exchange of goods g_x and g_y. The edges e_x and e_y correspond to the respective goods in $G_{\mathcal{A},\mathcal{B}}$, while the edges e'_x and e'_y correspond to those in $G_{\mathcal{A}',\mathcal{B}}$.

We consider two cases; in each case, we will construct a partition $C_{\mathcal{A},\mathcal{B}} \in \mathfrak{C}_{\mathcal{A},\mathcal{B}}$ with at least $|C_{\mathcal{A}',\mathcal{B}}| - 1$ directed circuits.

- **Case 1: e'_x and e'_y belong to different cycles in $C_{\mathcal{A}',\mathcal{B}}$.**
  Let $D_x = j_\mathcal{A} \xrightarrow{e'_x} i_\mathcal{B} \xrightarrow{\cdots} j_\mathcal{A}$ and $D_y = i_\mathcal{A} \xrightarrow{e'_y} j_\mathcal{B} \xrightarrow{\cdots} i_\mathcal{A}$ be the cycles in $C_{\mathcal{A}',\mathcal{B}}$ containing e'_x and e'_y, respectively. Define $C_{\mathcal{A},\mathcal{B}} = (C_{\mathcal{A}',\mathcal{B}} \setminus \{D_x, D_y\}) \cup \{i_\mathcal{A} \xrightarrow{e_x} i_\mathcal{B} \xrightarrow{\cdots} j_\mathcal{A} \xrightarrow{e_y} j_\mathcal{B} \xrightarrow{\cdots} i_\mathcal{A}\}$, where $i_\mathcal{B} \xrightarrow{\cdots} j_\mathcal{A}$ and $j_\mathcal{B} \xrightarrow{\cdots} i_\mathcal{A}$ represent the corresponding (possibly empty) trails in D_x and D_y, respectively. Note that $|C_{\mathcal{A},\mathcal{B}}| = |C_{\mathcal{A}',\mathcal{B}}| - 1$.

- **Case 2: e'_x and e'_y belong to the same cycle in $C_{\mathcal{A}',\mathcal{B}}$.**
  Let $D = j_\mathcal{A} \xrightarrow{e'_x} i_\mathcal{B} \xrightarrow{\cdots} i_\mathcal{A} \xrightarrow{e'_y} j_\mathcal{B} \xrightarrow{\cdots} j_\mathcal{A}$ be the cycle in $C_{\mathcal{A}',\mathcal{B}}$ containing e'_x and e'_y. Define $C_{\mathcal{A},\mathcal{B}} = (C_{\mathcal{A}',\mathcal{B}} \setminus \{D\}) \cup \{i_\mathcal{A} \xrightarrow{e_x} i_\mathcal{B} \xrightarrow{\cdots} i_\mathcal{A},\ j_\mathcal{A} \xrightarrow{e_y} j_\mathcal{B} \xrightarrow{\cdots} j_\mathcal{A}\}$, where the $\xrightarrow{\cdots}$ represents the corresponding (possibly empty) trails in D. Note that $|C_{\mathcal{A},\mathcal{B}}| = |C_{\mathcal{A}',\mathcal{B}}| + 1$.

In either case, there exists a partition $C_{\mathcal{A},\mathcal{B}} \in \mathfrak{C}_{\mathcal{A},\mathcal{B}}$ of cardinality at least $|C_{\mathcal{A}',\mathcal{B}}| - 1 = m - t - 1$. This means that $c^*_{\mathcal{A},\mathcal{B}} \geq m - t - 1$, which implies that $m - c^*_{\mathcal{A},\mathcal{B}} \leq t + 1 < T + 1$. However, this contradicts the assumption that $m - c^*_{\mathcal{A},\mathcal{B}} = T + 1$. Therefore, the distance between $\mathcal{A}$ and $\mathcal{B}$ is at least $T + 1$, as desired. $\square$

Having found a correspondence for the distance between two allocations, a natural question is whether there exists an efficient algorithm to compute this distance. It turns out that computing this distance is an NP-hard problem, so no polynomial-time algorithm exists unless P = NP. We show this via a series of reductions.

We start by considering the decision problem Directed Triangle Partition: given a directed graph with no directed cycles of length 1 or 2, determine whether there is a partition of edges into triangles (i.e., directed cycles of length 3). This decision problem is NP-hard via a reduction from 3SAT. The idea is similar to that used by Holyer (1981) in his proof of the corresponding result for *undirected* graphs; the details are involved and can be found in Appendix A.

Lemma 4.4.2. *Directed Triangle Partition is NP-hard.*

We now use Lemma 4.4.2 to show that computing $c^*_{\mathcal{A},\mathcal{B}}$ is NP-hard.

Lemma 4.4.3. *Given a directed graph such that for each vertex, its indegree and outdegree are equal, computing the maximum cardinality of a partition of the edges into directed cycles is an NP-hard problem.*

Proof. The result follows from reducing Directed Triangle Partition to the problem of deciding whether there exists a partition of the edges of a directed graph into $|E|/3$ directed cycles. Let $\widetilde{\mathcal{G}} = (V, E)$ be an instance of Directed Triangle Partition. If there is some vertex with unequal indegree and outdegree, then $\widetilde{\mathcal{G}}$ cannot be edge-partitioned into triangles. Otherwise, since $\widetilde{\mathcal{G}}$ does not have cycles of length 1 or 2 (by definition of Directed Triangle Partition), the edges of $\widetilde{\mathcal{G}}$ can be partitioned into triangles if and only if the maximum cardinality of a partition of the edges into directed cycles is $|E|/3$. Since Directed Triangle Partition is NP-hard by Lemma 4.4.2, so is the problem of finding the maximum cardinality of a partition of the edges into directed cycles. $\square$

Proposition 4.4.1 and Lemma 4.4.3 imply the following theorem.

Theorem 4.4.4. *Finding the distance between two allocations is an NP-hard problem.*

Proof. Start with an instance $\widetilde{\mathcal{G}} = (V, E)$ of the problem described in Lemma 4.4.3, and denote $V = \{v_1, \dots, v_n\}$. We shall construct, in polynomial time, an instance of the problem of finding the distance between two allocations. Let $N = \{1, \dots, n\}$ be the set of agents, $G = \{g_e\}_{e \in E}$ be the set of goods, and $s_i = |\{e \in E \mid \exists j \in N,\ e = (v_i, v_j)\}|$ be the size of agent i's bundle for each $i \in N$. The initial allocation $\mathcal{A} = (A_1, \dots, A_n)$ and target allocation $\mathcal{B} = (B_1, \dots, B_n)$ are such that $A_i = \{g_e \in G \mid \exists j \in N,\ e = (v_i, v_j) \in E\}$ and $B_i = \{g_e \in G \mid \exists j \in N,\ e = (v_j, v_i) \in E\}$ for each $i \in N$. Note that this induces the graph $\mathcal{G}_{\mathcal{A},\mathcal{B}}$ isomorphic to $\widetilde{\mathcal{G}}$. The distance between $\mathcal{A}$ and $\mathcal{B}$ is $|E| - c^*_{\mathcal{A},\mathcal{B}}$ by Proposition 4.4.1. Therefore, if we can find this distance, then we can find $c^*_{\mathcal{A},\mathcal{B}}$, solving the problem instance from Lemma 4.4.3. $\square$

4.4.1 General Utilities

We now discuss properties of the EF1 exchange graph. The following result demonstrates that deciding whether an EF1 exchange path exists is a PSPACE-complete problem.

Theorem 4.4.5. *Deciding the existence of an EF1 exchange path between two EF1 allocations is PSPACE-complete.*

Proof. First, we show membership in PSPACE—recall that PSPACE is the set of all decision problems that can be solved by a deterministic polynomial-space Turing machine. We can solve the problem non-deterministically by simply guessing an EF1 exchange path between the two EF1 allocations. Since the total number of allocations is at most n^m, if there exists an EF1 exchange path between the two allocations, then there exists one with length at most n^m; such a path can be verified in polynomial space (i.e., using a polynomial number of bits). This shows that the problem is in NPSPACE, the set of all decision problems that can be solved by a non-deterministic polynomial-space Turing machine. By Savitch's Theorem, NPSPACE $\subseteq$ PSPACE (Savitch, 1970), which implies that this problem is in PSPACE.

To prove that our problem is PSPACE-hard, we shall reduce the Perfect Matching Reconfiguration problem for a balanced (undirected) bipartite graph to our problem. Recall that the Perfect Matching Reconfiguration problem is the task of deciding if two perfect matchings of a balanced bipartite graph can be reached from each other via a sequence of *flips*, i.e., given perfect matchings $\widetilde{M}_0$ and $\widetilde{M}$ of a balanced bipartite graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$, whether there exists a sequence of perfect matchings $\widetilde{M}_0, \widetilde{M}_1, \ldots, \widetilde{M}_T$ such that

- $\widetilde{M}_T = \widetilde{M}$, and
- for each $t \in \{1, \ldots, T\}$, there exist edges $\widetilde{e}_t^1, \widetilde{e}_t^2, \widetilde{e}_t^3, \widetilde{e}_t^4$ of $\widetilde{G}$ such that $\widetilde{M}_{t-1} \setminus \widetilde{M}_t = \{\widetilde{e}_t^1, \widetilde{e}_t^3\}$, $\widetilde{M}_t \setminus \widetilde{M}_{t-1} = \{\widetilde{e}_t^2, \widetilde{e}_t^4\}$, and $\widetilde{e}_t^1\widetilde{e}_t^2\widetilde{e}_t^3\widetilde{e}_t^4$ forms a cycle.

The operation from $\widetilde{M}_{t-1}$ to $\widetilde{M}_t$ is called a *flip*, and we say that $\widetilde{M}_{t-1}$ and $\widetilde{M}_t$ are *adjacent* to each other. Perfect Matching Reconfiguration is known to be PSPACE-hard (Bonamy et al., 2019). Let $|\widetilde{V}| = 2v$, and let the two independent sets of $\widetilde{G}$ be $\widetilde{P} = \{\widetilde{p}_1, \ldots, \widetilde{p}_v\}$ and $\widetilde{Q} = \{\widetilde{q}_1, \ldots, \widetilde{q}_v\}$. For each $i \in \{1, \ldots, v\}$, let $\widetilde{q}_{k_i} \in \widetilde{Q}$ be the vertex adjacent to $\widetilde{p}_i$ in $\widetilde{M}_0$, and let $\widetilde{q}_{\ell_i} \in \widetilde{Q}$ be the vertex adjacent to $\widetilde{p}_i$ in $\widetilde{M}$. We shall show that this problem instance can be reduced to an instance of deciding the existence of an EF1 exchange path between two EF1 allocations.

Define an instance of the EF1 exchange path problem as follows: let $N = \{0, 1, \ldots, v\}$ be the set of agents, $G = \{p_1, \ldots, p_v, q_1, \ldots, q_v, r_1, r_2, r_3, r_4\}$ be the set of goods, and the utility function of each agent be

- $u_0(g) = 0$ for all $g \in G$, and
- for $i \in \{1, \ldots, v\}$,

$$u_i(g) = \begin{cases} 3 & \text{if } g \in \{p_i\} \cup \{q_k \mid \{\widetilde{p}_i, \widetilde{q}_k\} \in \widetilde{E}\}; \\ 2 & \text{if } g \in \{r_1, r_2, r_3, r_4\}; \\ 0 & \text{otherwise.} \end{cases}$$

In the initial allocation $\mathcal{A}_0$, agent 0 has the bundle $\{r_1, r_2, r_3, r_4\}$ and agent i has the bundle $\{p_i, q_{k_i}\}$ for each $i \in \{1, \dots, v\}$. In the target allocation $\mathcal{A}$, agent 0 again has the bundle $\{r_1, r_2, r_3, r_4\}$ and agent i has the bundle $\{p_i, q_{\ell_i}\}$ for each $i \in \{1, \dots, v\}$. Observe that both allocations are EF1—agent 0 assigns zero utility to every bundle, while each agent $i \in \{1, \dots, v\}$ assigns a utility of 6 to her own bundle, a utility of at most 6 to the bundle of every agent in $\{1, \dots, v\} \setminus \{i\}$, and a utility of $6+2$ to agent 0's bundle. Clearly, this instance can be constructed in polynomial time.

Suppose first that there exists a sequence of adjacent perfect matchings from $\widetilde{M}_0$ to $\widetilde{M}$. Then each flip from $\widetilde{M}_{t-1}$ to $\widetilde{M}_t$ corresponds to an exchange in the EF1 exchange path problem: if $\widetilde{M}_{t-1} \setminus \widetilde{M}_t = \{\{\widetilde{p}_i, \widetilde{q}_k\}, \{\widetilde{p}_j, \widetilde{q}_\ell\}\}$ and $\widetilde{M}_t \setminus \widetilde{M}_{t-1} = \{\{\widetilde{p}_i, \widetilde{q}_\ell\}, \{\widetilde{p}_j, \widetilde{q}_k\}\}$, then exchange q_k in agent i's bundle with q_ℓ in agent j's bundle. The new allocation is also EF1—as before, agent 0 assigns zero utility to every bundle, while each agent $i' \in \{1, \dots, v\}$ assigns a utility of 6 to her own bundle, a utility of at most 6 to the bundle of every agent in $\{1, \dots, v\} \setminus \{i'\}$, and a utility of $6+2$ to agent 0's bundle. By performing the exchanges according to the flips in sequence, we reach the target allocation. Therefore, an EF1 exchange path exists.

Conversely, assume that an EF1 exchange path exists between the initial allocation $\mathcal{A}_0$ and the target allocation $\mathcal{A}$. Consider the sequence of EF1 allocations $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_T = \mathcal{A}$. We show by induction that for every intermediate allocation $\mathcal{A}_t$, every agent $i \in \{1, \dots, v\}$ assigns a utility of 6 to her own bundle (consisting of p_i and q_k for some k), and agent 0 retains $\{r_1, r_2, r_3, r_4\}$. The base case $t = 0$ is trivial. For the inductive case, suppose that the statement is true for $t - 1$. If some agent $i \in \{1, \dots, v\}$ attempts to exchange one of her goods with a good from agent 0, then agent i's new bundle has utility 5 but agent 0's new bundle has utility $6 + 3$ for agent i, which violates EF1. Therefore, agent i must exchange goods with agent j for some $j \in \{1, \dots, v\}$. Note that agent 0's bundle is worth $6+2$ to agent i and j, so agent i's and j's own bundles must be worth at least 6 to i and j respectively. If agent i gives p_i to agent j, then agent j's new bundle consists of p_i (worth zero to her) and some q_ℓ, which is worth at most 3 to her—this violates EF1. By the same reasoning, agent j cannot give p_j to agent i. Therefore, they must exchange q_k in agent i's bundle with q_ℓ in agent j's bundle. As agent i and j must have bundles worth at least 6 to each of them, q_ℓ must be worth 3 to agent i and q_k must be worth 3 to agent j. This completes the induction.

As a result, the perfect matchings $\widetilde{M}_{t-1}$ (corresponding to $\mathcal{A}_{t-1}$) and $\widetilde{M}_t$ (corresponding to $\mathcal{A}_t$) must be adjacent to each other for all t, where $\widetilde{M}_{t-1} \setminus \widetilde{M}_t = \{\{\widetilde{p}_i, \widetilde{q}_k\}, \{\widetilde{p}_j, \widetilde{q}_\ell\}\}$ and $\widetilde{M}_t \setminus \widetilde{M}_{t-1} = \{\{\widetilde{p}_i, \widetilde{q}_\ell\}, \{\widetilde{p}_j, \widetilde{q}_k\}\}$. It follows that a sequence of adjacent perfect matchings $\widetilde{M}_0, \widetilde{M}_1, \dots, \widetilde{M}_T = \widetilde{M}$ indeed exists. This completes the proof. □

Regarding the existence of *optimal* EF1 exchange paths, we shall show later in Theorem 4.4.10 that the corresponding decision problem is NP-hard even for four agents with identical utilities.

4.4.2 Identical Binary Utilities

We now consider the most restrictive class of utility functions in this chapter: those that are identical *and* binary. We show that the EF1 exchange graph is connected for any number of agents with such utility functions.

Theorem 4.4.6. *Let an instance with $n \geq 2$ agents and identical binary utility functions be given. Then, the EF1 exchange graph is connected. Moreover, an EF1 exchange path between any two allocations can be found in polynomial time.*

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be two EF1 allocations. Since every good is worth either 1 or 0 to every agent, every agent's bundle in $\mathcal{A}$ and $\mathcal{B}$ must have a utility of either $\lfloor u(G)/n \rfloor$ or $\lfloor u(G)/n \rfloor + 1$ (otherwise, the gap between the utilities of some two agents' bundles is at least 2, and the corresponding allocation is not EF1). Let N' be the set of agents whose bundles in $\mathcal{A}$ and $\mathcal{B}$ have different utilities. Note that half of the agents in N' have bundles worth $\lfloor u(G)/n \rfloor$ in $\mathcal{A}$ and $\lfloor u(G)/n \rfloor + 1$ in $\mathcal{B}$; the other half have bundles worth $\lfloor u(G)/n \rfloor + 1$ in $\mathcal{A}$ and $\lfloor u(G)/n \rfloor$ in $\mathcal{B}$. If $N' \neq \varnothing$, let agent $i \in N'$ be an agent with a bundle worth $\lfloor u(G)/n \rfloor$ in $\mathcal{A}$, and let g_i be a good with utility 0 in A_i—this good exists because agent i has at least $\lfloor u(G)/n \rfloor + 1$ goods in her bundle (due to B_i's utility of $\lfloor u(G)/n \rfloor + 1$) but only has utility $\lfloor u(G)/n \rfloor$ in A_i. Let agent $j \in N'$ be an agent with a bundle worth $\lfloor u(G)/n \rfloor + 1$ in $\mathcal{A}$, and let g_j be a good with utility 1 in A_j—this good exists because A_j has utility at least 1. Exchange g_i with g_j; it can be verified that the resulting allocation is EF1. As this exchange reduces the size of the set N' by two, we can repeatedly make such exchanges between two agents in N' until $N' = \varnothing$. Note that such exchanges can be performed in polynomial time.

At this point, we have shown that there exists an EF1 allocation $\mathcal{A}'$ such that an EF1 exchange path exists between $\mathcal{A}$ and $\mathcal{A}'$, and for every agent i, her bundles in $\mathcal{A}'$ and $\mathcal{B}$ have the same utility. Define the item graph $\mathcal{G}_{\mathcal{A}',\mathcal{B}}$ as in the beginning of Section 4.4, and consider its subgraph with only the edges representing the goods with utility 1. For each agent, the indegree and the outdegree of the corresponding vertex in this subgraph are equal, so we can perform exchanges to 'resolve' these edges. Specifically, suppose there is an edge $e_x = (i, j)$ corresponding to a good g_x, where $i \neq j$. By the degree condition, there must exist another edge $e_y = (j, k)$ corresponding to a good g_y, where $j \neq k$ but possibly $k = i$. We let agents i and j exchange g_x and g_y, so g_x is now with its correct owner, agent j. Hence, at least one more good goes to the correct agent after the exchange. This exchange process can be performed in polynomial time, and no agent's utility changes during the process, which means that the intermediate allocations are all EF1. Similarly, if we consider the subgraph with only the edges representing the goods with utility 0, we can perform exchanges to resolve these edges as well. Therefore, there exists an EF1 exchange path from $\mathcal{A}'$ to $\mathcal{B}$, and thus an EF1 exchange path from $\mathcal{A}$ to $\mathcal{B}$, and this path can be found in polynomial time. □

In spite of this positive result, the polynomial-time algorithm described in Theorem 4.4.6 does not necessarily find an optimal EF1 exchange path between the two allocations. In fact, even for the special case where the EF1 exchange graph $\mathcal{G}^{\text{EF1}}$ and the exchange graph $\mathcal{G}$ coincide (e.g., when every agent assigns zero utility to every good, so every allocation is EF1), it is NP-hard to compute an optimal EF1 exchange path by Theorem 4.4.4, regardless of whether optimality refers to the length of the shortest path in $\mathcal{G}$ or in $\mathcal{G}^{\text{EF1}}$. Hence, a polynomial-time algorithm for this task does not exist unless P = NP. Moreover, we show next that, somewhat surprisingly, an optimal EF1 exchange path (with respect to $\mathcal{G}$) is not guaranteed to exist even for identical binary utilities.

Theorem 4.4.7. *For each $n \geq 3$, there exists an instance with n agents with identical binary utility functions satisfying the following properties: the EF1 exchange graph is connected, but for some pair of EF1 allocations, no optimal EF1 exchange path exists between them.*

Proof. For $n = 3$ agents, consider $\vec{s} = (2, 2, 2)$ and the utility of the goods as follows:

g	g_1	g_2	g_3	g_4	g_5	g_6
$u(g)$	1	1	1	0	0	0

Note that the EF1 exchange graph is connected by Theorem 4.4.6. We prove that an optimal EF1 exchange path between $\mathcal{A}$ and $\mathcal{B}$ does not exist, where $A_1 = \{g_2, g_6\}$, $A_2 = \{g_3, g_4\}$, $A_3 = \{g_1, g_5\}$, and $B_i = \{g_i, g_{i+3}\}$ for $i \in \{1, 2, 3\}$—it can be verified that both $\mathcal{A}$ and $\mathcal{B}$ are EF1, and the distance between $\mathcal{A}$ and $\mathcal{B}$ is 3 (through exchanging $g_1 \leftrightarrow g_6$, $g_2 \leftrightarrow g_4$, and $g_3 \leftrightarrow g_5$). Consider any EF1 exchange path between $\mathcal{A}$ and $\mathcal{B}$, and let $\mathcal{A}'$ be the EF1 allocation adjacent to $\mathcal{A}$ on the exchange path. If a valuable good (g_1, g_2, or g_3) is exchanged with a non-valuable good (g_4, g_5, or g_6) from $\mathcal{A}$ to reach $\mathcal{A}'$, then one agent has utility 0 and another agent has utility 2, which means that $\mathcal{A}'$ is not EF1. Therefore, the only exchanges possible from $\mathcal{A}$ are between valuable goods or between non-valuable goods. However, any of these exchanges causes at most one good to go to the correct bundle according to $\mathcal{B}$, so there are at least five goods in $\mathcal{A}'$ in the wrong bundle according to $\mathcal{B}$. As any exchange of goods reduces the number of goods in the wrong bundle by at most two, the distance between $\mathcal{A}'$ and $\mathcal{B}$ is at least 3. This means that the distance between $\mathcal{A}$ and $\mathcal{B}$ is at least 4. It follows that no optimal EF1 exchange path exists between $\mathcal{A}$ and $\mathcal{B}$.

For $n > 3$ agents, simply add $n - 3$ dummy agents who have the same utility function as the three original agents and have empty bundles. □

4.4.3 Binary Utilities

We saw in Theorem 4.4.6 that the EF1 exchange graph is always connected for any number of agents with identical binary utilities. Now, we consider the case where the agents have binary utilities which may differ between agents. It turns out that the EF1 exchange graph is not necessarily connected in this case, even when there are three agents. This also provides a contrast to the case of two agents (Theorem 4.3.4).

Theorem 4.4.8. *For each $n \geq 3$, there exists an instance with n agents with binary utility functions such that the EF1 exchange graph is disconnected.*

Proof. For $n = 3$ agents, consider the utility of the goods as follows:

g	g_1	g_2	g_3	g_4
$u_1(g)$	1	0	1	0
$u_2(g)$	1	0	1	0
$u_3(g)$	0	1	1	0

Let $\mathcal{A}$ and $\mathcal{B}$ be given such that $A_1 = \{g_1, g_2\}$, $A_2 = \{g_3, g_4\}$, $B_1 = \{g_3, g_4\}$, $B_2 = \{g_1, g_2\}$, and $A_3 = B_3 = \varnothing$—it can be verified that both $\mathcal{A}$ and $\mathcal{B}$ are EF1. Consider any EF1 exchange path between $\mathcal{A}$ and $\mathcal{B}$, and let $\mathcal{A}'$ be the EF1 allocation adjacent to $\mathcal{A}$ on the exchange path. We claim that $\mathcal{A}'$ cannot exist. The only possible exchanges from $\mathcal{A}$ are $g_i \leftrightarrow g_{5-i}$ or $g_i \leftrightarrow g_{i+2}$ for some $i \in \{1, 2\}$. If g_i is exchanged with g_{5-i}, then agent i's bundle has zero utility from agent i's perspective while agent $(3-i)$'s bundle has utility 2 from agent i's perspective, so agent i envies agent $(3 - i)$ by more than one good. On the other hand, if g_i is exchanged with g_{i+2}, then agent i's bundle has utility 2 from agent 3's perspective, and since agent 3 has an empty bundle, agent 3 envies agent i by more than one good. Therefore, $\mathcal{A}'$ does not exist, which contradicts the assumption that the path is an EF1 exchange path. It follows that no EF1 exchange path exists between $\mathcal{A}$ and $\mathcal{B}$.

For $n > 3$ agents, simply add dummy agents who assign zero value to every good and have empty bundles. □

4.4.4 Identical Utilities

Let us now consider the case where the utilities are identical across agents, though they need not be binary. As with the case of binary utilities, there are instances in which the EF1 exchange graph is not connected even for three agents.

Theorem 4.4.9. *For each $n \geq 3$, there exists an instance with n agents with identical utility functions such that the EF1 exchange graph is disconnected.*

Proof. For $n = 3$ agents, consider the utility of the goods as follows:

g	g_1	g_2	g_3	g_4	g_5	g_6	g_7
$u(g)$	4	3	1	4	2	2	4

Let $\mathcal{A}$ and $\mathcal{B}$ be given such that $A_1 = \{g_1, g_2, g_3\}$, $A_2 = \{g_4, g_5, g_6\}$, $B_1 = \{g_1, g_5, g_6\}$, $B_2 = \{g_2, g_3, g_4\}$, and $A_3 = B_3 = \{g_7\}$. It can be verified that both $\mathcal{A}$ and $\mathcal{B}$ are EF1. Consider any exchange path from $\mathcal{A}$ to $\mathcal{B}$. At some point, a good $g \in \{g_2, g_3, g_5, g_6\}$ has to be exchanged, but this will inevitably cause agent 3 to envy agent 1 or agent 2 by more than one good, so the exchange path cannot be EF1. Therefore, no EF1 exchange path exists between $\mathcal{A}$ and $\mathcal{B}$.

For $n > 3$ agents, simply add $n-3$ agents who have the same utility function as the three original agents and $n - 3$ goods with value 4 each, and allocate each of these goods to one of these $n - 3$ agents in both $\mathcal{A}$ and $\mathcal{B}$. □

We end this section with a result that determining whether an optimal EF1 exchange path exists between two allocations is NP-hard even for four agents with identical valuations. This can be shown via a reduction from the NP-hard problem Partition.

Theorem 4.4.10. *Deciding the existence of an optimal EF1 exchange path between two EF1 allocations is NP-hard, even for $n = 4$ agents with identical utility functions.*

Proof. We shall reduce the NP-hard problem Partition to this problem. Recall that the Partition problem is the task of deciding whether a multiset $X = \{x_1, \ldots, x_q\}$ of positive integers can be partitioned into two subsets such that the sum of the integers in one subset is equal to that in the other subset. Let the sum of all the integers in X be $2S$.

Define an instance of the EF1 exchange path problem with $n = 4$ agents and the set of goods $G = \{a_0, a_1, \ldots, a_q, b_0, b_1, \ldots, b_q, c_1, c_2, d_1, d_2\}$. The utility of each good is defined as follows:

- $u(a_0) = u(b_0) = u(c_1) = 2S$,
- $u(d_1) = u(d_2) = S$,
- $u(a_i) = x_i$ for all $i \in \{1, \ldots, q\}$,
- $u(b_i) = u(c_2) = 0$ for all $i \in \{1, \ldots, q\}$.

The initial allocation $\mathcal{A} = (A_1, A_2, A_3, A_4)$ and the target allocation $\mathcal{B} = (B_1, B_2, B_3, B_4)$ are given by $A_1 = \{a_0, a_1, \ldots, a_q\}$, $A_2 = \{b_0, b_1, \ldots, b_q\}$, $B_1 = \{a_0, b_1, \ldots, b_q\}$, $B_2 = \{b_0, a_1, \ldots, a_q\}$, $A_3 = B_4 = \{c_1, c_2\}$, and $A_4 = B_3 = \{d_1, d_2\}$—it can be verified that both $\mathcal{A}$ and $\mathcal{B}$ are EF1. Note that this instance can be constructed in polynomial time, and the distance between $\mathcal{A}$ and $\mathcal{B}$ is $q + 2$.

First, suppose that X can be partitioned into two subsets of sum S each, say $\{x_{i_1}, \ldots, x_{i_\ell}\}$ has sum S. We perform the following exchanges starting from $\mathcal{A}$:

- First, exchange $\{a_{i_1}, \ldots, a_{i_\ell}\}$ with $\{b_{i_1}, \ldots, b_{i_\ell}\}$ pair-by-pair. At this point, agents 1 and 2 have bundles worth $3S$ each, and agents 3 and 4 have bundles worth $2S$ each.
- Next, exchange c_1 with d_1, and exchange c_2 with d_2.
- Finally, exchange $\{a_1, \ldots, a_q\} \setminus \{a_{i_1}, \ldots, a_{i_\ell}\}$ with $\{b_1, \ldots, b_q\} \setminus \{b_{i_1}, \ldots, b_{i_\ell}\}$ pair-by-pair.

The allocation resulting from this sequence of exchanges is $\mathcal{B}$. It can be verified that this exchange path has length $q + 2$ and every intermediate allocation is EF1. Hence, there exists an optimal EF1 exchange path between $\mathcal{A}$ and $\mathcal{B}$.

Conversely, suppose that there exists an optimal EF1 exchange path between $\mathcal{A}$ and $\mathcal{B}$. The only exchanges possible are $a_i \leftrightarrow b_j$ for some $i, j \in \{1, \ldots, q\}$ and $c_i \leftrightarrow d_j$ for some $i, j \in \{1, 2\}$. In particular, at some point, c_i must be exchanged with d_j for the first time for some $i, j \in \{1, 2\}$. Consider the allocation following this exchange. One of agents 3 and 4 now has utility only S. By assumption, this allocation is EF1, so this agent does not envy agent 1 and agent 2. Removing the highest-utility good from agent 1's and agent 2's bundle (i.e., a_0 and b_0), the utility of each of the remaining bundles must be at most S. The only way this is possible is that $\{a_1, \ldots, a_q\}$ can be partitioned into two subsets such that the utility of each subset is exactly S, and each of agents 1 and 2 receives exactly one of those subsets. Correspondingly, this shows that X can be partitioned into two subsets of sum S each. □

4.5 Conclusion

In this chapter, we have initiated the study of reachability problems in fair division by investigating the connectivity of the EF1 exchange graph and the optimality of EF1 exchange paths. We showed that even for two agents, an EF1 exchange path between two given EF1 allocations does not necessarily exist. On the positive side, such a path always exists if both agents have identical or binary utility functions—in these cases, we can also ensure an optimal path regardless of EF1 considerations, and the path can be found in polynomial time. For three or more agents, however, the problem becomes much less tractable, both in terms of existence and computation. In particular, we proved that finding the smallest number of exchanges between two allocations is NP-hard even if we were to ignore the EF1 constraints, and deciding whether an EF1 exchange path between two allocations exists is PSPACE-complete. Moreover, the existence of an EF1 exchange path cannot be guaranteed even if the utilities are identical *or* binary, although such a guarantee is possible if the utilities are identical *and* binary.

This chapter leaves several questions and directions for future research. Firstly, while determining the existence of an EF1 exchange path between two given allocations is PSPACE-complete in general, an intriguing question is whether this can be done in polynomial time for two agents. One could also ask whether *near*-optimal EF1 exchange paths are possible for general utilities or for three or more agents. In addition, for the negative results obtained in this chapter, one could ask whether an (optimal or otherwise) exchange path between EF1 allocations exists if we allow the intermediate allocations to be *envy-free up to k goods (EFk)* for some small $k > 1$. Extending our results to fairness notions other than EF1 is also a meaningful direction. Finally, in addition to (or instead of) exchanging goods between agents, one may also consider the setting where an agent *transfers* one good to another agent in each operation—in this case, the size of the allocation does not need to be fixed.

Chapter 5

Reforming an Unfair Allocation by Exchanging Goods

5.1 Introduction

The fair division literature typically assumes that there is a set of unallocated goods and the objective is to allocate them fairly. In certain scenarios, however, an existing (possibly unfair) allocation of the goods is already in place, and the goal is to "reform" it in order to arrive at a fair allocation. This is the case, for instance, when new office occupants or dormitory residents move in at the beginning of an academic year—their preferences over the books, paintings, or furniture in their rooms may well differ from those of their predecessors, which makes the existing allocation unfair from their perspective. Moreover, even when the ownership of goods remains unchanged, the owners' preferences can still undergo changes over time. For example, consider the distribution of personnel among teams in an organization. As the personnel experience individual growth or decline, and as the needs of the teams evolve, these changes can necessitate a reevaluation and potential reformation of the current distribution.

In this chapter, we shall allow agents to *exchange* pairs of goods in the reformation process, and use envy-freeness up to one good (EF1) as our fairness criterion. Exchanges preserve the size of each agent's bundle, thereby ensuring that any cardinality constraints remain fulfilled. Note that capacity constraints are prevalent in fair division applications and have accordingly received interest in the literature (Biswas and Barman, 2018; Wu et al., 2021; Hummel and Hetland, 2022; Shoshan et al., 2023). Naturally, given an initial allocation, we wish to reach an EF1 allocation using a small number of exchanges. However, it is sometimes impossible to reach an EF1 allocation via *any* finite number of exchanges, so we start by exploring whether the corresponding decision problem can be answered efficiently. Since this problem is equivalent to determining whether an EF1 allocation with a certain size vector exists in a given instance, it is meaningful independently of exchange considerations.[1] We also investigate other fundamental questions in this setting. Namely, if it is possible to reach

[1]When an EF1 allocation is not guaranteed to exist in all instances due to cardinality requirements, an approach taken by previous work is to relax the EF1 condition (e.g., Wu et al., 2021). However, this leads to unnecessarily weak guarantees in instances where EF1 can be attained.

an EF1 allocation, can we efficiently determine the smallest number of exchanges required to achieve this goal? And how many exchanges might we need to make in the worst case in order to attain EF1?

5.1.1 Our Results

In our model, there is an initial allocation of a set of goods. As is often assumed in fair division, each agent has an additive utility function over the goods. At each step of the reformation process, two agents can exchange a pair of goods with each other to obtain another allocation, and the goal is to reach an EF1 allocation at the end of the process. More details on our model are provided in Section 5.2.

In Section 5.3, we consider the decision problem of determining whether a given initial allocation can be reformed into an EF1 allocation. As mentioned earlier, this problem is equivalent to determining whether an EF1 allocation with a given size vector exists, so we focus on the latter decision problem instead. We demonstrate interesting distinctions in the complexity based on the number of agents and their utility functions. Specifically, in the case of two agents, the problem can be solved in polynomial time if the agents have identical utilities, but becomes (weakly) NP-hard otherwise. For three or more agents, the problem is NP-hard even with identical utilities; however, it can be solved efficiently when the utilities are binary provided that the number of agents is constant. Finally, for an arbitrary (non-constant) number of agents, the problem is strongly NP-hard even for identical *or* binary utilities, but can be solved in polynomial time if the utilities are identical *and* binary. The results of this section are summarized in Table 5.1.

utilities	general	identical	binary	identical binary
$n = 2$	wNP-c (Th. 5.3.3)	P (Th. 5.3.2)	P (Th. 5.3.7)	P (Th. 5.3.7)
constant $n \geq 3$	wNP-c (Th. 5.3.5)	wNP-c (Th. 5.3.5)	P (Th. 5.3.7)	P (Th. 5.3.7)
general n	sNP-c (Th. 5.3.8)	sNP-c (Th. 5.3.8)	sNP-c (Th. 5.3.9)	P (Th. 5.3.10)

Table 5.1: Computational complexity of REFORMABILITY, or equivalently, deciding whether an EF1 allocation with a given size vector exists in a given instance. The top row represents the class of utility functions considered. The leftmost column represents the number of agents. "sNP-c" and "wNP-c" stand for strongly NP-complete and weakly NP-complete, respectively.

Having determined the reformability of the initial allocation, we next explore the problem of computing the optimal (i.e., minimum) number of exchanges required to reach an EF1 allocation in Section 5.4. For (a) two agents with identical utilities, (b) a constant number of agents with binary utilities, and (c) any number of agents with identical binary utilities, we show that this computational problem can be solved in polynomial time, just like the corresponding decision problem in Section 5.3. For the remaining scenarios, since deciding whether an allocation is reformable is already NP-hard (from Section 5.3), we instead focus on the special case where the allocation is balanced—an EF1 allocation is guaranteed to be reachable in this case (see Proposition 5.2.1). We show that the computational problem for this special case remains NP-hard. The results of this section are summarized in Table 5.2.

Finally, in Section 5.5, instead of considering specific instances, we derive worst-case bounds on the number of exchanges necessary in order to reach an EF1 allocation. We

utilities	general	identical	binary	identical binary
$n = 2$	NP-h (Th. 5.4.2)	P (Th. 5.4.1)	P (Th. 5.4.5)	P (Th. 5.4.5)
constant $n \geq 3$	NP-h (Th. 5.4.3)	NP-h (Th. 5.4.3)	P (Th. 5.4.5)	P (Th. 5.4.5)
general n	NP-h (Th. 5.4.3)	NP-h (Th. 5.4.3)	NP-h (Th. 5.4.6)	P (Th. 5.4.7)

Table 5.2: Computational complexity of Optimal Exchanges. The top row represents the class of utility functions considered. The leftmost column represents the number of agents. "NP-h" stands for NP-hard.

assume that each of the n agents possesses s goods—this again ensures that an EF1 allocation is reachable. We show that roughly $s(n-1)/2$ exchanges always suffice. Moreover, our bound is essentially tight for any n and s, and exactly tight when $n = 2$ as well as when s is divisible by n.

5.1.2 Related Work

As mentioned earlier, the majority of work in fair division assumes that there is no initial allocation of the resources—we now discuss the key exceptions and their differences from our model. Boehmer et al. (2022) studied the problem of *discarding* goods from an initial allocation in order to reach an envy-free or EF1 allocation. As it is possible to reallocate the goods in several practical situations, discarding them can be unnecessarily wasteful for the agents involved. In a similar vein, Dorn et al. (2021) investigated deleting goods to attain another fairness notion called *proportionality*; they assumed that agents have ordinal preferences (rather than cardinal utilities) over the goods, and considered both the settings with and without an initial allocation.[2] Aziz et al. (2019) focused on reallocating goods to make agents better off, but did not delve into the aspect of fairness. Chapter 4 aimed to transition from an initial allocation to a target allocation, both of which are EF1, while maintaining EF1 throughout the process. Segal-Halevi (2022) considered the reallocation of a *divisible* good and explored the trade-off between guaranteeing a minimum utility for every agent and ensuring each agent a certain fraction of her original utility.

Further afield, the idea of improving an initial allocation has also been examined when each agent receives only one good, a setting sometimes known as a *housing market*. Gourvès et al. (2017) assumed an underlying social network and allowed beneficial exchanges between agents who are neighbors in the network; their work led to a series of follow-up papers on similar models (Huang and Xiao, 2020; Li et al., 2021; Müller and Bentert, 2021; Ito et al., 2023). Note that these papers did not take fairness considerations into account. Ito et al. (2022) incorporated fairness in the form of envy-freeness into this setting—starting with an envy-free allocation, they let each agent exchange her current good with a preferred unassigned good as long as the exchange keeps the allocation envy-free.

[2]When there is no initial allocation, Dorn et al. (2021) considered deleting goods so that a proportional allocation of the remaining goods exists. In an earlier work, Aziz et al. (2016) examined discarding or adding goods to achieve envy-freeness, also in the absence of an initial allocation and under ordinal preferences.

5.2 Preliminaries

Refer to the preliminaries in Sections 2.1 and 2.2. We now describe other preliminaries specific to this chapter.

A size vector $\vec{s} = (s_1, \ldots, s_n)$ is *balanced* if $|s_i - s_j| \leq 1$ for all $i, j \in N$, and an allocation is *balanced* if it has a balanced size vector. We use $\mathcal{I}$ to denote a fair division instance.

Given an instance and two allocations $\mathcal{A}$ and $\mathcal{B}$ in the instance, we say that $\mathcal{A}$ can be *reformed* into $\mathcal{B}$ if there exists an exchange path between the two allocations on the exchange graph $\mathcal{G}$. The *optimal number of exchanges* required to reach $\mathcal{B}$ from $\mathcal{A}$ is the distance between $\mathcal{A}$ and $\mathcal{B}$ on the exchange graph $\mathcal{G}$.

We state a simple proposition that characterizes the existence of EF1 allocations based on the size vector.

Proposition 5.2.1. *Let $\vec{s} = (s_1, \ldots, s_n)$, and let $m = \sum_{i=1}^{n} s_i$.*

(a) If $\vec{s}$ is balanced, then every instance with n agents and m goods admits an EF1 allocation with size vector $\vec{s}$.

(b) If $\vec{s}$ is not balanced, then there exists an instance with n agents and m goods that does not admit any EF1 allocation with size vector $\vec{s}$.

Proof. (a) An EF1 allocation can be guaranteed by allowing agents to pick their favorite goods in a round-robin fashion, with agents with higher s_i (if any) starting before those with lower s_i, until each agent i has s_i goods.

(b) Let $\mathcal{I}$ be an instance with m goods such that $u_i(g) = 1$ for all $i \in N$ and $g \in G$. Let $\mathcal{A}$ be any allocation with size vector $\vec{s}$. Since $\vec{s}$ is not balanced, there exist distinct $i, j \in N$ such that $s_j - s_i \geq 2$. Then, we have $|A_j| = s_j > 0$, so $A_j \neq \varnothing$. Furthermore, $u_i(A_i) = s_i < s_j - 1 = u_i(A_j \setminus \{g\})$ for all $g \in A_j$. This shows that agent i is not EF1 towards agent j, and so $\mathcal{A}$ is not EF1. Therefore, $\mathcal{I}$ does not admit an EF1 allocation with size vector $\vec{s}$. □

Finally, we introduce an NP-hard decision problem called the Balanced Multi-Partition problem, which we will use later in the proofs of several results. In Balanced Multi-Partition, we are given positive integers p, q, K and a multiset of positive integers $X = \{x_1, \ldots, x_{pq}\}$ such that $K < x_j \leq 2K$ for all $j \in \{1, \ldots, pq\}$, and the sum of all the integers in X is $p(q+1)K$. The problem is to decide whether X can be partitioned into multisets $X_1, \ldots, X_p$ of equal cardinalities and sums, i.e., for each $i \in \{1, \ldots, p\}$, the cardinality of X_i is q, and the sum of all the integers in X_i is $(q+1)K$. The NP-hardness of this problem is based on a reduction from the equal-cardinality version of the NP-hard problem Partition (Garey and Johnson, 1979, p. 223).

Proposition 5.2.2. *For any fixed $p \geq 2$, Balanced Multi-Partition is NP-hard.*

Proof. We shall prove NP-hardness via a series of reductions from the equal-cardinality version of Partition. In this version, we are given positive integers q, K' and a multiset of positive integers $W = \{w_1, \ldots, w_{2q}\}$ such that the sum of the integers in W is $2K'$. The problem is to decide whether W can be partitioned into multisets W_1 and W_2 of equal cardinalities and sums. This problem is known to be NP-hard (Garey and Johnson, 1979, p. 223).

Let an instance of the equal-cardinality version of PARTITION be given, and let $p \geq 2$ be a fixed integer. If some integer in W is more than K', then W cannot be partitioned into the desired multisets; therefore, we assume that every integer in W is at most K'. Define a multiset $W^1 = \{w_j \mid j \in \{2q+1, \ldots, 2q+(p-2)\}\}$ such that $w_j = K'$ for all $w_j \in W^1$; define a multiset $W^0 = \{w_j \mid j \in \{2q+(p-2)+1, \ldots, pq\}\}$ such that $w_j = 0$ for all $w_j \in W^0$; and define $W' = W \cup W^1 \cup W^0$. Essentially, we are adding $p-2$ copies of the number K' and sufficiently many copies of the number 0 so that the total number of elements in W' is pq. Note that every integer in W' is at most K', and the sum of all the elements in W' is pK'. We claim that W' can be partitioned into multisets $W'_1, \ldots, W'_p$ of equal cardinalities and sums (i.e., each W'_i has cardinality q and sum K') if and only if W can be partitioned into multisets W_1 and W_2 of equal cardinalities and sums.

($\Leftarrow$) If we are given a partition into multisets W_1 and W_2, let $W'_1 = W_1$, let $W'_2 = W_2$, and let W'_i contain one element from W^1 and $q-1$ elements from W^0 for each $i \in \{3, \ldots, p\}$. Each of $W'_1, \ldots, W'_p$ has q elements with sum K'. This gives a desired partition of W'.

($\Rightarrow$) Assume that we are given a partition into multisets $W'_1, \ldots, W'_p$. If some W'_i contains at least two elements in W^1, then the sum of W'_i is more than K', which is not possible. Therefore, every W'_i contains at most one element in W^1. Furthermore, for each W'_i containing some element in W^1, if it contains some element in W, then its sum would exceed K', which is again not possible. Therefore, there are $p-2$ of the W'_i such that each of them contains one element from W^1 and $q-1$ elements from W^0. This means that two of the W'_i contain exactly the elements in W. These two W'_i induce the desired partition into W_1 and W_2 of W.

Now, define an instance of BALANCED MULTI-PARTITION as follows. Let $K = K' + q$, and let $X = \{x_1, \ldots, x_{pq}\}$ be such that $x_j = w_j + K + 1$ for all $j \in \{1, \ldots, pq\}$. For each $j \in \{1, \ldots, pq\}$, since $0 \leq w_j \leq K'$, we have $K < x_j \leq K' + K + 1 \leq 2K$. The sum of all integers in X is $pK' + pq(K+1) = p(q+1)K$. It is clear that X can be partitioned into multisets $X_1, \ldots, X_p$ of equal cardinalities and sums if and only if W' can be partitioned into multisets $W'_1, \ldots, W'_p$ of equal cardinalities and sums, since the difference between x_j and w_j is the same for all j. Note that the reductions in this proof can all be done in polynomial time. This proves the NP-hardness of BALANCED MULTI-PARTITION. □

5.3 Reformability of Allocations

We start by investigating the decision problem of whether a given initial allocation can be reformed into an EF1 allocation. By Proposition 2.2.1, this reformation is possible if and only if there exists an EF1 allocation with the same size vector as the initial allocation. Therefore, in the rest of this section, we shall equivalently focus on the problem of deciding the existence of an EF1 allocation with a given size vector—this problem can be of interest independently of reformation considerations, e.g., when space constraints are present.

Now, Proposition 5.2.1 tells us that an EF1 allocation with a balanced size vector always exists. This means that the only time when we may have difficulties in ascertaining whether an EF1 allocation exists is when the given size vector is *not* balanced. In fact, as some of our proofs in this section show, the decision problem is NP-hard even when the sizes of the agents' bundles differ by at most *two* (e.g., in Theorem 5.3.3).

We discuss the cases of two agents, a constant number of agents, and a general number of agents separately. For each of these cases, we explore how the hardness of the decision problem varies across different classes of utility functions. Our results are summarized in Table 5.1.

For convenience, we refer to as REFORMABILITY the problem of deciding whether an EF1 allocation with a given size vector exists in a given instance. Note that REFORMABILITY is in NP regardless of the number of agents, as we can verify whether a given allocation is EF1 in polynomial time by simply comparing the bundles of every pair of agents.

5.3.1 Two Agents

For two agents, interestingly, the computational complexity of the problem turns out to be different depending on whether the agents have identical utilities or not. We begin our discussion with the case of identical utilities. For two agents with identical utilities, we first provide a simple characterization for checking whether a desired EF1 allocation exists based on the size vector and the utilities of the goods. We show that an EF1 allocation with a given size vector exists if and only if the agent with a smaller number of goods is EF1 towards the other agent in the allocation where the former agent receives the most valuable goods. Note that it is not required to check that the latter agent is EF1 towards the former agent.

Lemma 5.3.1. *Given an instance with two agents with identical utilities, let $\vec{s} = (s_1, s_2)$ be a size vector with $s_1 \leq s_2$. Assume that the goods $g_1, \ldots, g_m$ are arranged in non-increasing order of utility, and let $G_0 = \{g_1, \ldots, g_{s_1}\}$. Then, there exists an EF1 allocation with size vector $\vec{s}$ if and only if agent 1 is EF1 towards agent 2 in the allocation $(G_0, G \setminus G_0)$.*

Proof. We say in this proof that for any nonempty set $G' \subseteq G$, the good $g_i \in G'$ is the *most valuable* good in G' if g_i is the good with the smallest index in G'; likewise, g_i is the *least valuable* good in G' if g_i is the good with the largest index in G'. Note that the most (resp. least) valuable good in G' is the one with the highest (resp. lowest) utility among all the goods in G', with ties broken by index.

($\Rightarrow$) Let (A_1, A_2) be an EF1 allocation with size vector $\vec{s}$. If $A_1 = G_0$, then we are done; therefore, assume that $A_1 \neq G_0$. By definition, agent 1 is EF1 towards agent 2 in (A_1, A_2). We show that for any allocation (B_1, B_2) where $B_1 \neq G_0$, if agent 1 is EF1 towards agent 2, then exchanging the least valuable good in B_1 with the most valuable good in B_2 leads to an allocation where agent 1 is still EF1 towards agent 2. By repeating this procedure on (A_1, A_2), we eventually arrive at the allocation $(G_0, G \setminus G_0)$ and the conclusion that agent 1 is EF1 towards agent 2 in the latter allocation.

Let $h_1 \in B_1$ be the least valuable good in agent 1's bundle, and $h_2 \in B_2$ be the most valuable good in agent 2's bundle. Since $B_1 \neq G_0$, we have $h_1 \in G \setminus G_0$ and $h_2 \in G_0$, and hence $u(h_2) \geq u(h_1)$. Since agent 1 is EF1 towards agent 2, we have $u(B_1) \geq u(B_2 \setminus \{h_2\})$. Let $B_1' = (B_1 \cup \{h_2\}) \setminus \{h_1\}$ and $B_2' = (B_2 \cup \{h_1\}) \setminus \{h_2\}$ be the bundles after exchanging h_1 and h_2. Then,

$$\begin{aligned} u(B_1') &= u((B_1 \cup \{h_2\}) \setminus \{h_1\}) \\ &= u(B_1) + u(h_2) - u(h_1) \\ &\geq u(B_1) \\ &\geq u(B_2 \setminus \{h_2\}) \\ &= u(B_2' \setminus \{h_1\}), \end{aligned}$$

which means that agent 1 is EF1 towards agent 2 in (B_1', B_2').

($\Leftarrow$) Suppose that agent 1 is EF1 towards agent 2 in the allocation $(G_0, G \setminus G_0)$. If agent 2 is also EF1 towards agent 1 in $(G_0, G \setminus G_0)$, then we are done; therefore, assume that agent 2 envies agent 1 by more than one good. For notational simplicity, let $h_j = g_{s_1+j}$ for $j \in \{1, \ldots, s_1\}$, so that the goods arranged in non-increasing order of utility are $g_1, g_2, \ldots, g_{s_1}, h_1, h_2, \ldots, h_{s_1}, g_{2s_1+1}, \ldots, g_m$. Let $A_1^1 = G_0 = \{g_1, \ldots, g_{s_1}\}$ and $A_2^1 = G \setminus G_0 = \{h_1, \ldots, h_{s_1}\} \cup \{g_{2s_1+1}, \ldots, g_m\}$.

Let $t = 1$. In the allocation (A_1^t, A_2^t), agent 1 is EF1 towards agent 2, but agent 2 envies agent 1 by more than one good. Since g_t is the most valuable good in A_1^t, we have $u(A_2^t) < u(A_1^t \setminus \{g_t\})$. Let $A_1^{t+1} = (A_1^t \cup \{h_t\}) \setminus \{g_t\}$ and $A_2^{t+1} = (A_2^t \cup \{g_t\}) \setminus \{h_t\}$ be the bundles after exchanging g_t and h_t. Then, we have

$$\begin{aligned} u(A_1^{t+1}) &= u((A_1^t \cup \{h_t\}) \setminus \{g_t\}) \\ &\geq u(A_1^t \setminus \{g_t\}) \\ &> u(A_2^t) \\ &= u((A_2^{t+1} \cup \{h_t\}) \setminus \{g_t\}) \\ &\geq u(A_2^{t+1} \setminus \{g_t\}), \end{aligned}$$

so agent 1 is EF1 towards agent 2 in (A_1^{t+1}, A_2^{t+1}). If agent 2 is also EF1 towards agent 1, then (A_1^{t+1}, A_2^{t+1}) is an EF1 allocation and we are done. Otherwise, agent 2 envies agent 1 by more than one good, and we increment t by 1 and repeat the discussion in this paragraph.

If we still have not found an EF1 allocation after $t = s_1$, then agent 1 is EF1 towards agent 2 in $(A_1^{s_1+1}, A_2^{s_1+1})$, where $A_1^{s_1+1} = \{h_1, \ldots, h_{s_1}\} \subseteq A_2^1$ and $A_2^{s_1+1} = \{g_1, \ldots, g_{s_1}\} \cup \{g_{2s_1+1}, \ldots, g_m\} \supseteq A_1^1$, and g_1 is the most valuable good in $A_2^{s_1+1}$. This implies that

$$u(A_1^{s_1+1}) \leq u(A_2^1) < u(A_1^1 \setminus \{g_1\}) \leq u(A_2^{s_1+1} \setminus \{g_1\}),$$

which means that agent 1 is *not* EF1 towards agent 2 in $(A_1^{s_1+1}, A_2^{s_1+1})$. This is a contradiction; therefore, (A_1^t, A_2^t) must be EF1 for some $t \in \{1, \ldots, s_1\}$. $\square$

Since the condition in Lemma 5.3.1 can be checked in polynomial time, we can derive the following result.

Theorem 5.3.2. *Reformability is in P for two agents with identical utilities.*

Proof. Without loss of generality, let the size vector be (s_1, s_2) with $s_1 \leq s_2$. Arrange the goods $g_1, \dots, g_m$ in non-increasing order of utility, and let G_0 be the set of s_1 goods with the highest utilities. By Lemma 5.3.1, there exists an EF1 allocation with size vector (s_1, s_2) if and only if agent 1 is EF1 towards agent 2 in the allocation $(G_0, G \setminus G_0)$. The latter condition can be checked in polynomial time. □

While deciding whether an EF1 allocation with the given size vector exists can be done efficiently for two agents with identical utilities, we remark here that deciding whether an *envy-free* allocation exists is NP-hard for two agents with identical utilities even if we allow any size vector—this follows directly from a reduction from Partition.[3]

We now proceed to general utilities. Lemma 5.3.1 assumes identical utilities, and there is no obvious way to generalize it to non-identical utilities. In fact, perhaps surprisingly, we show that the decision problem becomes NP-hard when we drop the assumption of identical utilities. The proof follows from a reduction from Balanced Multi-Partition with $p = 2$, an NP-hard problem by Proposition 5.2.2.

Theorem 5.3.3. *Reformability is weakly NP-complete for two agents.*

Proof. Clearly, this problem is in NP. The "weak" aspect is demonstrated later in Lemma 5.3.4, which says that there exists a pseudopolynomial-time algorithm that solves this problem for any constant number of agents. Therefore, it suffices to show that this problem is NP-hard.

To demonstrate NP-hardness, we shall reduce from the NP-hard problem Balanced Multi-Partition with $p = 2$ (see Proposition 5.2.2). Let a Balanced Multi-Partition instance be given with $p = 2$. Without loss of generality, assume that $q \geq 2$. Let $Y = \{y_1, \dots, y_{2q+2}\}$ be a multiset such that $y_j = x_j$ for $j \in \{1, \dots, 2q\}$, $y_{2q+1} = 2K$, and $y_{2q+2} = 0$. We claim that Y can be partitioned into two multisets Y_1 and Y_2 of equal cardinalities (i.e., of size $q+1$ each) with sums $(q+3)K$ and $(q+1)K$ respectively if and only if X can be partitioned into two multisets X_1 and X_2 of equal cardinalities and sums. If the latter condition is true, then let Y_1 (resp. Y_2) contain the corresponding elements in X_1 (resp. X_2), and let $y_{2q+1} \in Y_1$ and $y_{2q+2} \in Y_2$—this gives an appropriate partition of Y. Conversely, if the former condition is true, then we show that X can be partitioned appropriately. Note that if $y_{2q+1} \in Y_2$, then there are at least $q - 1 > 0$ integers in $\{y_1, \dots, y_{2q}\}$ that are also in Y_2. Since every integer in $\{y_1, \dots, y_{2q}\}$ is more than K, the sum of Y_2 will be more than $(q-1)K + 2K = (q+1)K$, which is a contradiction. This means that $y_{2q+1} \in Y_1$. Similarly, if $y_{2q+2} \in Y_1$, then there are exactly $q+1$ integers in $\{y_1, \dots, y_{2q}\}$ that are in Y_2. The sum of Y_2 will be more than $(q+1)K$, which is a contradiction. Hence, $y_{2q+2} \in Y_2$. Now, this means that $\{y_1, \dots, y_{2q}\}$ must be partitioned into two multisets of equal cardinalities (i.e., of size q each) with sum $(q+1)K$ each. This induces an appropriate partition of X.

Next, define a fair division instance as follows. There are $n = 2$ agents and a set of goods $G = \{g_1, \dots, g_{2q+6}\}$. Agent 2's utility is such that $u_2(g_j) = y_j$ for $j \in \{1, \dots, 2q+2\}$, $u_2(g_{2q+3}) = u_2(g_{2q+4}) = 0$, and $u_2(g_{2q+5}) = u_2(g_{2q+6}) = 2K$. Agent 1's utility is such that

[3] If we require both agents to receive the same number of goods, the problem for envy-freeness remains NP-hard by a reduction from an equal-cardinality version of Partition.

$u_1(g) = u_2(g) + 4K$ for all $g \in G$. The size vector $\vec{s}$ is $(q+2, q+4)$. This reduction can be done in polynomial time. We claim that there exists an EF1 allocation with size vector $\vec{s}$ in this instance if and only if Y can be partitioned into two multisets Y_1 and Y_2 of equal cardinalities (i.e., of size $q+1$ each) with sums $(q+3)K$ and $(q+1)K$ respectively.

($\Leftarrow$) Let J_1' and J_2' be a partition of $\{1, \dots, 2q+2\}$ of equal cardinalities such that $\sum_{j \in J_1'} y_j = (q+3)K$ and $\sum_{j \in J_2'} y_j = (q+1)K$. Let $A_1 = \{g_j\}_{j \in J_1'} \cup \{g_{2q+5}\}$ and $A_2 = G \setminus A_1$ be the two agents' bundles respectively. From agent 1's perspective, agent 1's bundle has utility $((q+3)K + 2K) + (q+2)(4K) = (5q+13)K$, agent 2's bundle has utility $((q+1)K + 2K) + (q+4)(4K) = (5q+19)K$, and a most valuable good in agent 2's bundle (e.g., g_{2q+6}) has utility $6K$, so agent 1 is EF1 towards agent 2. From agent 2's perspective, agent 2's bundle has utility $(q+1)K + 2K = (q+3)K$, agent 1's bundle has utility $(q+3)K + 2K = (q+5)K$, and a most valuable good in agent 1's bundle (e.g., g_{2q+5}) has utility $2K$, so agent 2 is EF1 towards agent 1. Accordingly, (A_1, A_2) is an EF1 allocation with size vector $(q+2, q+4)$.

($\Rightarrow$) Let (A_1, A_2) be an EF1 allocation with size vector $\vec{s}$. From agent 1's perspective, $u_1(G) = (10q+32)K$ and a most valuable good (e.g., g_{2q+5}) has utility $6K$. For agent 1 to be EF1 towards agent 2, we must have $u_1(A_1) \geq ((10q+32)K - 6K)/2 = (5q+13)K$ and $u_2(A_1) = u_1(A_1) - (q+2)(4K) \geq (q+5)K$. This means that $u_1(A_2) = u_1(G) - u_1(A_1) \leq (5q+19)K$ and $u_2(A_2) = u_1(A_2) - (q+4)(4K) \leq (q+3)K$. On the other hand, from agent 2's perspective, $u_2(G) = (2q+8)K$ and a most valuable good has utility $2K$. For agent 2 to be EF1 towards agent 1, we must have $u_2(A_2) \geq ((2q+8)K - 2K)/2 = (q+3)K$ and $u_1(A_2) = u_2(A_2) + (q+4)(4K) \geq (5q+19)K$. This means that $u_2(A_1) = u_2(G) - u_2(A_2) \leq (q+5)K$ and $u_1(A_1) = u_2(A_1) + (q+2)(4K) \leq (5q+13)K$. By combining these inequalities, we conclude that these inequalities are tight, i.e., agent 1's utilities for both agents' bundles are *exactly* $(5q+13)K$ and $(5q+19)K$ respectively so that the sum is $(10q+32)K$, and agent 2's utilities for both agents' bundles are *exactly* $(q+5)K$ and $(q+3)K$ respectively so that the sum is $(2q+8)K$. Additionally, both agents must each have a most valuable good worth $6K$ and $2K$ to them respectively. Without loss of generality, we may assume that $g_{2q+5} \in A_1$ and $g_{2q+6} \in A_2$ (note that g_{2q+1} is also a most valuable good, but we use g_{2q+5} and g_{2q+6} for simplicity).

Since $g_{2q+6} \in A_2$, there are $q+3$ goods in $A_2 \setminus \{g_{2q+6}\}$ and $u_2(A_2 \setminus \{g_{2q+6}\}) = (q+3)K - 2K = (q+1)K$. These goods are chosen from $G_1 = \{g_1, \dots, g_{2q+1}\}$ and $G_0 = \{g_{2q+2}, g_{2q+3}, g_{2q+4}\}$. Recall from the construction that $u_2(g) > K$ for all $g \in G_1$, and $u_2(g) = 0$ for all $g \in G_0$. If $A_2 \setminus \{g_{2q+6}\}$ contains at least $q+1$ goods from G_1, then $u_2(A_2 \setminus \{g_{2q+6}\}) > (q+1)K$, a contradiction. Therefore, $A_2 \setminus \{g_{2q+6}\}$ contains at most q goods from G_1, and at least 3 goods from G_0. Since $|G_0| = 3$, we must have $G_0 \subseteq A_2$. Note that $u_2(G_0) = 0$, so $u_2((A_2 \setminus \{g_{2q+6}\}) \setminus G_0) = (q+1)K$. Therefore, the q goods from G_1 in agent 2's bundle have a total utility of $(q+1)K$. These goods, together with g_{2q+2}, induce the set Y_2 with cardinality $q+1$ and sum $(q+1)K$. Then, $Y_1 = Y \setminus Y_2$ and Y_2 give a required partition of Y. $\square$

For two agents with binary utilities, we shall show later that the decision problem is in P (see Theorem 5.3.7).

5.3.2 Constant Number of Agents

Next, we discuss the complexity of the decision problem for a *constant number of agents*. In this case, we devise a pseudopolynomial-time algorithm for deciding the existence of an EF1 allocation with a given size vector.

Lemma 5.3.4. *Let an instance with n agents and a size vector be given, where n is a constant. Suppose that the utility of each good is an integer, and let $R = \max_i u_i(G)$. Then, there exists an algorithm running in time polynomial in m and R that decides whether the instance admits an EF1 allocation with the size vector.*

Proof. The algorithm uses dynamic programming. We construct a table with m columns and L rows, where L will be specified later. The index of each row is represented by a tuple containing $a_{i,j}$, $b_{i,j}$, and c_i for each $i, j \in N$, i.e., $(a_{1,1}, a_{1,2}, \dots, a_{n,n}, b_{1,1}, b_{1,2}, \dots, b_{n,n}, c_1, \dots, c_n)$. The value of $a_{i,j}$ is the utility of agent j's bundle from agent i's perspective, i.e., $a_{i,j} = u_i(A_j)$; the value of $b_{i,j}$ is the utility of a most valuable good in agent j's bundle from agent i's perspective, i.e., $b_{i,j} = \max_{g \in A_j} u_i(g)$ (note that this value is zero if $A_j = \varnothing$); and the value of c_i is the number of goods in agent i's bundle. Note that $a_{i,j}, b_{i,j} \in \{0, \dots, R\}$ and $c_i \in \{0, \dots, m\}$, so there are $L = (R+1)^{2n^2}(m+1)^n$ rows, which is polynomial in m and R. An entry in column q represents whether an allocation involving $\{g_1, \dots, g_q\}$ is possible for the tuple representing the row, and is either positive or negative.

Initialize every entry to negative. Consider the n possibilities of adding g_1 to each of the agents' bundles respectively, and set the corresponding entries in the first column of the table to positive. In particular, for each $j \in N$, the row represented by the tuple such that $a_{i,j} = b_{i,j} = u_i(g_1)$ and $c_j = 1$ for all $i \in N$, and zero for all other values in the tuple, has the entry (in the first column) set to positive.

Now, for each $q \in \{2, \dots, m\}$ in ascending order, for each positive entry in column $q-1$, consider the n possibilities of adding g_q into each of the n agents' bundles respectively, and set the corresponding entry for each of these possibilities in column q to positive. Once this procedure is done, consider all positive entries in column m. If some positive entry corresponds to an EF1 allocation with the required size vector, then the instance admits such an EF1 allocation; otherwise, no such allocation exists.

Since n is a constant, the number of entries in the table is polynomial in m and R. At each column, there is a polynomial number of rows with positive entries, and hence the update is polynomial. Finally, checking for a feasible EF1 allocation at the last column can also be done in polynomial time. $\square$

We now move to *polynomial-time* algorithms that determine the existence of an EF1 allocation with a given size vector. Recall that such an algorithm exists for two agents with identical utilities (Theorem 5.3.2). However, it turns out that such an algorithm does not exist for three or more agents with identical utilities, unless P = NP. In particular, we establish the NP-hardness of the decision problem via a reduction from Balanced Multi-Partition with $p = 2$, an NP-hard problem by Proposition 5.2.2.

Theorem 5.3.5. *Reformability is weakly NP-complete for $n \geq 3$ agents with identical utilities, where n is a constant.*

Proof. Clearly, this problem is in NP. The "weak" aspect is demonstrated in Lemma 5.3.4. Therefore, it suffices to show that this problem is NP-hard.

To show NP-hardness, we shall reduce from the NP-hard problem Balanced Multi-Partition with $p = 2$ (see Proposition 5.2.2). Let a Balanced Multi-Partition instance with $p = 2$ be given. Define a fair division instance as follows. There are $n \geq 3$ agents with identical utilities, and a set of goods $G = \{g_1, \ldots, g_{2q}, h_1, \ldots, h_n\}$ such that $u(g_j) = x_j$ for $j \in \{1, \ldots, 2q\}$ and $u(h_k) = (q+1)K$ for $k \in \{1, \ldots, n\}$. The size vector $\vec{s}$ is such that $s_1 = s_2 = q + 1$ and $s_k = 1$ for all $k \in \{3, \ldots, n\}$. This reduction can be done in polynomial time. We claim that there exists an EF1 allocation with size vector $\vec{s}$ in this instance if and only if X can be partitioned into multisets X_1, X_2 of equal cardinalities and sums.

($\Leftarrow$) Let (X_1, X_2) be such a partition. Define an allocation such that agent k receives h_k for $k \in N$, agent 1 additionally receives the q goods corresponding to the integers in X_1, and agent 2 additionally receives the q goods corresponding to the integers in X_2. We show that this allocation is EF1. The utilities of agent 1's and agent 2's bundles are $2(q + 1)K$ each, and the utilities of the other agents' bundles are $(q+1)K$ each, so agents 1 and 2 do not envy anyone else. Therefore, it remains to check that agent k is EF1 towards agents 1 and 2 for $k \in \{3, \ldots, n\}$. Upon the removal of the single good h_1 (resp. h_2) from agent 1's (resp. agent 2's) bundle, the remaining bundle has utility $(q + 1)K$, so agent k is EF1 towards agent 1 (resp. agent 2). Therefore, the allocation is EF1, as desired.

($\Rightarrow$) Let $(A_1, \ldots, A_n)$ be an EF1 allocation with size vector $(q + 1, q + 1, 1, \ldots, 1)$. If agent 1's bundle has at least two goods from $\{h_1, \ldots, h_n\}$, then her bundle without the most valuable good has utility more than $(q+1)K$ since her bundle also contains other goods with positive utility. Agent 3, having a bundle of utility at most $(q+1)K$, will not be EF1 towards agent 1, contradicting the assumption that the allocation is EF1. Therefore, agent 1's bundle has at most one good from $\{h_1, \ldots, h_n\}$; likewise for agent 2's bundle. This means that every agent receives *exactly* one good from $\{h_1, \ldots, h_n\}$. Having established this, agent 3's bundle has a utility of $(q + 1)K$, and agent 3 is EF1 towards agent 1. This means that agent 1's bundle without a most valuable good (say, some h_k) must have utility at most $(q + 1)K$. The same argument can be used to show the same statement for agent 2's bundle. This means that the goods $\{g_1, \ldots, g_{2q}\}$ must be divided between agents 1 and 2 with each agent receiving a utility of *exactly* $(q + 1)K$. Such a division of $\{g_1, \ldots, g_{2q}\}$ induces a partition of X into two multisets of equal cardinalities and sums, as desired. □

Since the decision problem is NP-hard even for identical utilities, it must also be NP-hard for general utilities. We now consider another class of utilities: binary utilities. When there are n agents, every good g belongs to one of 2^n types of goods represented by the vector $(u_1(g), \ldots, u_n(g))$. For the purpose of determining whether an EF1 allocation exists, it suffices to consider different goods of the same type as *indistinguishable*. We say that two allocations are in the same equivalence class if the number of goods of each type that each agent has is the same in both allocations. If $\mathcal{A}$ is an EF1 allocation, then all allocations in the same equivalence class as $\mathcal{A}$ are also EF1 and can be reformed from $\mathcal{A}$. We shall proceed with a result which enumerates all (essentially equivalent) EF1 allocations in time polynomial in the number of goods, provided that the number of agents is a constant.

Lemma 5.3.6. *Let an instance with n agents with binary utilities and a size vector be given, where n is a constant. Then, there exists an algorithm running in time polynomial in m that outputs all equivalent classes of EF1 allocations with the size vector.*

Proof. An agent's bundle can be represented by a 2^n-vector where each component of the vector is the number of goods of that type in her bundle. Since the number of goods of each type is an integer between 0 and m, there are $m+1$ possible values for each component, and hence at most $(m+1)^{2^n}$ possible vectors to represent each agent's bundle. Allocations in an equivalence class can be represented by an ordered collection of n such vectors—one for each agent—and there are at most $((m+1)^{2^n})^n$ such collections. Since $((m+1)^{2^n})^n$ is polynomial in m whenever n is a constant, there is at most a polynomial number of possible equivalence classes of allocations in the instance. For each of these equivalence classes of allocations, we can check whether an allocation in the equivalence class is EF1 and has the required size vector in polynomial time, and output the equivalence class if so. Therefore, the overall running time is polynomial in m, as claimed. □

Lemma 5.3.6 implies that the decision problem can be solved efficiently for binary utilities.

Theorem 5.3.7. *Reformability is in P for a constant number of agents with binary utilities.*

Proof. Use the algorithm as described in Lemma 5.3.6 to enumerate all equivalence classes with an EF1 allocation with the size vector, and output "yes" if and only if such an equivalence class is found. Note that if some allocation in an equivalence class is EF1, then all allocations in the same equivalence class are also EF1. □

5.3.3 General Number of Agents

For any constant number of agents, the problem of determining the existence of an EF1 allocation with a given size vector is *weakly* NP-hard by Theorem 5.3.5 (even for identical utilities). For a general number of agents, the pseudopolynomial-time algorithm as described in Lemma 5.3.4 does not work, since that algorithm is at least exponential in the number of agents. Therefore, the decision problem for a general number of agents might not be *weakly* NP-hard. We show that the problem is indeed *strongly* NP-hard by a reduction from 3-Partition, a strongly NP-hard problem (Garey and Johnson, 1979, p. 224).

Theorem 5.3.8. *Reformability is strongly NP-complete for identical utilities.*

Proof. Clearly, this problem is in NP. Therefore, it suffices to show that it is strongly NP-hard.

To this end, we shall reduce from 3-Partition. In 3-Partition, we are given positive integers q and K, and a multiset $X = \{x_1, \dots, x_{3q}\}$ of positive integers of total sum qK. The problem is to decide whether X can be partitioned into multisets $X_1, \dots, X_q$ of equal cardinalities and sums, i.e., for each $i \in \{1, \dots, q\}$, $|X_i| = 3$ and the sum of all the integers in X_i is exactly K. This decision problem is known to be strongly NP-hard, even if $K/4 < x_j < K/2$ for every $j \in \{1, \dots, 3q\}$ (Garey and Johnson, 1979, p. 224).

Let an instance of 3-PARTITION be given. Define a fair division instance as follows. There are $n = q + 1$ agents with identical utilities, and a set of goods $G = \{g_1, \ldots, g_{3q+6}\}$ such that $u(g_j) = x_j$ for $j \in \{1, \ldots, 3q\}$ and $u(g_k) = K/5$ for $k \in \{3q + 1, \ldots, 3q + 6\}$. The size vector $\vec{s}$ is such that $s_j = 3$ for all $j \in \{1, \ldots, q\}$ and $s_{q+1} = 6$. This reduction can be done in polynomial time. We claim that there exists an EF1 allocation with size vector $\vec{s}$ in this instance if and only if there exists a partition of X into multisets $X_1, \ldots, X_q$ of equal cardinalities and sums.

($\Leftarrow$) Let such a partition be given. Define an allocation such that each of the first q agents receives the three goods corresponding to each multiset, and agent $q + 1$ receives $\{g_{3q+1}, \ldots, g_{3q+6}\}$. Note that agents 1 to q have bundles worth K each, and agent $q + 1$ has a bundle worth $6K/5$. Since each of the goods from agent $(q + 1)$'s bundle is worth exactly $K/5$, every agent is EF1 towards agent $q + 1$. Accordingly, the allocation is EF1.

($\Rightarrow$) Let an EF1 allocation with size vector $\vec{s}$ be given. Note that every good is worth at least $K/5$, so agent $(q + 1)$'s bundle without a most valuable good is worth at least K. If agent $q+1$ receives a bundle worth more than $6K/5$, then some agent receives a bundle worth less than K, and will not be EF1 towards agent $q + 1$. Therefore, agent $q + 1$ must receive a bundle worth at most $6K/5$. The only way this is possible is when agent $q + 1$ receives $\{g_{3q+1}, \ldots, g_{3q+6}\}$. Now, since agents 1 to q are EF1 towards agent $q + 1$, these agents must each receive a bundle worth at least K. The only way this is possible is when each of them receives a bundle worth *exactly* K. This induces the desired partition. $\square$

For binary utilities, the decision problem for a *constant* number of agents is in P (Theorem 5.3.7). The crucial reason is that in this case, the number of different types of goods is also a constant, which allows us to enumerate all the (essentially equivalent) EF1 allocations in polynomial time (Lemma 5.3.6). This is no longer possible when the number of agents is non-constant. In fact, we show that the decision problem is strongly NP-hard for a general number of agents with binary utilities. To this end, we reduce from GRAPH k-COLORABILITY, which is strongly NP-hard for any fixed $k \geq 3$ (Garey and Johnson, 1979, p. 191).

Theorem 5.3.9. *REFORMABILITY is strongly NP-complete for binary utilities.*

Proof. Clearly, this problem is in NP. Therefore, it suffices to show that it is strongly NP-hard.

To this end, we shall reduce from GRAPH k-COLORABILITY with $k \geq 3$. In GRAPH k-COLORABILITY, we are given a graph $\widetilde{G} = (V, E)$ and a positive integer k, and the problem is to decide whether $\widetilde{G}$ is k-colorable, i.e., whether each of the vertices in V can be assigned one of k colors in such a way that no two adjacent vertices are assigned the same color. This decision problem is known to be strongly NP-hard for any fixed $k \geq 3$ (Garey and Johnson, 1979, p. 191).

Let an instance of GRAPH k-COLORABILITY be given with a fixed $k \geq 3$, where $V = \{v_1, \ldots, v_p\}$ and $E = \{e_1, \ldots, e_q\}$. Define a fair division instance as follows. There are $n = q + k$ agents where the first q agents are called *edge agents* and the last k agents are called *color agents*. There are $m = kp$ goods. Each color agent assigns zero utility to every good. For $r \in \{1, \ldots, q\}$, if $e_r = \{v_i, v_j\}$, then the r^{th} edge agent assigns a utility of 1 to each of g_i and g_j, and zero utility to every other good. Note that only the first p goods correspond to vertices and are valuable to the edge agents whose corresponding edges are incident to the vertices; the remaining $(k-1)p$ goods are not valuable to any agent. The size vector $\vec{s}$ is such that $s_r = 0$ for each edge agent r and $s_c = p$ for each color agent c. This reduction can be done in polynomial time. We claim that there exists an EF1 allocation with size vector $\vec{s}$ in this instance if and only if $\widetilde{G}$ is k-colorable.

($\Leftarrow$) Let a proper k-coloring of $\widetilde{G}$ be given. For $t \in \{1, \ldots, p\}$, if vertex v_t is assigned the color c, then allocate g_t to the color agent c. Since there are p such goods and each color agent is supposed to have p goods in her bundle, it is possible to allocate all of these goods. Subsequently, allocate the remaining goods arbitrarily among the color agents until every color agent has exactly p goods. We claim that this allocation is EF1. Every color agent assigns zero utility to every good and is thus EF1 towards every other agent. Each edge agent assigns a utility of 1 to only two goods, so we only need to check that these two goods are in different bundles. Indeed, these two goods correspond to vertices which are adjacent to each other in $\widetilde{G}$, and proper coloring of $\widetilde{G}$ implies that the vertices are of different colors, so the corresponding goods are in different color agents' bundles. Therefore, the allocation is EF1.

($\Rightarrow$) Let an EF1 allocation with size vector $\vec{s}$ be given. For $t \in \{1, \ldots, p\}$, if the good g_t is with color agent c, assign v_t to color c. We claim that this coloring is a proper k-coloring of $\widetilde{G}$. Since there are k color agents, at most k colors are used. Therefore, it suffices to check that adjacent vertices are assigned different colors. Let $v_i, v_j \in V$ be adjacent vertices. Then, there exists an edge $e_r = \{v_i, v_j\}$. The edge agent r assigns a utility of 1 to each of g_i and g_j. Since agent r's bundle is empty, g_i and g_j must be in different (color agents') bundles in order for agent r to be EF1 towards every other agent. This implies that v_i and v_j are assigned different colors. $\square$

Even though the decision problem is strongly NP-hard for identical *or* binary utilities, we prove next that it can be solved in polynomial time for identical *and* binary utilities. Indeed, this can be done by checking whether the total number of valuable goods is within a certain threshold that can be computed in polynomial time.

Theorem 5.3.10. *Reformability is in P for identical binary utilities.*

Proof. Let $\vec{s}$ be the given size vector. Let $s_0 = \min_{i \in N} s_i$ be the size of the smallest bundle, and $n_0 = |\{i \in N \mid s_i = s_0\}|$ be the number of agents with exactly s_0 goods in their bundles. We claim that an EF1 allocation with size vector $\vec{s}$ exists if and only if the number of valuable goods is at most $s_0 n + n - n_0$. Note that checking whether the number of valuable goods is at most $s_0 n + n - n_0$ can be done in polynomial time.

If an EF1 allocation with size vector $\vec{s}$ exists, then an agent with bundle size s_0 receives at most s_0 valuable goods. For this agent to be EF1 towards every other agent, every other agent can only receive at most $s_0 + 1$ valuable goods. Since there are $n - n_0$ agents with bundle sizes at least $s_0 + 1$ and n_0 agents with bundle sizes exactly s_0, the total number of valuable goods is at most $(n - n_0)(s_0 + 1) + n_0 s_0 = s_0 n + n - n_0$.

Conversely, if the number of valuable goods is at most $s_0 n + n - n_0$, then we can allocate the valuable goods in a round-robin fashion up to the bundle size of each agent, followed by the non-valuable goods. Note that every agent's bundle size is at least s_0. If there are at most $s_0 n$ valuable goods, then these valuable goods can be distributed fairly with the difference in the number of valuable goods between agents being at most one, and hence the allocation is EF1. Otherwise, the first $s_0 n$ valuable goods can be distributed so that every agent receives s_0 of them. Since there are $n - n_0$ agents with bundle size at least $s_0 + 1$ and at most $n - n_0$ valuable goods left, the remaining valuable goods can be arbitrarily allocated to these agents so that each of these agents receives at most one more valuable good. Then, each agent receives s_0 or $s_0 + 1$ valuable goods, and so the allocation is EF1. □

5.4 Optimal Number of Exchanges

In this section, we consider the complexity of computing the *optimal number of exchanges* required to reach an EF1 allocation from an initial allocation.

Recall that the decision problem in Section 5.3 is to determine whether there exists an EF1 allocation that can be reached from a given initial allocation. This is equivalent to determining whether the optimal number of exchanges to reach an EF1 allocation is finite or not. We have established a few scenarios where there exist polynomial-time algorithms for this task: (a) two agents with identical utilities (Theorem 5.3.2), (b) any constant number of agents with binary utilities (Theorem 5.3.7), and (c) any number of agents with identical binary utilities (Theorem 5.3.10). For these scenarios, we can run the respective polynomial-time algorithms to decide whether such an EF1 allocation exists—if none exists, then the optimal number of exchanges is ∞. Therefore, for the proofs in this section pertaining to these scenarios, we proceed with the assumption that such an EF1 allocation exists. We will show that the problem of computing the optimal number of exchanges for these scenarios is also in P.[4]

For the remaining scenarios where the decision problem in Section 5.3 is NP-hard, it is NP-hard to even decide whether the optimal number of exchanges to reach an EF1 allocation is finite or not. Therefore, for these scenarios, we shall focus on the special case where the given size vector is *balanced*, so that the optimal number of exchanges is guaranteed to be finite (see Proposition 5.2.1). Even with this assumption, we will show that the computational problem for these scenarios remains NP-hard.

For convenience, we refer to as Optimal Exchanges the problem of computing the optimal number of exchanges required to reach an EF1 allocation given an instance and an initial allocation in the instance. The results in this section are summarized in Table 5.2.

5.4.1 Two Agents

We begin with the case of two agents. For two agents with identical utilities, we show that there exists a polynomial-time algorithm that computes the optimal number of exchanges. This algorithm performs the exchanges until an EF1 allocation is reached, while keeping track of the number of exchanges required. The algorithm is "greedy" in the sense that at each step, it performs an exchange involving the most valuable good from the agent whose bundle

[4] Our algorithms can be modified to compute an optimal *sequence* of exchanges as well.

has the higher utility, and the least valuable good from the other agent. We demonstrate that this choice is the best in terms of the number of exchanges required to reach an EF1 allocation.

Theorem 5.4.1. *Optimal Exchanges is in P for two agents with identical utilities.*

Proof. We assume that an EF1 allocation with the given size vector exists. If the initial allocation $\mathcal{A}$ is EF1, we are done. Otherwise, assume without loss of generality that agent 2 has a higher utility than agent 1 in $\mathcal{A}$. Let $\vec{s} = (s_1, s_2)$ be the size vector. By rearranging the labels of the goods, assume that the goods are in non-increasing order of utility, i.e., $u(g_1) \geq u(g_2) \geq \cdots \geq u(g_m)$. The algorithm proceeds as follows: repeatedly exchange the most valuable good in agent 2's bundle with the least valuable good in agent 1's bundle until agent 1 is EF1 towards agent 2. The optimal number of exchanges required is then the number of exchanges made in this algorithm.

First, we claim that in each exchange, a good from $\{g_1, \ldots, g_{s_1}\}$ in agent 2's bundle is always exchanged with a good from $\{g_{s_1+1}, \ldots, g_m\}$ in agent 1's bundle. Suppose on the contrary that this is not true. Let $\mathcal{A}'$ be the allocation just before we make the exchange that violates this claim. The only way for the claim to be violated is when $A'_1 = \{g_1, \ldots, g_{s_1}\}$ and $A'_2 = \{g_{s_1+1}, \ldots, g_m\}$. If $s_1 \leq s_2$, then by Lemma 5.3.1, there does not exist an EF1 allocation with size vector $\vec{s}$—this would contradict our assumption that an EF1 allocation with size vector $\vec{s}$ exists. Otherwise, $s_1 > s_2$, and every good in A'_1 has a higher utility than every good in A'_2, so agent 1 is EF1 towards agent 2. This would contradict our assumption that the algorithm has not terminated. Hence, the claim is true.

By the claim in the previous paragraph, the total number of exchanges made is at most $\min\{s_1, s_2\} \leq m$. Each exchange can be performed in polynomial time, and so the algorithm terminates in polynomial time. We show next that an EF1 allocation is obtained when the algorithm terminates. It suffices to show that agent 2 is EF1 towards agent 1 in the final allocation. To this end, we show that agent 2 is EF1 towards agent 1 at every step of the algorithm. Let the initial allocation be $\mathcal{A}^0 = \mathcal{A}$, and let $\mathcal{A}^t$ be the allocation after t steps of the algorithm. Note that $\mathcal{A}^0$ satisfies the condition that agent 2 is EF1 towards agent 1, since agent 2 has a higher utility than agent 1 in $\mathcal{A}$. We show that if $\mathcal{A}^t$ has the property that agent 2 is EF1 towards agent 1 and agent 1 is *not* EF1 towards agent 2, then $\mathcal{A}^{t+1}$ has the property that agent 2 is EF1 towards agent 1. Suppose that $g \in A_2^t$ is exchanged with $h \in A_1^t$; note that g is a good with the highest utility in A_2^t. This means that $u(A_1^t) < u(A_2^t \setminus \{g\})$. Then,

$$\begin{aligned} u(A_2^{t+1}) &\geq u(A_2^{t+1} \setminus \{h\}) \\ &= u(A_2^t \setminus \{g\}) \\ &> u(A_1^t) \\ &\geq u(A_1^t \setminus \{h\}) \\ &= u(A_1^{t+1} \setminus \{g\}), \end{aligned}$$

showing that agent 2 is EF1 towards agent 1 in $\mathcal{A}^{t+1}$.

Finally, we show that the optimal number of exchanges required to reach an EF1 allocation is at least the number of exchanges made in this algorithm. Let T be the number of exchanges made in this algorithm. For each $t \in \{1, \dots, T\}$, let $g^t \in A_2$ (resp. $h^t \in A_1$) be the good in agent 2's bundle (resp. agent 1's bundle) that is exchanged at the t^{th} step of the algorithm. Note that we have $u(g^1) \geq \cdots \geq u(g^T) \geq u(h^T) \geq \cdots \geq u(h^1)$. We have $u(A_1^{T-1}) < u(A_2^{T-1} \setminus \{g^T\})$, where g^T is a good with the highest utility in agent 2's bundle in $\mathcal{A}^{T-1}$. Suppose on the contrary that only $k \leq T-1$ exchanges are required to reach an EF1 allocation. Since $\mathcal{A}$ is not EF1, we have $1 \leq k < T$. Let (B_1, B_2) be the EF1 allocation after the k exchanges. The utility of B_1 is upper-bounded by the utility of A_1 after adding k goods of the highest utility from A_2 and removing k goods of the lowest utility from A_1, so we have

$$\begin{aligned} u(B_1) &\leq u((A_1 \cup \{g^1, \dots, g^k\}) \setminus \{h^1, \dots, h^k\}) \\ &= u(A_1) + \sum_{t=1}^{k} (u(g^t) - u(h^t)) \\ &\leq u(A_1) + \sum_{t=1}^{T-1} (u(g^t) - u(h^t)) \\ &= u((A_1 \cup \{g^1, \dots, g^{T-1}\}) \setminus \{h^1, \dots, h^{T-1}\}) \\ &= u(A_1^{T-1}). \end{aligned}$$

On the other hand, the utility of B_2 without the most valuable good is lower-bounded by the utility of A_2 after adding k goods of the lowest utility from A_1 and removing $k+1 \leq T$ goods of the highest utility from A_2, so we have

$$\begin{aligned} u(B_2 \setminus \{g\}) &\geq u((A_2 \cup \{h^1, \dots, h^k\}) \setminus \{g^1, \dots, g^{k+1}\}) \\ &= u(A_2) - u(g^1) - \sum_{t=1}^{k} (u(g^{t+1}) - u(h^t)) \\ &\geq u(A_2) - u(g^1) - \sum_{t=1}^{T-1} (u(g^{t+1}) - u(h^t)) \\ &= u((A_2 \cup \{h^1, \dots, h^{T-1}\}) \setminus \{g^1, \dots, g^T\}) \\ &= u(A_2^{T-1} \setminus \{g^T\}) \end{aligned}$$

for every $g \in B_2$. This gives the inequality $u(B_1) \leq u(A_1^{T-1}) < u(A_2^{T-1} \setminus \{g^T\}) \leq u(B_2 \setminus \{g\})$ for all $g \in B_2$. Hence, agent 1 is not EF1 towards agent 2 in (B_1, B_2), contradicting the assumption that (B_1, B_2) is EF1. It follows that at least T exchanges are required to reach an EF1 allocation. $\square$

However, if the utilities are not identical, then computing the optimal number of exchanges is NP-hard, even for balanced allocations. We show this by modifying the construction from the NP-hardness proof of Theorem 5.3.3 in determining the existence of an EF1 allocation with a given size vector.

Theorem 5.4.2. *Optimal Exchanges is NP-hard for two agents, even when the initial allocation is balanced.*

Proof. We modify the construction from the proof of Theorem 5.3.3. Recall that we have $Y = \{y_1, \ldots, y_{2q+2}\}$ with $K < y_j \leq 2K$ for $j \in \{1, \ldots, 2q\}$, $y_{2q+1} = 2K$, and $y_{2q+2} = 0$. A fair division instance $\mathcal{I}'$ is defined with $n = 2$ agents and a set of goods $G' = \{g_1, \ldots, g_{2q+6}\}$. Agent 2's utility is such that $u_2(g_j) = y_j$ for $j \in \{1, \ldots, 2q+2\}$, $u_2(g_{2q+3}) = u_2(g_{2q+4}) = 0$, and $u_2(g_{2q+5}) = u_2(g_{2q+6}) = 2K$. Agent 1's utility is such that $u_1(g) = u_2(g) + 4K$ for all $g \in G$. The size vector $\vec{s}'$ is $(q+2, q+4)$. In Theorem 5.3.3, it was proven that there exists an EF1 allocation with size vector $\vec{s}'$ in this instance if and only if Y can be partitioned into two multisets Y_1 and Y_2 of equal cardinalities (i.e., of size $q+1$ each) with sums $(q+3)K$ and $(q+1)K$ respectively. Both problems were proven to be NP-hard.

Define a new fair division instance $\mathcal{I}$ as follows. There are $n = 2$ agents and a set of goods $G = \{g_1, \ldots, g_{4q+12}\}$. For $j \in \{1, \ldots, 2q+6\}$, the utility of g_j for each agent is identical to that in the original fair division instance $\mathcal{I}'$. For $j \in \{2q+7, \ldots, 4q+12\}$, we have $u_i(g_j) = 0$ for $i \in \{1, 2\}$. The size vector $\vec{s}$ is $(2q+6, 2q+6)$. In the initial allocation $\mathcal{A}$, agent 1 has $A_1 = \{g_{2q+7}, \ldots, g_{4q+12}\}$ and agent 2 has $A_2 = \{g_1, \ldots, g_{2q+6}\}$. This reduction can be done in polynomial time. We claim that the optimal number of exchanges required to reach an EF1 allocation from $\mathcal{A}$ is at most $q+2$ in $\mathcal{I}$ if and only if there exists an EF1 allocation with size vector $\vec{s}'$ in $\mathcal{I}'$.

($\Leftarrow$) Let (A_1', A_2') be an EF1 allocation with size vector $\vec{s}'$ in $\mathcal{I}'$. Note that $A_1' \subseteq A_2$ and $|A_1'| = q+2$. In $\mathcal{A}$, exchange the $q+2$ goods from A_1' with any $q+2$ goods in A_1. This requires a total of $q+2$ exchanges. The new allocation has exactly the same goods as that in (A_1', A_2') along with other goods with zero utility, and so is EF1. Therefore, the optimal number of exchanges to reach an EF1 allocation from $\mathcal{A}$ is at most $q+2$.

($\Rightarrow$) Suppose that an EF1 allocation $\mathcal{B}$ is reached from $\mathcal{A}$ after $t \leq q+2$ exchanges in $\mathcal{I}$. We may assume that every good is not exchanged more than once. By the same reasoning as in the proof of Theorem 5.3.3, we must have $u_1(B_1) \geq (5q+13)K$ and $u_2(B_1) \leq (q+5)K$. Since t goods are transferred from A_2, we have $u_1(B_1) = u_2(B_1) + t(4K) \leq (q+5)K + (q+2)(4K) = (5q+13)K$. This means that the inequalities for $u_1(B_1)$ are tight, and we have $u_1(B_1) = 5q+13$ and $t = q+2$. Letting $G_1 = A_2 \cap B_1$, we have $|G_1| = q+2$ and $G_1 \subseteq A_2 \subseteq G'$. Since (B_1, B_2) is an EF1 allocation, the allocation that removes all goods with zero utility is also EF1, namely, $(G_1, G' \setminus G_1)$. This induces an EF1 allocation with size vector $\vec{s}'$ in $\mathcal{I}'$. $\square$

5.4.2 Constant Number of Agents

While a polynomial-time algorithm to compute the optimal number of exchanges exists for two agents with identical utilities, this is not the case for three or more agents unless P = NP. Indeed, we establish the NP-hardness of this problem via a reduction from Balanced Multi-Partition with $p \geq 2$, an NP-hard problem by Proposition 5.2.2.

Theorem 5.4.3. *Optimal Exchanges is NP-hard for $n \geq 3$ agents with identical utilities, even when the initial allocation is balanced.*

Proof. We shall reduce from the NP-hard problem Balanced Multi-Partition (see Proposition 5.2.2). Let a Balanced Multi-Partition instance be given with $p \geq 2$. Define a fair division instance as follows. There are $n = p+1$ agents with identical utilities, and a set of goods $G = \{g_1, \ldots, g_{n(pq+q+2)}\}$ such that $u(g_j) = x_j$ for all $j \in \{1, \ldots, pq\}$, $u(g_j) = K$ for all $j \in \{pq+1, \ldots, pq+q+2\}$, and $u(g) = 0$ for the remaining goods g. Note that every good in $G_1 = \{g_1, \ldots, g_{pq}\}$ has utility more than K and at most $2K$, every good in $G_2 = \{g_{pq+1}, \ldots, g_{pq+q+2}\}$ has utility exactly K, and every good in $G \setminus (G_1 \cup G_2)$ has zero utility. The size vector $\vec{s}$ is such that $s_i = pq+q+2$ for all $i \in N$. In the initial allocation $\mathcal{A}$, agent n has $G_1 \cup G_2$, while the remaining agents have the remaining goods. This reduction can be done in polynomial time. We claim that the optimal number of exchanges required to reach an EF1 allocation from $\mathcal{A}$ is at most pq if and only if X can be partitioned into multisets $X_1, \ldots, X_p$ of equal cardinalities and sums.

($\Leftarrow$) Let $(X_1, \ldots, X_p)$ be such a partition. For each $j \in \{1, \ldots, pq\}$, if x_j is in X_i for some $i \in \{1, \ldots, p\}$, exchange g_j in agent n's bundle with a zero-utility good in agent i's bundle. After pq exchanges, each agent $i \in \{1, \ldots, p\}$ has q goods corresponding to the integers in X_i along with other goods with zero utility, and the total utility of these goods is $(q+1)K$. Meanwhile, agent n has G_2 along with other goods with zero utility. There are $q+2$ goods with utility K each, and so the utility of agent n's bundle without a most valuable good is $(q+1)K$. This shows that the resulting allocation is EF1. Therefore, the optimal number of exchanges required to reach an EF1 allocation from $\mathcal{A}$ is at most pq.

($\Rightarrow$) Suppose that the optimal number of exchanges required to reach an EF1 allocation from $\mathcal{A}$ is at most pq. Let $\mathcal{B}$ be one such EF1 allocation after these exchanges. Since at most pq exchanges were made, agent n has at least $|G_1 \cup G_2| - pq = q+2$ goods from $G_1 \cup G_2$ in $\mathcal{B}$. Every good in $G_1 \cup G_2$ has utility at least K, so the utility of agent n's bundle in $\mathcal{B}$ without a most valuable good is at least $(q+1)K$. For every agent to be EF1 towards agent n, they must each have a utility of at least $(q+1)K$ in $\mathcal{B}$. The total utility of agent 1 to agent p's bundles is therefore at least $p(q+1)K$, which is exactly the utility of G_1. Since every good in G_1 has a higher utility than every good in G_2, this implies that the only possibility is that all the goods in G_1 go to agents 1 to p, leaving G_2 (along with other goods with zero utility) with agent n. This means that agent n's bundle in $\mathcal{B}$ without a most valuable good has utility *exactly* $(q+1)K$, and that the goods in G_1 must be split among agents 1 to p so that every agent receives a utility of $(q+1)K$ from G_1. Furthermore, none of these agents can receive more than q goods from G_1; otherwise, if some agent receives at least $q+1$ goods from G_1, then the utility of her bundle is more than $(q+1)K$, which leaves another agent with utility less than $(q+1)K$, a contradiction. Therefore, agent 1 to p each receives exactly q goods from G_1. Hence, it is possible to partition the goods in G_1 into p bundles so that each bundle has exactly q goods and the utility of each bundle is exactly $(q+1)K$. This induces a partition of X into p multisets of equal cardinalities and sums. □

Next, we consider binary utilities. We have shown that deciding whether the optimal number of exchanges to reach an EF1 allocation is finite is in P (Theorem 5.3.7). We now show that the same is true for computing this exact number. To this end, we first prove that finding the optimal number of exchanges between two equivalence classes of allocations can be done efficiently.

Lemma 5.4.4. *Let an instance with n agents with binary utilities be given, where n is a constant. Then, there exists an algorithm running in time polynomial in m that computes the optimal number of exchanges required to reach some allocation in a given equivalence class from another given allocation.*

Proof. If the size vectors of the given allocation and an allocation in the given equivalence class are different, then the optimal number of exchanges is ∞. Therefore, we may henceforth assume that the size vectors are the same.

For each agent i and each type of good c in the given initial allocation, we have an n-vector such that the j^{th} component of this vector represents the number of goods of type c that need to be moved from agent i in the initial allocation to agent j in some final allocation in the given equivalence class under some exchange. Since the number of goods of each type is an integer between 0 and m, there are $m+1$ possible numbers for each component, and hence at most $(m+1)^n$ possible vectors to represent this information. The movement of goods from the initial allocation to the final allocation can be represented by an ordered collection of such vectors over all agents and over all types of goods in each agent's initial allocation. Since there are n agents and 2^n types of goods, there are at most $((m+1)^n)^{n\cdot 2^n}$ such collections. Since $((m+1)^n)^{n\cdot 2^n}$ is polynomial in m whenever n is a constant, there are at most a polynomial number of such movements between the two allocations. For each of these movements, we can verify in polynomial time whether it indeed gives some final allocation in the equivalence class. We thus have an enumeration of all such feasible movements in polynomial time.

For each feasible movement between the initial allocation $\mathcal{A}$ and the final allocation $\mathcal{B}$, define a directed graph where the vertices are the agents and each edge e_g represents a good such that if $g \in A_i \cap B_j$, then $e_g = (i, j)$. Proposition 4.4.1 showed that the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}$ is $m - c^*$, where c^* is the maximum number of disjoint cycles in the item exchange graph. Note that each cycle consists of a subset of the agents in some order, so the number of cycle types, L, is at most $(n+1)!$, which is a constant. We have a L-vector such that the k^{th} component of this vector represents the number of cycles in the item exchange graph of cycle type k. Since the number of cycles of each cycle type is an integer between 0 and m, there are $m+1$ possible numbers for each component, and hence at most $(m+1)^L$ possible vectors to represent the cycles in the graph, which is polynomial in m. We can enumerate all such vectors, consider only those vectors that represent the item exchange graph, and output the maximum number of disjoint cycles from these vectors as c^* in polynomial time. Then, calculating $m - c^*$ gives the optimal number of exchanges for that feasible movement of goods. Finally, the minimum optimal number of exchanges across all feasible movements is the quantity we desire. $\square$

This lemma, together with Lemma 5.3.6, yields the result.

Theorem 5.4.5. *Optimal Exchanges is in P for a constant number of agents with binary utilities.*

Proof. We use the algorithm in Lemma 5.3.6 to enumerate all the possible equivalence classes of EF1 allocations with the given size vector in polynomial time. For each equivalence class of allocations, we use the algorithm in Lemma 5.4.4 to compute the optimal number of exchanges required to reach some allocation from this equivalence class from the given initial allocation in polynomial time. We then return the smallest such number. □

5.4.3 General Number of Agents

Let us now consider a non-constant number of agents. We have shown that computing the optimal number of exchanges required to reach an EF1 allocation is NP-hard, even for identical utilities (Theorem 5.4.3). We thus consider binary utilities.

Although this problem is in P when the number of agents is a constant (Theorem 5.4.5), we show that it is NP-hard for a general number of agents, even when we consider the special case where the initial allocation is balanced. For this, we reduce from Exact Cover by 3-Sets (X3C), an NP-hard problem (Garey and Johnson, 1979, p. 221).

Theorem 5.4.6. *Optimal Exchanges is NP-hard for binary utilities, even when the initial allocation is balanced.*

Proof. We shall reduce from X3C. In X3C, we are given positive integers p and q, a set $X = \{x_1, \dots, x_{3q}\}$, and a collection $C = \{Y_1, \dots, Y_p\}$ of three-element subsets of X, i.e., for each $j \in \{1, \dots, p\}$, $|Y_j| = 3$ and $Y_j \subseteq X$. The problem is to decide whether there exists an exact cover for X in C, i.e., whether there exists $D \subseteq C$ such that $|D| = q$ and for all $x \in X$, there exists $Y \in D$ such that $x \in Y$. This decision problem is known to be NP-hard (Garey and Johnson, 1979, p. 221).

Let an X3C instance be given. Note that if there exists an exact cover D for X, and if some $x' \in X$ appears in exactly one $Y' \in C$, then Y' must be in D, and moreover, the other two elements in $Y' \setminus \{x'\}$ must not appear in any other three-element sets in D. In this case, we can reduce the problem further by considering the set $X \setminus Y'$ and the collection $\{Y_j \in C \mid Y_j \cap Y' = \varnothing\}$ instead. Therefore, assume without loss of generality that each $x \in X$ appears in at least two three-element sets in C.

Define a fair division instance as follows. There are $n = 3q+1$ agents with binary utilities, and a set of goods $G = \{g_{i,j} \mid i \in \{1, \dots, 3q+1\}, j \in \{1, \dots, p\}\}$. For notational simplicity, let $h_j = g_{3q+1,j}$ for all $j \in \{1, \dots, p\}$; the good h_j is associated with Y_j. The size vector is $\vec{s} = (p, \dots, p)$. In the initial allocation $\mathcal{A} = (A_1, \dots, A_{3q+1})$, we have $A_i = \{g_{i,j} \mid j \in \{1, \dots, p\}\}$ for all $i \in N$. Agent $3q+1$ is a special agent and assigns zero utility to every good. For each non-special agent $i \in \{1, \dots, 3q\}$, let n_i be the number of three-element subsets in C that contain x_i, i.e., $n_i = |\{Y \in C \mid x_i \in Y\}|$. By assumption, we have $n_i \geq 2$. Then, agent i values exactly $n_i - 2$ goods in A_i, e.g., $u_i(g_{i,j}) = 1$ for $j \in \{1, \dots, n_i - 2\}$ and $u_i(g_{i,j}) = 0$ for $j \in \{n_i - 1, \dots, p\}$. Agent i also values the goods associated with any Y_j that contains x_i, i.e., $u_i(h_j) = 1$ if and only if $x_i \in Y_j$. Agent i assigns zero utility to every other good not mentioned. Note that in the initial allocation, from each non-special agent i's perspective, the utility of agent i's bundle is $n_i - 2$, the utility of agent $(3q+1)$'s bundle is n_i, and the utilities of the other agents' bundles are zero. This reduction can be done in polynomial time. We claim that the optimal number of exchanges required to reach an EF1 allocation from $\mathcal{A}$ is at most q if and only if there exists an exact cover for X in C.

($\Leftarrow$) Let D be an exact cover for X. For each $Y_j \in D$, we perform one exchange as follows: select any $x_i \in Y_j$ arbitrarily, and exchange $g_{i,p}$ in agent i's bundle with h_j in agent $(3q+1)$'s bundle. Note that there are exactly q exchanges, since $|D| = q$. We claim that the final allocation is EF1. Since agent $3q+1$ does not value any good, she is EF1 towards every other agent. Therefore, we only need to consider agent i's envy for $i \in \{1, \dots, 3q\}$. Note that there exists $j \in \{1, \dots, p\}$ such that $x_i \in Y_j$ and $Y_j \in D$. This means that h_j is moved to some non-special agent's bundle in an exchange (possibly agent i). Regardless of whom h_j goes to, agent i's utility for her own bundle is at least $n_i - 2$, and agent i's utility for agent $(3q+1)$'s bundle is exactly $n_i - 1$, so agent i is EF1 towards agent $3q+1$. Furthermore, if h_j goes to some agent $i' \neq i$, then agent i's utility for the bundle of agent i' is 1, so agent i is EF1 towards agent i'. Every other non-special agent's bundle yields zero utility to agent i. This shows that agent i is EF1 towards every other agent. Accordingly, the final allocation is EF1.

($\Rightarrow$) Suppose that after at most q exchanges, an EF1 allocation is reached. Let $i \in \{1, \dots, 3q\}$. The valuable goods from agent i's perspective are with agent i or with agent $3q+1$. Since agent i's utility for her own bundle in $\mathcal{A}$ is $n_i - 2$ and her utility for agent $(3q+1)$'s bundle is n_i, some valuable good from agent $(3q+1)$'s bundle needs to be moved to another agent's bundle (possibly i's) in an exchange. Now, each good in agent $(3q+1)$'s bundle is valuable to exactly three agents. Since the movement of each good in agent $(3q+1)$'s bundle can only resolve the envy for at most three agents, at least q goods need to be moved to make agents 1 to $3q$ EF1. This means that exactly q exchanges are made; moreover, each good h_j moved from agent $(3q+1)$'s bundle is associated with three distinct agents. The set of q goods moved from agent $(3q+1)$'s bundle induces an exact cover D with cardinality q. □

Finally, we consider identical binary utilities. We show that for this restricted class of utilities, the computational problem can be solved efficiently regardless of whether the size vector of the initial allocation is balanced or not.

Theorem 5.4.7. *Optimal Exchanges is in P for identical binary utilities.*

Proof. Let $\mathcal{A} = (A_1, \dots, A_n)$ be the given allocation. As mentioned at the beginning of Section 5.4, we assume that an EF1 allocation can be reached from $\mathcal{A}$. By the proof of Theorem 5.3.10, there must be at most $s_0 n + n - n_0$ valuable goods, where $s_0 = \min_{i \in N} |A_i|$ and $n_0 = |\{i \in N \mid |A_i| = s_0\}|$. Suppose that there are $m_1 \leq s_0 n + n - n_0$ valuable goods. An EF1 allocation requires every agent to receive at least $F := \lfloor m_1/n \rfloor$ valuable goods and at most $F+1$ valuable goods. Since an EF1 allocation with the same size vector as $\mathcal{A}$ exists, every agent must have at least F goods in $\mathcal{A}$, i.e., $s_0 \geq F$. Let N_0 be the set of agents who have at most F valuable goods in the initial allocation, and N_1 be the set of agents who have at least $F+1$ valuable goods in the initial allocation. Note that $N = N_0 \cup N_1$. Let $c_0 = \sum_{i \in N_0}(F - u(A_i))$ and $c_1 = \sum_{i \in N_1}(u(A_i) - (F+1))$. We claim that the optimal number of exchanges required to reach an EF1 allocation is $\max\{c_0, c_1\}$. Note that this value can be computed in polynomial time, so it suffices to prove the claim.

First, we show that the optimal number of exchanges required to reach an EF1 allocation is at least $\max\{c_0, c_1\}$. Each agent $i \in N_0$ needs to receive at least $F - u(A_i) \geq 0$ valuable goods in order to arrive at a bundle with utility at least F. In receiving these valuable goods, agent i must give away the same number of non-valuable goods from her bundle in A_i—note that this is possible since every agent has at least F goods. Therefore, there exist at least $F - u(A_i)$ valuable goods from other agents' bundles that should go to agent i's bundle and at least $F - u(A_i)$ non-valuable goods from agent i's bundle that should go to other agents' bundles. Summing up over all $i \in N_0$, we have that at least $\sum_{i \in N_0} 2(F - u(A_i)) = 2c_0$ goods are in the wrong hands. Since each exchange places at most two goods in correct hands, the number of exchanges required is at least $2c_0/2 = c_0$. By an analogous argument on the agents in N_1, we have that the number of exchanges required is at least c_1. This proves that the optimal number of exchanges required to reach an EF1 allocation is at least $\max\{c_0, c_1\}$.

Next, we describe an algorithm that allows us to reach an EF1 allocation with at most $\max\{c_0, c_1\}$ exchanges. The algorithm is as follows: repeatedly exchange a valuable good from an agent with the highest utility with a non-valuable good from an agent with the lowest utility, until every agent has at least F valuable goods and at most $F + 1$ valuable goods. We show that this ending will always be reached. Suppose on the contrary that this is not the case, and consider the final allocation just before the algorithm cannot proceed further. Since every agent has at least F goods in $\mathcal{A}$, it must be possible that every agent receives at least F valuable goods in the final allocation, and so $F = s_0$. This means that some agent has more than $F + 1$ valuable goods in the final allocation, and every agent who has F goods in $\mathcal{A}$ has F valuable goods in the final allocation. Then, the number of valuable goods is $m_1 > Fn_0 + (F + 1)(n - n_0) = s_0 n + n - n_0$, which is a contradiction. This shows that it is possible to reach the desired ending.

Now, we are ready to show that the optimal number of exchanges required is at most $\max\{c_0, c_1\}$. If $c_0 \geq c_1$, then the first c_1 exchanges involve exchanging valuable goods from agents in N_1 with non-valuable goods from agents in N_0. At this point, every agent in N_1 has exactly $F + 1$ valuable goods, and every agent in N_0 has at most F valuable goods. Call this allocation $(B_1, \ldots, B_n)$. We have that $\sum_{i \in N_0}(F - u(B_i)) = c_0 - c_1$. If $|N_1| < c_0 - c_1$, then after $|N_1|$ further exchanges, we have that every agent has at most F valuable goods and some agent has fewer than F valuable goods, contradicting the assumption that $F = \lfloor m_1/n \rfloor$. Therefore, we must have $|N_1| \geq c_0 - c_1$. Now, after $c_0 - c_1$ further exchanges, every agent in N_0 has exactly F valuable goods each and every agent in N_1 has between F and $F + 1$ valuable goods (inclusive), giving an EF1 allocation. Hence, if $c_0 \geq c_1$, then the optimal number of exchanges required to reach an EF1 allocation is at most $c_1 + (c_0 - c_1) = c_0$. By an analogous argument, if $c_0 < c_1$, then the optimal number of exchanges required to reach an EF1 allocation is at most c_1. It follows that the optimal number of exchanges required to reach an EF1 allocation is at most $\max\{c_0, c_1\}$. □

5.5 Worst-Case Bounds

In this section, instead of instance-specific optimization, we turn our attention to the *worst-case* number of exchanges required to reach an EF1 allocation from some initial allocation. Since an EF1 allocation may not always be reachable (as can be seen from Section 5.3), we shall focus on the special case where the number of goods in each agent's bundle is the same. We say that a size vector $\vec{s} = (s_1, \dots, s_n)$ is *s-balanced* for a positive integer s if $s_i = s$ for all $i \in N$, and an allocation is *s-balanced* if it has an s-balanced size vector.

We shall examine the worst-case number of exchanges for identical binary utilities and for general utilities separately.

5.5.1 Identical Binary Utilities

Given n and s, let $f_{\text{id,bin}}(n, s)$ be the smallest integer such that for every instance with n agents with *identical binary utilities* and ns goods and every s-balanced allocation $\mathcal{A}$ in the instance, there exists an EF1 allocation that can be reached from $\mathcal{A}$ using at most $f_{\text{id,bin}}(n, s)$ exchanges. We show that $f_{\text{id,bin}}(n, s)$ is roughly $sn/4$.

Theorem 5.5.1. *Let n and s be positive integers. If n is even, then*

$$\frac{n}{2}\left\lfloor\frac{s}{2}\right\rfloor \leq f_{\text{id,bin}}(n, s) \leq \frac{sn}{4}.$$

If n is odd, then

$$\frac{n+1}{2}\left\lfloor\frac{s(n-1)}{2n}\right\rfloor \leq f_{\text{id,bin}}(n, s) \leq \frac{s(n-1)(n+1)}{4n}.$$

Proof. Recall that the proof of Theorem 5.4.7 provided a way to compute the optimal number of exchanges to reach an EF1 allocation from a given initial allocation. To recap, let m_1 be the total number of valuable goods, $F = \lfloor m_1/n \rfloor$ be the minimum number of valuable goods each agent must receive in an EF1 allocation, N_0 be the set of agents who have at most F valuable goods in the initial allocation, N_1 be the set of agents who have at least $F+1$ valuable goods in the initial allocation, $c_0 = \sum_{i \in N_0}(F - u(A_i))$, and $c_1 = \sum_{i \in N_1}(u(A_i) - (F+1))$. The optimal number of exchanges is $\max\{c_0, c_1\}$.

We first prove the lower bounds for $f_{\text{id,bin}}(n, s)$ by providing an explicit initial allocation and showing that the optimal number of exchanges to reach an EF1 allocation is at least $\lfloor \lfloor n/2 \rfloor s/n \rfloor \cdot \lceil n/2 \rceil$, which corresponds to the lower bounds for both even and odd n. In the initial allocation, $\lfloor n/2 \rfloor$ agents have s valuable goods each and the remaining $\lceil n/2 \rceil$ agents have s non-valuable goods each. There are a total of $m_1 = \lfloor n/2 \rfloor \cdot s$ valuable goods, and $F = \lfloor \lfloor n/2 \rfloor s/n \rfloor$. The value of c_0 is $\sum_{i \in N_0}(F - u(A_i)) = \sum_{i \in N_0}(\lfloor \lfloor n/2 \rfloor s/n \rfloor - 0) = \lfloor \lfloor n/2 \rfloor s/n \rfloor \cdot \lceil n/2 \rceil$. Since $\max\{c_0, c_1\} \geq c_0$, the lower bounds follow.

We now prove the upper bounds for $f_{\text{id,bin}}(n, s)$. Let an s-balanced allocation $\mathcal{A} = (A_1, \dots, A_n)$ be given, and let $n_0 = |N_0|$ and $n_1 = |N_1|$. We first derive upper bounds for c_0 and c_1. Note that $m_1 \leq sn_1 + \sum_{i \in N_0} u(A_i)$. We have

$$\begin{aligned}
c_0 &= \sum_{i \in N_0}(F - u(A_i)) \\
&= (n - n_1)F - \sum_{i \in N_0} u(A_i) \\
&\leq (n - n_1)\frac{m_1}{n} - \sum_{i \in N_0} u(A_i)
\end{aligned}$$

$$
\begin{aligned}
&= \left(1 - \frac{n_1}{n}\right) m_1 - \sum_{i \in N_0} u(A_i) \\
&\le \left(1 - \frac{n_1}{n}\right) \left(s n_1 + \sum_{i \in N_0} u(A_i) \right) - \sum_{i \in N_0} u(A_i) \\
&\le s n_1 - \frac{s n_1^2}{n} + \sum_{i \in N_0} u(A_i) - \sum_{i \in N_0} u(A_i) \\
&= \frac{s n_1}{n}(n - n_1).
\end{aligned}
$$

On the other hand, $m_1 \ge \sum_{i \in N_1} u(A_i)$. We have

$$
\begin{aligned}
c_1 &= \sum_{i \in N_1} (u(A_i) - (F+1)) \\
&= \sum_{i \in N_1} u(A_i) - n_1 \left(\left\lfloor \frac{m_1}{n} \right\rfloor + 1 \right) \\
&\le \sum_{i \in N_1} u(A_i) - n_1 \left(\frac{m_1}{n} \right) \\
&\le \sum_{i \in N_1} u(A_i) - \frac{n_1}{n} \sum_{i \in N_1} u(A_i) \\
&= \frac{1}{n}(n - n_1) \sum_{i \in N_1} u(A_i) \\
&\le \frac{1}{n}(n - n_1) s n_1 \\
&= \frac{s n_1}{n}(n - n_1).
\end{aligned}
$$

We have thus shown that $\max\{c_0, c_1\} \le n_1(n - n_1)s/n$. Therefore, the optimal number of exchanges is at most $n_1(n - n_1)s/n$, which is a quadratic expression in n_1. When n is even, $n_1(n - n_1)s/n$ attains a maximum value at $n_1 = n/2$, and this value is $sn/4$. When n is odd, $n_1(n - n_1)s/n$ attains a maximum value at $n_1 = (n+1)/2$ and $n_1 = (n-1)/2$, and this value is $s(n-1)(n+1)/4n$. The upper bounds for $f_{\text{id,bin}}(n, s)$ follow. $\square$

Our result show that the number of exchanges in the worst-case scenario for identical binary utilities is equal to one-quarter of the total number of goods. In other words, roughly half of the goods need to be exchanged in the worst-case scenario.

5.5.2 General Utilities

We now proceed to utilities that are not necessarily identical nor binary. Given n and s, let $f(n, s)$ be the smallest integer such that for every instance with n agents and ns goods and every s-balanced allocation $\mathcal{A}$ in the instance, there exists an EF1 allocation that can be reached from $\mathcal{A}$ using at most $f(n, s)$ exchanges. We shall examine the bounds for $f(n, s)$.

We first derive an upper bound for $f(n, s)$. At a high level, we use an algorithm by Biswas and Barman (2018) to find an EF1 allocation under cardinality constraints such that every agent retains roughly s/n of her goods from her original bundle. The algorithm also distributes the goods in each agent's initial bundle to the other agents as evenly as possible in order to maximize the number of goods that can be exchanged one-to-one, thereby minimizing the total number of exchanges. One can check that roughly $s(n-1)/2$ exchanges are required to reach this EF1 allocation from the initial allocation.

Theorem 5.5.2. *Let n and s be positive integers, and let $q = \lfloor s/n \rfloor$ and $r = s - qn$ be the quotient and remainder when s is divided by n, respectively. Then,*

$$f(n, s) \leq \begin{cases} s(n-1)/2 & \text{if } r = 0, \\ s(n-1)/2 + r(n-3)/2 + 1 & \text{otherwise.} \end{cases}$$

Moreover, we have $f(2, s) \leq (s - r)/2$ for all s.

Proof. Let $\mathcal{A}$ be an s-balanced allocation. It suffices to find an EF1 s-balanced allocation $\mathcal{B}$ such that the optimal number of exchanges to reach $\mathcal{B}$ from $\mathcal{A}$ is at most the expression given in the theorem statement.

When $n = 2$, allocate the goods in A_1 to the two agents in a round-robin fashion with agent 1 going first, and allocate the goods in A_2 to the two agents in a round-robin fashion with agent 2 going first. Call this new allocation $\mathcal{B}$. Note that $\mathcal{B}$ is clearly s-balanced. We have $A_i \cap B_{3-i} = (s - r)/2$ for $i \in \{1, 2\}$, so the optimal number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}$ is (exactly) $(s - r)/2$. To see that $\mathcal{B}$ is EF1, observe that agent 1 does not envy agent 2 with respect to the goods chosen from A_1 and is EF1 towards agent 2 with respect to the goods chosen from A_2, so agent 1 is EF1 towards agent 2 in $\mathcal{B}$; likewise, agent 2 is EF1 towards agent 1 in $\mathcal{B}$. This shows that $f(2, s) \leq (s - r)/2$.

When $n \geq 3$, we shall find an EF1 s-balanced allocation $\mathcal{B}$ by generalizing the method for two agents. We define $n + r$ categories of goods $C_1, \dots, C_n, D_1, \dots, D_r$ as follows. For $i \in N$, category C_i contains qn goods arbitrarily selected from A_i only; note that r goods remain unselected in A_i. Next, we form D_w recursively as follows: let $w \in \{1, \dots, r\}$ be the smallest index such that D_w does not have n goods yet, let $i \in N$ be the smallest index such that A_i still has unselected goods, arbitrarily select a good in A_i, and add it to D_w. At the end of this process, every category C_i has exactly qn goods from A_i, and every category D_w has exactly n goods from consecutive agents' bundles, say, $A_{i_w}, A_{i_w+1}, \dots, A_{j_w}$.

We now proceed to form $\mathcal{B}$ using the algorithm by Biswas and Barman (2018) which finds an EF1 allocation under cardinality constraints. In particular, there exists an EF1 allocation $\mathcal{B} = (B_1, \dots, B_n)$ such that $|C_i \cap B_j| = |C_i|/n = q$ for all $i, j \in N$ and $|D_w \cap B_j| = |D_w|/n = 1$ for all $w \in \{1, \dots, r\}$, $j \in N$. Also, $\mathcal{B}$ is s-balanced because $|B_j| = qn + r = s$ for all $j \in N$. We shall bound the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}$.

For each unordered pair of distinct $i, j \in N$, exchange the q goods from $C_i \cap B_j$ (which are in A_i) with the q goods from $C_j \cap B_i$ (which are in A_j). This requires a total of $qn(n-1)/2$ exchanges. Call this intermediate allocation $\mathcal{A}' = (A'_1, \dots, A'_n)$. At this point, the only goods that are possibly in the wrong bundles in $\mathcal{A}'$ (as compared to $\mathcal{B}$) are the goods in all the D_w, and there are at most rn such goods. For each $i \in N$, let $X_i = A'_i \cap (D_1 \cup \cdots \cup D_r)$.

If $r = 0$, then $\mathcal{A}' = \mathcal{B}$, and we are done since the total number of exchanges is $qn(n-1)/2 = s(n-1)/2$. Else, $r > 0$. Consider the directed graph where the vertices are the agents and each edge e_g represents a good $g \in G$ such that if $g \in A'_i \cap B_j$, then $e_g = (i, j)$. Proposition 4.4.1

showed that the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}'$ is $m - c^*$, where c^* is the maximum possible cardinality of a partition of the edges of the graph into (directed) circuits. In $\mathcal{A}'$, qn^2 goods from all the C_i are in the correct bundle by the previous process, and the edges representing these goods each has its own circuit, say, (i, i) if the good is in A'_i. We shall show that the edges representing the rn goods in all the D_w can be partitioned into at least $2r - 1$ disjoint circuits. This will give at least $qn^2 + (2r - 1) = sn - (rn - 2r + 1)$ as the cardinality of one such partition of the edges of the graph into circuits. Accordingly, $c^* \geq sn - (rn - 2r + 1)$, and the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}'$ is $m - c^* \leq rn - 2r + 1$. Then, the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}$ (via $\mathcal{A}'$) is at most $qn(n-1)/2 + (rn - 2r + 1) = s(n-1)/2 + r(n-3)/2 + 1$, establishing the theorem.

Let $w \in \{1, \dots, r\}$ be given. We shall show that there exists a cycle formed with a subset of the edges representing the goods in D_w. The goods in D_w come from consecutive agents' bundles in $\mathcal{A}'$, say, agents i_w to j_w. Every agent receives exactly one good from D_w in $\mathcal{B}$; in particular, agents i_w to j_w each receives exactly one good from D_w. Consider the good g in $D_w \cap B_{i_w}$. If g is in X_{i_w}, then the edge $e_g = (i_w, i_w)$ is a desired cycle. Otherwise, g belongs to some agent $i' \in \{i_w + 1, \dots, j_w\}$ in $\mathcal{A}'$. Then, the edge e_g is (i', i_w). Next, we consider the good g' in $D_w \cap B_{i'}$, and find the agent that has g' in $\mathcal{A}'$. The edge representing g' then points to i' from that agent. By repeating this, we eventually find a cycle formed with some of these edges and with a subset of the agents i_w to j_w as vertices. Let $G_w \subseteq D_w$ be the set of goods that are represented by the edges in this cycle. Note that each X_i for $i \in \{i_w, \dots, j_w\}$ contains at most one good in G_w, and each X_i for $i \in N \setminus \{i_w, \dots, j_w\}$ does not contain any good in G_w.

Now, consider the goods represented by the edges of the r cycles—one for each w. Note that these cycles are disjoint since the sets G_w are pairwise disjoint. Let $G_0 = \bigcup_{w=1}^{r} G_w$. We claim that $|G_0| < 2n$. Since the r goods in X_1 are entirely contained in D_1, we have $|X_1 \cap G_1| \leq 1$ and $|X_1 \cap G_w| = 0$ for $w \in \{2, \dots, r\}$, which implies that $|\bigcup_{w=1}^{r}(X_1 \cap G_w)| \leq 1$. Now, for each $i \in N \setminus \{1\}$, the r goods in X_i can only be contained in at most two D_w—to see this, observe that if the r goods are contained in $D_{w'}$, $D_{w'+1}$, and $D_{w'+2}$, then $D_{w'+1} \subseteq X_i$, which implies that $r = |X_i| \geq |D_{w'+1}| = n$, a contradiction. Thus, we have $|X_i \cap G_w| \leq 1$ for all $w \in \{1, \dots, r\}$, and $|X_i \cap G_w| = 1$ for at most two w, and so $|\bigcup_{w=1}^{r}(X_i \cap G_w)| \leq 2$. Since $G_0 = \bigcup_{i \in N} \bigcup_{w=1}^{r}(X_i \cap G_w)$, we have $|G_0| \leq 1 + (n-1) \cdot 2 < 2n$, proving the claim.

Finally, consider the edges representing the rn goods in all the D_w. We have shown that fewer than $2n$ of these edges can be used to form r disjoint circuits (in fact, cycles). There are more than $rn - 2n = (r-2)n$ edges remaining. Since we can always require every circuit to have length at most n, there exists a partition of the remaining edges into more than $r - 2$ disjoint circuits, i.e., at least $r - 1$ disjoint circuits. The total number of circuits in this partition is at least $r + (r - 1) = 2r - 1$. This completes the proof. □

If no good is involved in more than one exchange, then $s(n-1)/2$ exchanges means that a total of $s(n-1) = m(1 - 1/n)$ goods are exchanged. When n is large, the fraction of goods involved in the exchanges becomes close to 1. While this bound might not seem impressive, we show next that it is, in fact, already essentially tight. Specifically, we establish a lower bound for $f(n, s)$ by constructing an instance (with binary utilities) and an s-balanced allocation

$\mathcal{A}$ in the instance such that roughly $s(n-1)/2$ exchanges are necessary to reach an EF1 allocation from $\mathcal{A}$.

Theorem 5.5.3. *Let n and s be positive integers, and let $q = \lfloor s/n \rfloor$ and $r = s - qn$ be the quotient and remainder when s is divided by n, respectively. Then,*

$$f(n,s) \geq \begin{cases} s(n-1)/2 & \text{if } r = 0, \\ s(n-1)/2 - (n-r)/2 & \text{otherwise.} \end{cases}$$

Proof. Let $G = \{g_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq s\}$ be the set of goods such that each good $g_{i,j}$ is worth 0 to agent i and worth 1 to all agents except i. We have $u_i(G) = s(n-1)$. We claim that an EF1 allocation requires every agent to receive a bundle worth at least $s - q - \lceil r/n \rceil$ from her perspective. To see this, suppose on the contrary that some agent i receives a bundle worth less than $s - q - \lceil r/n \rceil$ to her. For the allocation to be EF1, every other agent receives a bundle worth at most $s - q - \lceil r/n \rceil$ to agent i. Then, we must have $u_i(G) < n(s - q - \lceil r/n \rceil)$. When $r = 0$, it holds that $\lceil r/n \rceil = 0$ and $n(s-q-\lceil r/n \rceil) = n(s-q) = sn-s = s(n-1)$. When $r > 0$, it holds that $\lceil r/n \rceil = 1$ and $n(s-q-\lceil r/n \rceil) = n(s-q-1) = sn-(qn+r)-(n-r) = s(n-1)-(n-r) \leq s(n-1)$. In both cases, we have $u_i(G) < s(n-1) = u_i(G)$, a contradiction.

Let $\mathcal{A}$ be the allocation such that $A_i = \{g_{i,j} \mid 1 \leq j \leq s\}$ for every i. In order to reach an EF1 allocation, each agent must give away at least $s - q - \lceil r/n \rceil$ goods from her bundle in order to receive from the other agents the same number of valuable goods from her perspective. The total number of goods that are currently in the wrong hands across all agents is at least $n(s-q-\lceil r/n \rceil)$, and the optimal number of exchanges required to reach an EF1 allocation is at least half of this number, since each exchange places at most two goods in the correct hands. When $r = 0$, the optimal number of exchanges required is at least $n(s - q - \lceil r/n \rceil)/2 = s(n-1)/2$. When $r > 0$, the optimal number of exchanges required is at least $n(s - q - \lceil r/n \rceil)/2 = s(n-1)/2 - (n-r)/2$. ☐

For two agents, Theorems 5.5.2 and 5.5.3 give a tight bound of $f(2,s) = (s-r)/2 = m/4 - r/2 = \lfloor m/4 \rfloor$. This means that in the worst-case scenario, the number of exchanges required to reach an EF1 allocation is roughly one-quarter of the total number of goods between the two agents, or equivalently, roughly half of the goods need to be exchanged between the two agents to reach an EF1 allocation.

Theorems 5.5.2 and 5.5.3 also give a tight bound of $f(n,s) = s(n-1)/2$ whenever s is divisible by n. By observing the proof of Theorem 5.5.2, we can achieve an EF1 allocation with $f(n,s)$ exchanges without involving each good in more than one exchange. This means that a $(1-1/n)$ fraction of all goods need to be exchanged in the worst-case scenario. Intuitively, this happens when each agent only values the goods in the bundle of every agent except her own in the initial allocation, and therefore needs to ensure that these goods are evenly distributed among all agents including herself.

5.6 Conclusion

In this chapter, we have studied the reformability of unfair allocations and the number of exchanges required in the reformation process. We revealed several distinctions in the complexity of these problems based on the number of agents and their utility functions, and showed that the number of exchanges required to reach an EF1 allocation is relatively high in the worst case.

One could ask whether the hardness results in Section 5.4 still hold if we only require the computation to be correct up to a certain factor. While our worst-case bounds for general utilities are already exactly tight in certain scenarios and almost tight generally, an open question is to tighten them for more than two agents when the number of goods in each agent's bundle is not divisible by the number of agents. Additionally, although these bounds also work for binary utilities, one could try to derive bounds for *identical* utilities—we provide some insights in Appendix B.1. Another interesting direction is to require each exchange to be beneficial for both agents involved—in Appendix B.2, we show that the problem of deciding whether a given initial allocation can be reformed into an EF1 allocation using only beneficial exchanges is NP-complete for binary utilities. Finally, beyond EF1, one could consider reforming an allocation using other notions as fairness benchmarks.

Part II

Divisible Goods

Chapter 6

On Connected Strongly-Proportional Cake-Cutting

6.1 Introduction

Consider a group of siblings who inherited a land estate and would like to divide it fairly among themselves. The simplest procedure for attaining a fair division is to sell the land and divide the proceeds equally; this procedure guarantees each sibling a proportional share of the total land value.

But in some cases, it is possible to give each sibling a much better deal. As an example, suppose that the land estate contains one part that is fertile and arable, and one part that is barren but has potential for coal mining. This land is to be divided between two siblings, one of whom is a farmer and the other is a coal factory owner. If we give the former piece of land to the farmer and the latter piece of land to the coal factory owner, both siblings will feel that they receive more than half of the total land value. Our main question of interest is: when is such a superior allocation possible?

We study this question in the framework of *cake-cutting*. In this setting, there is a divisible resource called a *cake*, which can be cut into arbitrarily small pieces without losing its value. The cake is represented simply by an interval which can model a one-dimensional object, such as time. There are n agents, each of whom has a personal measure of value over the cake. The goal is to partition the cake into n pieces and allocate one piece per agent such that the agents feel that they receive a "fair share" according to some fairness notion.

A common fairness criterion—nowadays called *proportionality*—requires that each agent i receives a piece of cake that is worth, according to i's valuation, at least $1/n$ of the total cake value. In his seminal paper, Steinhaus (1948) described an algorithm, developed by his students Banach and Knaster, that finds a proportional allocation; moreover, this allocation is *connected*—each agent receives a single contiguous part of the cake. This algorithm is now called the *last diminisher* algorithm.

But the guarantee of proportionality allows for the possibility that each agent receives a piece worth *exactly* $1/n$; when this is the case, there is little advantage in using a cake-cutting procedure over selling the land and giving $1/n$ to each partner. A stronger criterion, called *strong-proportionality* or *super-proportionality*, requires that each agent i receives a piece of cake worth *strictly more* than $1/n$ of the total cake value from i's perspective. This raises the question of when such a strongly-proportional allocation exists.

Obviously, a strongly-proportional allocation does not exist when all the agents' valuations are identical, since if any agent receives more than $1/n$ of the cake, then some other agent must receive less than $1/n$ of the cake. Interestingly, in all other cases, a strongly-proportional allocation exists. Even when *two* agents have non-identical valuations, there exists an allocation in which *all* n agents receive more than $1/n$ of the total cake value from their perspectives (Dubins and Spanier, 1961; Rebman, 1979). Woodall (1986) presented an algorithm for finding such a strongly-proportional allocation. Barbanel (1996a) generalized this algorithm to agents with unequal entitlements, and Jankó and Joó (2022) presented a simple algorithm for this generalized problem and extended it to infinitely many agents.

The problem with all these algorithms is that, in contrast to the last diminisher algorithm for proportional cake-cutting, they do not guarantee a *connected* allocation. Connectivity is an important practical consideration when allocating cakes; for example, if the cake is the availability of a meeting room by time and needs to be allocated to different teams throughout the day, then a two-hour slot is easier for a team to utilize than six disjoint twenty-minute slots. Indeed, connectivity is the most commonly studied constraint in cake-cutting literature (Stromquist, 1980; Su, 1999; Stromquist, 2008; Goldberg et al., 2020; Suksompong, 2021; Elkind et al., 2022), and relaxing this constraint may present each agent instead with a "countable union of crumbs" (Stromquist, 1980).

Thus, our main questions of interest are:

> *What are the necessary and sufficient conditions for the existence of a connected strongly-proportional cake allocation? What are the query complexities to determine these conditions?*

6.1.1 Our Results

The cake to be allocated, modeled by a unit interval $[0, 1]$, is to be divided among n agents who may have different entitlements for the cake, with the entitlements summing to 1. Each agent receives an interval of the cake that is disjoint from the other agents' intervals. Each agent has a valuation function (or utility function) on the intervals of the cake that is non-negative, finitely additive, and continuous with respect to length. In this regard, the value of a single point is zero to every agent, and we can assume without loss of generality that agents receive *closed* intervals of the cake, and that any two agents' pieces can possibly intersect at the endpoints of their respective intervals. In order to access agents' valuations in the algorithms, we allow algorithms to make eval and (right-)mark[1] queries of each agent as in the standard Robertson-Webb model (Robertson and Webb, 1998). More details of our model are provided in Section 6.2.

In Section 6.3, we consider *hungry* agents—those who have positive valuations for any part of the cake with positive length. For agents with equal entitlements, we show that

[1] We choose *right*-mark instead of the usual *left*-mark for convenience. Our algorithms still work if only left-mark queries are available (together with eval). See Appendix C.1 for a more detailed explanation.

a connected strongly-proportional allocation exists if and only if there are two agents with different r-marks for some $r \in \{1/n, 2/n, \ldots, (n-1)/n\}$, where an r-mark is a point that divides the cake into two such that the left part of the cake is worth r to that agent. This implies that the existence of such an allocation can be decided using $n(n-1)$ queries. The proof of sufficiency is constructive, so a connected strongly-proportional allocation can be computed using $O(n^2)$ queries if it exists. We also prove that any algorithm that decides whether a connected strongly-proportional allocation exists must make at least $n(n-1)/2$ queries, giving an asymptotically tight bound (within a factor of 2) of $\Theta(n^2)$. For agents with possibly unequal entitlements, we show that a lower bound number of queries to decide whether a connected strongly-proportional allocation exists is $n \cdot 2^{n-2}$. Together with a result from Section 6.4 later on the upper bound number of queries, this yields a tight bound of $\Theta(n \cdot 2^n)$ queries.

In Section 6.4, we consider agents who are not necessarily hungry. The characterization from Section 6.3 for hungry agents with equal entitlements does not work for non-hungry agents, which motivates us to find another characterization by considering permutations of agents. We show that a connected strongly-proportional allocation exists if and only if there exists a permutation of agents such that when the agents go in the order as prescribed by the permutation and make their rightmost marks worth their entitlements to each of them one after another, the mark made by the last agent does not reach the end of the cake. This result holds regardless of the agents' entitlements. While an algorithm to determine this condition requires $n \cdot n!$ queries, we show that this number can be reduced by a factor of $2^{\omega(n)}$ to $n \cdot 2^{n-1}$ via dynamic programming. We also prove a lower bound number of queries of $\Omega(n \cdot 2^n)$ to determine this condition, even for agents with equal entitlements. Therefore, for agents who are not necessarily hungry, we also obtain a tight bound of $\Theta(n \cdot 2^n)$, whether the entitlements are equal or not. A connected strongly-proportional allocation can be computed using $O(n \cdot 2^n)$ queries if it exists.

Table 6.1 summarizes of our results from Sections 6.3 and 6.4.

	hungry agents	**general agents**
equal entitlements	$\Theta(n^2)$ (Theorem 6.3.5)	$\Theta(n \cdot 2^n)$ (Theorem 6.4.7)
possibly unequal entitlements	$\Theta(n \cdot 2^n)$ (Theorem 6.3.7)	$\Theta(n \cdot 2^n)$ (Theorem 6.4.7)

Table 6.1: Number of queries required to decide the existence of a connected strongly-proportional allocation of a cake for n agents, and to compute one if it exists.

In Section 6.5, we consider a stronger fairness notion where each agent i needs to receive a connected piece of cake that is worth more than $w_i + z$ for some small z, where w_i is agent i's entitlement. We show that the number of queries needed to decide whether such an allocation exists is in $\Theta(n \cdot 2^n)$, even for hungry agents with equal entitlements. This is analogous to the results in Sections 6.3.2 and 6.4, which shows that the stronger fairness notion considered in this section does not make the problem any harder (nor easier).

In Section 6.6, we consider a connected strongly-proportional allocation of a *pie* instead of a cake, and show that no finite algorithm can decide the existence of such an allocation even for hungry agents with equal entitlements, demonstrating the intractability of the problem in this new setting.

6.1.2 Further Related Work

A weaker fairness notion of *proportionality* is well-studied in cake-cutting literature. It is known that a connected proportional allocation always exists for agents with equal entitlements and such an allocation can be computed using $\Theta(n \log n)$ queries (Steinhaus, 1948; Even and Paz, 1984; Woeginger and Sgall, 2007). Cseh and Fleiner (2020) presented an algorithm that finds a possibly non-connected proportional allocation for agents with general entitlements—in particular, their algorithm uses a finite but *unbounded* number of queries when agents have irrational entitlements. In contrast, we show that a connected *strongly-proportional* allocation may not exist, and such an allocation can be computed (if it exists) using $\Theta(n \cdot 2^n)$ queries. A number of works studied the number of *cuts* required for a proportional allocation, rather than the number of queries (Segal-Halevi, 2019; Crew et al., 2020).

A parallel line of work studied a stronger fairness notion of *super envy-freeness*: it requires, in addition to strong-proportionality, that each agent values the piece of every other agent at strictly less than $1/n$ the total cake value (Barbanel, 1996b; Webb, 1999; Chèze, 2020).

6.2 Preliminaries

Refer to the preliminaries in Sections 2.1 and 2.3. We now describe other preliminaries specific to this chapter.

An allocation $(A_1, \ldots, A_n)$ is *connected* if A_i is a single (closed) interval for each $i \in N$. We assume that $F_i(x) := u_i([0, x])$ is a continuous function on the cake $C = [0, 1]$, and hence $u_i(\{x\}) = 0$ for all $x \in C$. Therefore, F_i is a non-decreasing function on C with $F_i(0) = 0$, $F_i(1) = 1$, and $u_i([x, y]) = F_i(y) - F_i(x)$. An agent i is *hungry* if $u_i(X) > 0$ for all intervals $X \subseteq C$ with positive length; this is equivalent to the condition that F_i is strictly increasing.

Each agent i has an *entitlement* $w_i > 0$ of the cake such that $\sum_{i \in N} w_i = 1$. Let $\mathbf{w}$ denote $(w_1, \ldots, w_n)$. We say that agents have *equal entitlements* if $w_i = 1/n$ for all $i \in N$. For each subset $N' \subseteq N$ of agents, define $w_{N'} = \sum_{i \in N'} w_i$. Note that $w_\varnothing = 0$ and $w_N = 1$. We say that agents have *generic entitlements* if $w_{N_1} \neq w_{N_2}$ for all distinct $N_1, N_2 \subseteq N$.

An *instance* consists of the set of agents N, their valuation functions $(u_i)_{i \in N}$, and their entitlements $\mathbf{w}$.[2]

Given an instance, an allocation $(A_1, \ldots, A_n)$ is *proportional* (resp. *strongly-proportional*) if $u_i(A_i) \geq w_i$ (resp. $u_i(A_i) > w_i$) for all $i \in N$. For agents with equal entitlements, a proportional (resp. strongly-proportional) allocation requires every agent to receive a piece of cake with value at least (resp. greater than) $1/n$.

Algorithms can make eval and mark queries of each agent in the Robertson-Webb model. More specifically, for each agent $i \in N$, value $r \in [0, 1]$, and points $x, y \in C$ with $x \leq y$, $\textsc{Eval}_i(x, y)$ returns $u_i([x, y])$, and $\textsc{Mark}_i(x, r)$ returns the *rightmost* (largest) point $z \in C$

[2] An *instance* as defined in Chapter 2 excludes the entitlements, but we include the entitlements in this chapter.

such that $u_i([x,z]) = r$ (such a point exists due to the continuity of the valuations); if $u_i([x,1]) < r$, then $\textsc{Mark}_i(x,r)$ returns ∞.

For $i \in N$ and $r \in [0,1]$, a point $x \in C$ is an *r-mark* of agent i if $u_i([0,x]) = r$. While the point returned by $\textsc{Mark}_i(0,r)$ is an r-mark of agent i, the converse is not true since $\textsc{Mark}_i(0,r)$ only returns the *rightmost* r-mark of agent i. However, when agent i is *hungry*, then the r-mark is unique, and the two notions coincide. Let $\mathcal{T}$ denote the subset $\{1/n, 2/n, \ldots, (n-1)/n\}$ of C—we shall consider r-marks for $r \in \mathcal{T}$ in Section 6.3.1.

6.3 Hungry Agents

We begin with the simpler case where all agents are hungry. We first state a result which finds a connected strongly-proportional allocation of a cake for hungry agents using a small number of queries when given a connected *proportional* allocation in which one agent has a strongly-proportional piece. The proof proceeds by slightly moving the boundary between two adjacent agents' pieces such that an agent j who received exactly w_j eventually gets a slightly larger piece.

Lemma 6.3.1. *Let an instance with n hungry agents be given. Suppose that we are given a connected proportional allocation $(A_1, \ldots, A_n)$ such that $u_i(A_i) > w_i$ for some $i \in N$. Then, there exists a connected strongly-proportional allocation, and such an allocation can be computed using $O(n)$ queries.*

Proof. First, we find the values of $u_j(A_j)$ for all $j \in N$. If $u_j(A_j) > w_j$ for all $j \in N$, then we are done. Otherwise, there exist two distinct agents $i, j \in N$ with neighboring pieces such that $u_i(A_i) > w_i$ and $u_j(A_j) = w_j$. By slightly moving the boundary between A_i and A_j, we can get a new allocation in which agents i and j each receives a piece worth more than w_i and w_j respectively. To formally describe the process of moving the boundary, we consider two complementary cases.

Case 1: A_i is to the left of A_j. Denote $A_i = [z_1, z_2]$ and $A_j = [z_2, z_3]$. Let $y = \textsc{Mark}_i(z_1, w_i)$; note that $y \in (z_1, z_2)$ since $u_i(A_i) > w_i$. Let y^* be the midpoint of y and z_2. Adjust the two agents' pieces such that agent i now receives $[z_1, y^*]$ and agent j now receives $[y^*, z_3]$; see Figure 6.1 for an illustration.

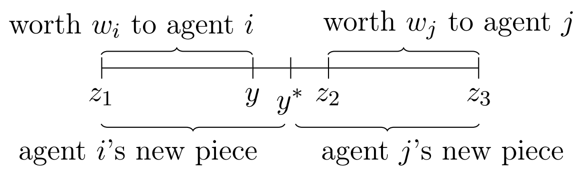


Figure 6.1: Agent i's and j's new pieces in the proof of Lemma 6.3.1.

Since $[z_1, y^*] \supsetneq [z_1, y]$ and the latter is worth w_i to hungry agent i, the new piece, $[z_1, y^*]$, is worth more than w_i to agent i. Likewise, since $[y^*, z_3] \supsetneq [z_2, z_3]$ and the latter is worth w_j to hungry agent j, the new piece, $[y^*, z_3]$, is worth more than w_j to agent j.

Case 2: A_i is to the right of A_j. Denote $A_j = [z_1, z_2]$ and $A_i = [z_2, z_3]$. Let $y = \textsc{Mark}_i(z_2, u_i(A_i) - w_i)$; note that $y \in (z_2, z_3)$ since $u_i(A_i) > w_i$. Let y^* be the midpoint of z_2 and y. Adjust the two agents' pieces such that agent j now receives $[z_1, y^*]$ and agent i now receives $[y^*, z_3]$.

Since $[z_1, y^*] \supsetneq [z_1, z_2]$ and the latter is worth w_j to hungry agent j, the new piece, $[z_1, y^*]$, is worth more than w_j to agent j. Likewise, since $[y^*, z_3] \supsetneq [y, z_3]$ and the latter is worth w_i to hungry agent i (due to additivity, we have $u_i([y, z_3]) = u_i([z_2, z_3]) - u_i([z_2, y]) = w_i$), the new piece, $[y^*, z_3]$, is worth more than w_i to agent i.

In both Case 1 and Case 2, only agent i's and j's pieces change; all of the other agents' pieces do not change. All in all, one additional agent j receives more than w_j of the cake. Proceeding this way at most $n-1$ times yields a connected strongly-proportional allocation.

Finding the values of all $u_j(A_j)$ at the beginning requires n queries, while the adjustment of the boundaries between two agents' pieces requires a constant number of queries, so the total number of queries is in $O(n)$. $\square$

We present the results separately for agents with equal entitlements and agents with possibly unequal entitlements. For n hungry agents with equal entitlements, we state in Section 6.3.1 a simple necessary and sufficient condition for the existence of a connected strongly-proportional allocation. We provide an asymptotically tight bound of $\Theta(n^2)$ for the number of queries needed by an algorithm to determine the existence of such an allocation, as well as to compute one such allocation if it exists. For agents with possibly unequal entitlements, we show in Section 6.3.2 that a lower bound number of queries needed to decide the existence of a connected strongly-proportional allocation is in $\Omega(n \cdot 2^n)$.

6.3.1 Equal Entitlements

Recall that $\mathcal{T} = \{1/n, 2/n, \ldots, (n-1)/n\}$. Our condition uses a particular set of r-marks: those with $r \in \mathcal{T}$.

Theorem 6.3.2. *Let an instance with n hungry agents with equal entitlements be given. Then, a connected strongly-proportional allocation exists if and only if there exist two distinct agents $i, j \in N$ and $r \in \mathcal{T}$ such that the r-mark of agent i is different from the r-mark of agent j.*

Proof. Since the agents are hungry, there is exactly one r-mark of agent i for each $r \in [0, 1]$ and $i \in [n]$.

($\Rightarrow$) We prove the contraposition. Suppose that for each $r \in \mathcal{T}$, every agent has the same r-mark. Every agent also has the same 0-mark of 0 and the same 1-mark of 1. For each $t \in \{0, \ldots, n\}$, denote the common t/n-mark by z_t.

Consider now any connected allocation, which is represented by $n-1$ cuts on the cake. For each $t \in \{1, \ldots, n\}$, denote the t-th cut from the left by x_t; also denote $x_0 = 0$ and $x_n = 1$. Each agent receives a piece $[x_{t-1}, x_t]$ for some $t \in \{1, \ldots, n\}$, and every such piece is allocated to some agent.

Since $x_0 = z_0$ and $x_n = z_n$, there must be some $t \in \{1, \ldots, n\}$ for which $x_{t-1} \geq z_{t-1}$ and $x_t \leq z_t$. This means that the piece $[x_{t-1}, x_t]$ is contained in the interval $[z_{t-1}, z_t]$. Let i

denote the agent who receives the piece $[x_{t-1}, x_t]$. Then, agent i's value for her piece is

$$u_i([x_{t-1}, x_t]) \leq u_i([z_{t-1}, z_t]) = u_i([0, z_t]) - u_i([0, z_{t-1}]) = t/n - (t-1)/n = 1/n,$$

so the allocation is not strongly-proportional. This holds for any connected allocation; therefore, no connected strongly-proportional allocation exists.

($\Leftarrow$) Suppose that there exist two distinct agents $i, j \in N$ and $r \in \mathcal{T}$ such that the r-mark of agent i is different from the r-mark of agent j. We shall construct a connected strongly-proportional allocation by first constructing a connected *proportional* allocation such that at least one agent receives a piece with value more than $1/n$, then use Lemma 6.3.1 to construct a strongly-proportional one.

Let $t \in \{1, \dots, n-1\}$ be the integer such that $r = t/n$. Let i_L be an agent with the leftmost (smallest) r-mark among all the agents, and i_R be an agent with the rightmost (largest) r-mark among all the agents (if there are multiple agents with the same leftmost or rightmost r-mark, we can choose an agent arbitrarily in each case). Denote the leftmost r-mark by z_L and the rightmost r-mark by z_R. Note that $z_L < z_R$, since there are agents with different r-marks.

Since there are n agents, there are n r-marks (possibly some of them are equal) in the interval $[z_L, z_R]$. Let $x \in [z_L, z_R]$ be the t-th r-mark from the left. Then, there exists a partition of the agents into two subsets N_1 and N_2 such that

- $|N_1| = t$, and the r-mark of all agents in N_1 is at most x, and
- $|N_2| = n - t$, and the r-mark of all agents in N_2 is at least x.

Every agent in N_1 values $[0, x]$ at least r, and every agent in N_2 values $[x, 1]$ at least $1 - r$; see Figure 6.2 for an illustration.

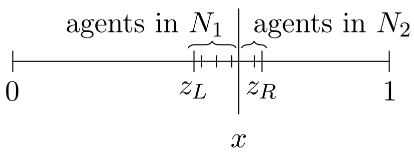


Figure 6.2: The r-marks of all the agents in the proof of Theorem 6.3.2. The point x is at one of the r-marks and divides agents into N_1 and N_2.

Next, we consider any connected proportional cake-cutting algorithm as a black box (e.g., last diminisher). We apply the algorithm on $[0, x]$ and N_1 such that every agent in N_1 receives a connected piece with value at least $1/t$ of her value of $[0, x]$, and apply the algorithm on $[x, 1]$ and N_2 such that every agent in N_2 receives a connected piece with value at least $1/(n-t)$ of her value of $[x, 1]$. We show that this allocation (of $C = [0, 1]$) is proportional. For an agent in N_1, since she values $[0, x]$ at least $r = t/n$, the piece she receives has value at least $(1/t)r = 1/n$. Likewise, for an agent in N_2, since she values $[x, 1]$ at least $1 - r = (n-t)/n$, the piece she receives has value at least $(1/(n-t))(1-r) = 1/n$.

Now, we show that agent i_L or i_R (or both) receives a piece with value strictly more than $1/n$. If $x = z_R$, then we claim that agent i_L receives such a piece. Since the r-mark of agent

i_L is at $z_L < x$, we have $i_L \in N_1$. Since agent i_L is hungry, the piece $[0, x]$ is worth more than r to her, and so the piece she receives has value more than $(1/t)r = 1/n$. Otherwise, $x < z_R$, and a similar argument shows that agent i_R receives such a piece.

Having established a connected proportional allocation in which at least one agent receives more than $1/n$, we apply Lemma 6.3.1 to obtain a connected *strongly-proportional* allocation. □

It is interesting to compare the condition in Theorem 6.3.2 with the one for non-connected allocations. In both cases, a disagreement between *two* agents is sufficient for allocating *all* n agents more than their fair share. However, in the non-connected case, the disagreement can be in an r-mark for any $r \in (0, 1)$ (see the discussion in Section 6.1), whereas in the connected case, the disagreement should be in an r-mark for some $r \in \mathcal{T}$; the r-marks for other values of r are completely irrelevant.

It is clear from Theorem 6.3.2 that we can decide whether a connected strongly-proportional allocation exists for hungry agents with equal entitlements by checking the t/n-marks of all of the n agents for all $t \in \{1, \dots, n-1\}$. This is described in Algorithm 6.1. The number of queries used in the algorithm is at most $n(n-1)$.

Algorithm 6.1 Determining the existence of a connected strongly-proportional allocation for n hungry agents with equal entitlements.

```
for t = 1, ..., n − 1 do
    z_t ← Mark_1(0, t/n)                          ▷ agent 1's t/n-mark
    for i = 2, ..., n do
        if Mark_i(0, t/n) ≠ z_t then return true
    end for
end for
return false
```

Theorem 6.3.3. *Algorithm 6.1 decides whether a connected strongly-proportional allocation exists for n hungry agents with equal entitlements using at most $n(n-1)$ queries.*

Next, we show an asymptotically tight lower bound for the number of queries required to decide the existence of such an allocation for hungry agents. The idea behind the proof is that we must check the t/n-marks of all the agents and all $t \in \{1, \dots, n-1\}$; otherwise, we can craft two instances—one with the t/n-marks coinciding, and the other with some t/n-marks not coinciding—that are consistent with the information obtained by the algorithm and yet give opposite results. Doing this check requires at least $n(n-1)/2$ queries, as each query provides information on at most two points.

Theorem 6.3.4. *Any algorithm that decides whether a connected strongly-proportional allocation exists for n hungry agents with equal entitlements requires at least $n(n-1)/2$ queries.*

Proof. Suppose by way of contradiction that some algorithm decides the existence of a connected strongly-proportional allocation for n hungry agents with equal entitlements using fewer than $n(n-1)/2$ queries. We assume that for all $i \in N$, $r \in [0, 1]$ and $x \in C$,

$\textsc{Eval}_i(0,x)$ returns the value x and $\textsc{Mark}_i(0,r)$ returns the point r. We make the following adjustments to the algorithm: whenever the algorithm makes an $\textsc{Eval}_i(x,y)$ query, it is instead given the answers to $\textsc{Mark}_i(0,x)=x$ and $\textsc{Mark}_i(0,y)=y$, and whenever the algorithm makes a $\textsc{Mark}_i(x,r)$ query, it is instead given the answers to $\textsc{Mark}_i(0,x)=x$ and $\textsc{Mark}_i(0,x+r)=x+r$.[3] This means that every query made by the algorithm provides the algorithm only with information on at most *two* r-marks of some agent and no other information that cannot be deduced from these r-marks. Note that the algorithm can still deduce the values of $\textsc{Eval}_i(x,y)$ and $\textsc{Mark}_i(x,r)$ by taking the difference between the two answers given, which means that the information provided to the algorithm after the adjustment is a superset of the information provided to the algorithm before the adjustment.

The answers given to the algorithm are consistent with the instance where every agent's valuation is uniformly distributed over the cake—in which case there is no connected strongly-proportional allocation of the cake by Theorem 6.3.2—and so the algorithm should output "false". However, we shall now show that the information provided to the algorithm is also consistent with an instance with a connected strongly-proportional allocation. This means that the algorithm is not able to differentiate between the two, resulting in a contradiction.

Since fewer than $n(n-1)/2$ queries were made by the algorithm, fewer than $n(n-1)$ r-marks (for $r \in (0,1)$) of all the agents are known. In particular, there exists an agent $i \in N$ such that fewer than $n-1$ r-marks of agent i are known, and hence there exists $t \in \{1,\ldots,n-1\}$ such that the t/n-mark of agent i is not known. We now modify agent i's valuation function slightly from the uniform distribution. Let $\epsilon \in (0,1/n)$ be a number such that every known r-mark of agent i is of distance more than ϵ from t/n. Let the t/n-mark of agent i to be at $t/n+\epsilon$. Construct agent i's valuation function such that its distribution between all known r-marks of agent i (including the new t/n-mark) is uniform within the respective intervals—note that this construction is valid and unique since these known r-marks are strictly increasing in r. Let the other agents' valuation functions be uniformly distributed on the whole cake. Then, agent i's t/n-mark is different from every other agents' t/n-mark. By Theorem 6.3.2, this instance admits a connected strongly-proportional allocation of the cake, forming the desired contradiction. □

Theorems 6.3.3 and 6.3.4 show that the number of queries required to determine the existence of a connected strongly-proportional allocation for n hungry agents with equal entitlements is in $\Theta(n^2)$. The same can be said for *computing* such an allocation—we can modify Algorithm 6.1 using the details in the proof of Theorem 6.3.2 to output a connected strongly-proportional allocation of the cake instead, if such an allocation exists.

Theorem 6.3.5. *The number of queries required to decide the existence of a connected strongly-proportional allocation for n hungry agents with equal entitlements, or to compute such an allocation if it exists, is in $\Theta(n^2)$.*

[3] Assuming $x+r \leq 1$; otherwise, $\textsc{Mark}_i(0,x+r)=\infty$.

6.3.2 Possibly Unequal Entitlements

We now consider hungry agents who may not necessarily have equal entitlements. Since the entitlement of a subset of agents may not be a multiple of $1/n$, we cannot use the condition in Theorem 6.3.2 which uses r-marks for $r \in \mathcal{T}$. This requires us to devise a more general condition to determine the existence of a connected strongly-proportional allocation, which can be checked using $O(n \cdot 2^n)$ queries. Since the condition also works for non-hungry agents, we defer the discussion to Section 6.4.1 (see Theorems 6.4.4 and 6.4.5).

We now show an asymptotically-tight *lower bound* for the case when agents may have unequal entitlements. We show an even stronger result: for every vector of *generic entitlements*, the number of queries required to decide the existence of a connected strongly-proportional allocation is in $\Omega(n \cdot 2^n)$. The proof uses an adversarial argument similar to the one in Theorem 6.3.4.

Theorem 6.3.6. *Let* $\mathbf{w}$ *be any vector of generic entitlements. Then, any algorithm that decides whether a connected strongly-proportional allocation exists for n hungry agents with entitlements* $\mathbf{w}$ *requires at least* $n \cdot 2^{n-2}$ *queries.*

Proof. Since the entitlements are generic, we can arrange the 2^n different subsets of agents in strictly increasing order of their entitlements, i.e., we label the subsets of N as $N_1, \ldots, N_{2^n}$ such that $w_{N_1} < \cdots < w_{N_{2^n}}$. Note that $N_1 = \varnothing$ and $N_{2^n} = N$, giving $w_{N_1} = 0$ and $w_{N_{2^n}} = 1$.

Let $d = \min_{k=1}^{2^n-1}(w_{N_{k+1}} - w_{N_k})$ be the smallest gap between entitlements of different agent subsets. For each $k \in \{2, \ldots, 2^n - 1\}$, define $I_k = [w_{N_k},\ w_{N_k} + d/2]$. Note that, by the choice of d, all the I_k are pairwise disjoint.

Suppose by way of contradiction that some algorithm decides the existence of a connected strongly-proportional allocation for n hungry agents with generic entitlements using fewer than $n \cdot 2^{n-2}$ queries. We follow the construction in the proof of Theorem 6.3.4 where we modify the algorithm such that every query returns information on at most *two* r-marks of some agent, and these information are consistent with the instance where every agent's valuation is uniformly distributed over the cake. Therefore, the algorithm should output "false". We shall now show that the information provided to the algorithm is also consistent with an instance with a connected strongly-proportional allocation. This means that the algorithm is not able to differentiate between the two, resulting in a contradiction.

Since fewer than $n \cdot 2^{n-2}$ queries were made by the algorithm, there exists an agent $i \in N$ such that at most $2^{n-2} - 1$ queries about the r-marks of agent i (for $r \in (0, 1)$) are made. Since each query returns information on at most *two* r-marks, at most $2^{n-1} - 2$ r-marks of agent i are known. There are $2^{n-1} - 1$ non-empty subsets N_k of N that do not contain agent i, so there exists $k \in \{2, \ldots, 2^n - 1\}$ such that $i \notin N_k$ and no known r-mark of agent i is in the interval I_k. Let $w = w_{N_k}$. Let the w-mark of agent i be at $w + d/4$. Construct agent i's valuation function such that its distribution between all known r-marks of agent i (including the new w-mark) is uniform within the respective intervals—note that this construction is valid and unique since these known r-marks are strictly increasing in r. Let the other agents' valuation functions be uniformly distributed on the whole cake.

We show that a connected strongly-proportional allocation exists. The leftmost pieces are allocated to agents in N_k in any arbitrary order, where every agent $j \in N_k$ receives a piece of length w_j. Agent i receives the piece $[w, w + w_i]$. Finally, the remaining cake is allocated to the remaining agents such that every agent j receives a piece of length w_j. Note that every agent $j \in N \setminus \{i\}$ receives a piece worth exactly w_j, since their valuation functions are uniform. The value of $[w + d/4, w + w_i]$ is w_i to agent i, so agent i's piece $[w, w + w_i] \supsetneq [w + d/4, w + w_i]$ is worth more than w_i to hungry agent i. Therefore, the allocation is proportional (and clearly connected) with agent i receiving a piece strictly greater than w_i. By Lemma 6.3.1, a connected strongly-proportional allocation of the cake exists, forming the desired contradiction. □

Using the results from Theorem 6.3.6 and from Theorem 6.4.5 later, we get a tight bound for hungry agents with possibly unequal entitlements.

Theorem 6.3.7. *The number of queries required to decide the existence of a connected strongly-proportional allocation for n hungry agents, or to compute such an allocation if it exists, is in* $\Theta(n \cdot 2^n)$.

The lower bound in Theorem 6.3.6 is derived from the number of different values of w_{N_k}. In particular, a lower bound number of queries is

$$\frac{1}{2} \sum_{i=1}^{n} |\{w_{N'} : \varnothing \neq N' \subseteq N, i \notin N'\}|. \tag{6.1}$$

For *generic* entitlements, each term in the sum equals $2^{n-1} - 1$, so we get roughly the lower bound of $n \cdot 2^{n-2}$ in Theorem 6.3.6. In contrast, for *equal* entitlements, each term in the sum equals $n - 1$, so we get the lower bound of $n(n-1)/2$ in Theorem 6.3.4.

For entitlements that are neither generic nor equal, the resulting lower bound is between these two extremes. It is an interesting open question to find an algorithm with a query complexity matching the lower bound in (6.1) in these intermediate cases. The main difficulty in extending our algorithm for equal entitlements (Algorithm 6.1) to unequal entitlements is due to the step in Theorem 6.3.2 where we used a black-box algorithm for *proportional* cake-cutting (such as last diminisher) to divide a part of the cake among the agents in N_1 and the other part among the agents in N_2. Such a black box algorithm does not exist for unequal entitlements, since a connected proportional allocation might not even exist for unequal entitlements in the first place.

6.4 General Agents

We now consider the general case where agents need not be hungry. Recall that the condition we developed in Theorem 6.3.2 involves checking for the coincidence of r-marks of all the agents for $r \in \mathcal{T}$. However, there are some difficulties in generalizing the condition for non-hungry agents, even for equal entitlements. The proof of Theorem 6.3.2 relies crucially on the fact that an r-mark of an agent is unique, which may not be true for non-hungry agents. For each agent $i \in N$, $F_i(x) = u_i([0, x])$ is a continuous function with domain $C = [0, 1]$ and range $[0, 1]$. For each $r \in [0, 1]$, the set of r-marks of agent i is $F_i^{-1}(\{r\})$. Since $\{r\}$ is a closed set and F_i is continuous, $F_i^{-1}(\{r\})$ is a non-empty closed set. If agent i is not necessarily hungry, then the fact that F_i is non-decreasing implies the set of r-marks of agent i is thus a non-empty closed *interval* (though possibly the singleton set $[x, x] = \{x\}$).

Another difficulty is that there may be different instances with the same t/n-marks but give different results regarding the existence of such an allocation. We show this via the following two examples.

Example 6.1. Consider a cake-cutting instance for $n = 3$ agents with equal entitlements where the cake is made up of 11 homogenous regions. The following table shows the agents' valuations for each region.

Alice	9	0	0	0	9	0	0	0	0	0	9
Bob	1	4	4	3	1	5	1	1	2	4	1
Chana	1	8	2	2	1	1	1	2	4	4	1

All agents value[4] the entire cake at 27, so the t/n-marks are at values 9 and 18. Alice has two intervals of t/n-marks—the two intervals of zeros. Bob and Chana each has two t/n-marks that are single points, denoted by vertical lines—note that both Bob and Chana are hungry. We show that no connected strongly-proportional allocation exists.

Suppose by way of contradiction that a connected strongly-proportional allocation exists. Alice must receive a piece with value larger than 9, so her piece must touch the middle 9 as well as either the left 9 or the right 9. In the former case, the cake remaining for Bob and Chana is at most:

Bob	1	5	1	1	2	4	1
Chana	1	1	1	2	4	4	1

In the latter case, the remaining cake is at most:

Bob	1	4	4	3	1
Chana	1	8	2	2	1

In both cases, no matter how the remaining cake is divided between Bob and Chana, at least one agent gets a piece of cake with value at most 9, so no connected strongly-proportional allocation exists.

Example 6.2. Consider the following instance modified from Example 6.1.

Alice	9	0	0	0	9	0	0	0	0	0	9
Bob	1	4	4	3	1	5	5	1	1	1	1
Chana	1	8	2	2	1	1	1	2	4	4	1

The t/n-marks of the agents are identical to those in Example 6.1. However, a connected strongly-proportional allocation exists, as the following table shows:

[4]The value of the cake should technically be normalized to 1, but this can be done by simply dividing every value by 27. We use integers here and in all subsequent examples for simplicity.

Alice	9 0 0 0 9		
Bob		5 5	
Chana			2 4 4 1

Examples 6.1 and 6.2 show that the condition for determining the existence of a connected strongly-proportional allocation cannot be extended trivially from the result for hungry agents. Instead, let us discuss the extent to which the results from Section 6.3.1 can be extended. We start with a necessary condition regarding the r-marks for $r \in \mathcal{T}$. This condition is similar to that in Theorem 6.3.2.

Proposition 6.4.1 (Necessary condition)**.** *Let an instance with n agents with equal entitlements be given. If there exists a connected strongly-proportional allocation, then there exist two distinct agents $i, j \in N$ and $r \in \mathcal{T}$ such that the interval of r-marks of agent i is disjoint*[5] *from the interval of r-marks of agent j.*

Proof. Let a connected strongly-proportional allocation be given, and let $\sigma : N \to N$ be the permutation such that agent $\sigma(k)$ receives the k-th piece from the left. Suppose by way of contradiction that for each $r \in \mathcal{T}$, the intervals of r-marks of every pair of agents have non-empty intersection. We show by backward induction that for each $k \in \{1, \dots, n\}$, every agent in $\{\sigma(1), \dots, \sigma(k)\}$ assigns a total value of at most k/n to the leftmost k pieces.

The base case of $k = n$ is clear—every agent assigns a value of at most $n/n = 1$ to the whole cake. Suppose that the statement is true for $k+1$ for some $k \in \{1, \dots, n-1\}$; we shall prove the statement for k. Since agent $\sigma(k+1)$ assigns a value of at most $(k+1)/n$ to the leftmost $k+1$ pieces, the left endpoint of her piece must be strictly to the left of her interval of k/n-marks in order for her to receive a piece worth more than $1/n$. Now, consider agent $\sigma(i)$ for $i \in \{1, \dots, k\}$. Since agent $\sigma(i)$'s interval of k/n-marks intersects with agent $\sigma(k+1)$'s interval of k/n-marks, the remaining cake after removing $\sigma(k+1)$'s piece is worth at most k/n to agent $\sigma(i)$. This proves the inductive statement.

Now, the statement for $k = 1$ states that agent $\sigma(1)$ receives a piece worth at most $1/n$. This contradicts the assumption that the allocation is strongly-proportional. □

Next, we provide a sufficient condition for a connected strongly-proportional allocation using intervals of r-marks for $r \in \mathcal{T}$. It differs from the necessary condition of Proposition 6.4.1 in that it requires the intervals of r-marks of *all* agents, rather than just two, to be pairwise disjoint.

Proposition 6.4.2 (Sufficient condition)**.** *Let an instance with n agents with equal entitlements be given. If there exists $r \in \mathcal{T}$ such that the intervals of r-marks of all agents are pairwise disjoint, then a connected strongly-proportional allocation exists.*

Proof. Let $r = t/n$ be such that the intervals of r-marks of all the agents are pairwise disjoint. Since these intervals of r-marks are closed and are pairwise disjoint, we can arrange them from smallest to largest. Moreover, the gap between any two consecutive intervals of r-marks

[5] Unlike for pieces of cake where "disjoint" means *finite* intersection, we revert to the standard definition of "disjoint" to mean *empty* intersection for intervals involving r-marks.

is a non-empty open interval. Let N_1 be the set of t agents whose intervals of r-marks are the smallest, and let N_2 be the remaining agents. Then, there exists a point x between the r-marks of the agents in N_1 and that of the agents in N_2. Note that the t agents in N_1 each values the cake $[0, x]$ more than t/n, and the $n - t$ agents in N_2 each values the cake $[x, 1]$ more than $(n - t)/n$. We apply any connected proportional cake-cutting algorithm on each of $[0, x]$ on N_1 and $[x, 1]$ on N_2 such that every agent receives a connected piece worth more than $1/n$. This gives a connected strongly-proportional allocation. □

Propositions 6.4.1 and 6.4.2 coincide for $n = 2$ agents, yielding the following result.

Corollary 6.4.3. *Let an instance with two agents with equal entitlements be given. Then, a connected strongly-proportional allocation exists if and only if the intervals of* $1/2$*-marks of the two agents are disjoint.*

The two conditions do not coincide for $n \geq 3$ agents, however. In the search for a necessary and sufficient condition for three or more agents, one could consider weakening the condition in Proposition 6.4.2 to require the intervals of r-marks of just *two* agents to be pairwise disjoint for some $r \in \mathcal{T}$. However, as one could see in Example 6.1, even when the intervals of r-marks of Bob and Chana are disjoint for *all* $r \in \mathcal{T}$, there is no connected strongly-proportional allocation. Another possibility is to require that the interval of r-marks of one agent to be disjoint from every other agents' intervals of r-marks for some $r \in \mathcal{T}$. However, the following example shows that this is still not correct.

Example 6.3. Consider the following instance for $n = 3$ agents.

Alice	4	2	2	1	3
Bob	4	0	2	2	4
Chana	4	0	2	2	4

All agents value the entire cake at 12, so the 2/3-marks are at value 8, denoted by vertical lines. Alice's 2/3-mark is disjoint from Bob's and Chana's 2/3-marks. However, no connected strongly-proportional allocation exists. If Alice receives the leftmost piece or the rightmost piece, then the remaining cake is worth at most 8 to both Bob and Chana, and both of them cannot simultaneously get a piece worth more than 4 each since they have identical valuations. If Alice receives the middle piece instead, then Bob and Chana must receive the leftmost and the rightmost piece in some order. However, the leftmost piece must touch the third region, the rightmost piece must touch the fourth region, and Alice's piece is confined to the third and fourth regions which is only worth at most 3 to her.

Examples 6.1 to 6.3 show that the existence of a connected strongly-proportional allocation cannot be determined based on t/n-marks alone. This inspires us to find another condition that characterizes the existence of a connected strongly-proportional allocation.

In Section 6.4.1, we generalize the condition from Theorem 6.3.2 for n non-hungry agents, regardless of whether they have equal entitlements or not. We show that this condition can be checked by an algorithm using $O(n \cdot 2^n)$ queries. Now, the result in Theorem 6.3.6 says that the lower bound number of queries needed for an algorithm to determine the existence of a connected strongly-proportional allocation for n hungry agents with generic entitlements is $\Omega(n \cdot 2^n)$—we show in Section 6.4.2 that this lower bound also applies to (not necessarily hungry) agents with *equal entitlements*.

6.4.1 Upper Bound

Our condition requires agents to mark pieces of cake one after another in a certain order. We explain this operation more precisely. Let $\sigma : N \to N$ be a permutation of agents, and let $x \in C$ and $r_1, \ldots, r_n \in [0,1]$. The agents proceed in the order $\sigma(1), \ldots, \sigma(n)$. Agent $\sigma(1)$ starts first and makes a mark at $x_1 = \textsc{Mark}_{\sigma(1)}(x, r_{\sigma(1)})$, the rightmost point such that $[x, x_1]$ is worth $r_{\sigma(1)}$ to her. Then, agent $\sigma(2)$ continues from x_1, and makes a mark at $x_2 = \textsc{Mark}_{\sigma(2)}(x_1, r_{\sigma(2)})$, the rightmost point such that $[x_1, x_2]$ is worth $r_{\sigma(2)}$ to her. Each agent $\sigma(i)$ repeats the same process of making a mark at $x_i = \textsc{Mark}_{\sigma(i)}(x_{i-1}, r_{\sigma(i)})$ such that $[x_{i-1}, x_i]$ is the largest possible piece worth $r_{\sigma(i)}$ to her. We shall overload the definition of $\textsc{Mark}$ and define[6] $\textsc{Mark}_\sigma(x, \mathbf{r})$ as the point x_n resulting from this sequential marking process, where $\mathbf{r} = (r_1, \ldots, r_n)$. If $[x_{i-1}, 1]$ is worth less than $r_{\sigma(i)}$ to agent $\sigma(i)$ at any point, then $\textsc{Mark}_\sigma(x, \mathbf{r})$ is defined as ∞. This operation is described in Algorithm 6.2. Note that each $\textsc{Mark}_\sigma(x, \mathbf{r})$ operation requires at most n ($\textsc{Mark}_i$) queries.

Algorithm 6.2 Computing $\textsc{Mark}_\sigma(x, \mathbf{r})$ for n agents.

$x_0 \leftarrow x$
for $i = 1, \ldots, n$ **do**
  $x_i \leftarrow \textsc{Mark}_{\sigma(i)}(x_{i-1}, r_{\sigma(i)})$
  if $x_i = \infty$ **then return** ∞
end for
return x_n

Our necessary and sufficient condition for n (possibly non-hungry) agents requires us to check whether the point $\textsc{Mark}_\sigma(0, \mathbf{w})$ is less than 1 for some permutation σ. The point $\textsc{Mark}_\sigma(0, \mathbf{w})$ is determined when agents go in the order as prescribed by σ and make their rightmost marks worth their entitlements to each of them one after another. The idea behind the proof is that starting from the agent who receives the rightmost piece in σ and going leftwards, each agent is able to move the boundaries of her piece such that she receives a small piece of cake with positive value ϵ from the right and gives away a small piece of cake with value $\epsilon/2$ to the agent on the left, thereby increasing the value of her piece by a positive value $\epsilon/2$.

Theorem 6.4.4. *Let an instance with n agents be given. Then, a connected strongly-proportional allocation exists if and only if there exists a permutation $\sigma : N \to N$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$.*

Proof. ($\Rightarrow$) Suppose that a connected strongly-proportional allocation exists. Let $\sigma : N \to N$ be the permutation such that agent $\sigma(k)$ receives the k-th piece from the left in this allocation, and let $y_0, y_1, \ldots, y_n$ be the points such that agent $\sigma(k)$ receives the piece $[y_{k-1}, y_k]$ with

[6] The subscript of $\textsc{Mark}$ here is a permutation σ, not an agent number.

$y_0 = 0$ and $y_n = 1$. We shall show that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$. Let $x_0, x_1, \ldots, x_n$ be the points as described by Algorithm 6.2 for $\textsc{Mark}_\sigma(0, \mathbf{w})$. We shall show by induction that $x_k < y_k$ for all $k \in \{1, \ldots, n\}$; then, $\textsc{Mark}_\sigma(0, \mathbf{w}) = x_n < y_n = 1$ gives the desired conclusion.

For the base case $k = 1$, we have $x_1 = \textsc{Mark}_{\sigma(1)}(0, w_{\sigma(1)})$, so x_1 is a point for which $u_{\sigma(1)}([0, x_1]) = w_{\sigma(1)}$. Since agent $\sigma(1)$ receives a piece $[y_0, y_1] = [0, y_1]$ worth more than $w_{\sigma(1)}$, we must have $x_1 < y_1$. For the inductive case, assume that $x_k < y_k$ for some $k \in \{1, \ldots, n-1\}$, and consider $k + 1$. We have $x_{k+1} = \textsc{Mark}_{\sigma(k+1)}(x_k, w_{\sigma(k+1)}) \leq \textsc{Mark}_{\sigma(k+1)}(y_k, w_{\sigma(k+1)})$ since $x_k < y_k$. Since agent $\sigma(k + 1)$ receives a piece $[y_k, y_{k+1}]$ worth more than $w_{\sigma(k+1)}$, we have $\textsc{Mark}_{\sigma(k+1)}(y_k, w_{\sigma(k+1)}) < y_{k+1}$. Therefore, the result $x_{k+1} < y_{k+1}$ holds, proving the induction statement.

($\Leftarrow$) Suppose that there exists a permutation $\sigma : N \to N$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$. Let $x_0, x_1, \ldots, x_n$ be the points as described by Algorithm 6.2 for $\textsc{Mark}_\sigma(0, \mathbf{w})$. Since x_k, which is $\textsc{Mark}_{\sigma(k)}(x_{k-1}, w_{\sigma(k)})$, is the rightmost point z such that $[x_{k-1}, z]$ is worth $w_{\sigma(k)}$ to agent $\sigma(k)$, the piece $[x_{k-1}, y_k]$ is worth more than $w_{\sigma(k)}$ to agent $\sigma(k)$ whenever $y_k > x_k$.

We shall define the points $y_1, \ldots, y_n \in C$ in the reverse order such that $y_k > x_k$ for all $k \in \{1, \ldots, n\}$. Define $y_n = 1 > \textsc{Mark}_\sigma(0, \mathbf{w}) = x_n$. Next, for each $k \in \{1, \ldots, n-1\}$, assume that y_{k+1} is defined such that $y_{k+1} > x_{k+1}$. Since $[x_k, y_{k+1}]$ is worth more than $w_{\sigma(k+1)}$ to agent $\sigma(k + 1)$, it must be worth $w_{\sigma(k+1)} + \epsilon_{k+1}$ to agent $\sigma(k + 1)$ for some $\epsilon_{k+1} > 0$. Define $y_k = \textsc{Mark}_{\sigma(k+1)}(x_k, \epsilon_{k+1}/2)$. Then, we have $y_k > x_k$. This completes the definition of $y_1, \ldots, y_n$.

Let $y_0 = x_0 = 0$. We shall show that the allocation with the cut points at $y_0, \ldots, y_n$ such that $[y_{k-1}, y_k]$ is allocated to agent $\sigma(k)$ for $k \in N$ is strongly-proportional. Agent $\sigma(1)$ receives $[y_0, y_1] = [x_0, y_1]$ which is worth more than $w_{\sigma(1)}$ to her. For $k \in \{2, \ldots, n\}$, since $[x_{k-1}, y_k]$ is worth $w_{\sigma(k)} + \epsilon_k$ and $[x_{k-1}, y_{k-1}]$ is worth $\epsilon_k/2$ to agent $\sigma(k)$, the piece $[y_{k-1}, y_k]$ is worth $(w_{\sigma(k)} + \epsilon_k) - \epsilon_k/2 > w_{\sigma(k)}$ to agent $\sigma(k)$. This completes the proof. □

The condition in Theorem 6.4.4 reduces to the condition in Theorem 6.3.2 for hungry agents with equal entitlements, i.e., when $\mathbf{w} = (1/n, \ldots, 1/n)$. In particular, when every agent has the same r-mark for each $r \in \mathcal{T}$, then each of the n marks made in the $\textsc{Mark}_\sigma(0, \mathbf{w})$ operation coincides at some $x_i \in \mathcal{T} \cup \{1\}$ for every permutation, and so $\textsc{Mark}_\sigma(0, \mathbf{w}) = 1$ for all σ. This corresponds to the case where no connected strongly-proportional allocation exists.

The analysis in Theorem 6.4.4 relies crucially on the fact that the $\textsc{Mark}_i$ queries return the *rightmost* points. If the *leftmost* points are returned instead, then the condition does not work—this can be seen from Example 6.1 where Chana, Alice, and Bob could (left-)mark their respective $1/n$ piece of the cake one after another in this order and still have a positive-valued cake left, but no connected strongly-proportional allocation exists as we demonstrated in Example 6.1.

Alice		0 0 9		0 9
Bob			5 1 1 2	4 1
Chana	1 8			4 1

We can determine whether the condition in Theorem 6.4.4 holds by checking all permutation σ to see whether the point $\textsc{Mark}_\sigma(0, \mathbf{w})$ is less than 1 for some σ. Since there are $n!$ possible permutations of N and each $\textsc{Mark}_\sigma$ operation requires at most n queries, the total number of queries required in the algorithm is at most $n \cdot n!$.

However, we can reduce the number of queries to $n \cdot 2^{n-1}$ by dynamic programming. Our approach is similar to the method used by Aumann et al. (2012)—in their work, they iteratively find a value w such that there exists a connected allocation where every agent receives *at least* w, while here we require every agent i to receive a connected piece with value *strictly more* than w_i.

We now describe our algorithm. For every subset $N' \subseteq N$, our algorithm caches the *best* mark $b_{N'}$ obtained by the subset of agents. The best mark $b_{N'}$ is the leftmost point possible over all permutations of the agents in N' when the agents go in the order as prescribed by the permutation and make their rightmost marks worth their entitlements to each of them one after another. The algorithm aims to compute this point for every N'.

The best mark for the empty set of agents is initialized as $b_\varnothing = 0$. Thereafter, for every $k \in \{1, \dots, n\}$, we assume that the best mark for every subset of $k-1$ agents is calculated earlier and cached. We now need to find $b_{N'}$ for every subset $N' \subseteq N$ with k agents. The last agent to make the best mark for N' could be any of the agents $i \in N'$. Therefore, for each $i \in N'$, we retrieve the best mark for $N' \setminus \{i\}$, which is $b_{N' \setminus \{i\}}$ and has been cached earlier, and let agent i make the rightmost mark such that the cake starting from $b_{N' \setminus \{i\}}$ is worth w_i to her. By iterating through all $i \in N'$, we find the leftmost such point and cache this point as $b_{N'}$. When $k = n$, we obtain b_N, which is the best $\textsc{Mark}_\sigma(0, \mathbf{w})$ over all permutations σ. Therefore, the algorithm returns "true" if $b_N < 1$, and "false" otherwise. This implementation reduces the number of queries by a factor of $2^{\omega(n)}$.

This algorithm is described in Algorithm 6.3. The correctness of the algorithm relies on the statement in Theorem 6.4.4 and the fact that b_N in the algorithm is less than 1 if and only if there exists a permutation $\sigma : N \to N$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$.

Algorithm 6.3 Determining the existence of a connected strongly-proportional allocation for n agents with fewer queries.

$b_\varnothing \leftarrow 0$
for $k = 1, \dots, n$ **do**
  for each subset $N' \subseteq N$ with $|N'| = k$ **do**
    $b_{N'} \leftarrow \infty$
    for each agent $i \in N'$ **do**
      $y \leftarrow \textsc{Mark}_i(b_{N' \setminus \{i\}}, w_i)$
      if $y < b_{N'}$ **then** $b_{N'} \leftarrow y$ ▷ this finds the "best" $b_{N'}$
    end for
  end for
end for
if $b_N < 1$ **then return** true **else return** false

Theorem 6.4.5. *Algorithm 6.3 decides whether a connected strongly-proportional allocation exists for n agents using at most $n \cdot 2^{n-1}$ queries.*

Proof. To show that Algorithm 6.3 is correct, it suffices to show that b_N in the algorithm is less than 1 if and only if there exists a permutation $\sigma : N \to N$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$, by Theorem 6.4.4.

($\Rightarrow$) If b_N in the algorithm is less than 1, then b_N is contributed by some agent $i_n \in N$ making the rightmost w_{i_n}-mark after $b_{N\setminus\{i_n\}}$. Let $\sigma(n) = i_n$. We then consider the agent i_{n-1} contributing the rightmost $w_{i_{n-1}}$-mark for $b_{N\setminus\{i_n\}}$, and so on. Repeat the procedure $n-1$ times to obtain the identities of the agents $\sigma(n-1), \ldots, \sigma(1)$. Then, σ is the desired permutation.

($\Leftarrow$) Suppose there exists a permutation $\sigma : N \to N$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$. For each $k \in \{1, \ldots, n\}$, let $N_k = \{\sigma(1), \ldots, \sigma(k)\}$, and let x_k^σ be the mark where agents $\sigma(1), \ldots, \sigma(k)$ make their rightmost mark worth their entitlements to each of them one after another in this order. We prove by induction on k that $b_{N_k} \le x_k^\sigma$. The base case of $k = 1$ is clear, as the two quantities are equal. Assume that the inequality is true for $k \in \{1, \ldots, n-1\}$; we shall prove the result for $k+1$. The point $b_{N_{k+1}}$ is the smallest point over all permutations where agents $\sigma(1), \ldots, \sigma(k+1)$ make their rightmost $1/n$-mark one after another in some order. In particular, x_{k+1}^σ is one of these points under consideration. Therefore, we must have $b_{N_{k+1}} \le x_{k+1}^\sigma$, proving the induction statement. Then, we have $b_N = b_{N_n} \le x_n^\sigma = \textsc{Mark}_\sigma(0, \mathbf{w}) < 1$.

Next, we show that the number of queries made by Algorithm 6.3 is at most $n \cdot 2^{n-1}$. Let $k \in \{1, \ldots, n\}$ be given. There are $\binom{n}{k}$ subsets N' with cardinality k, and for each N', each of the $|N'| = k$ agents makes a mark query. This means that $k\binom{n}{k}$ queries are made. Hence, the total number of queries is $\sum_{k=1}^{n} k\binom{n}{k} = n \cdot 2^{n-1}$ by a combinatorial identity. $\square$

6.4.2 Lower Bound

Theorem 6.3.6 provides a lower bound for hungry agents with unequal entitlements; we shall now prove a similar lower bound for general agents with equal entitlements.

At a high level, the technique used is similar to that in the proofs of Theorems 6.3.4 and 6.3.6: we use an adversarial argument where we construct an instance with agents having uniform valuations on the cake such that no strongly-proportional allocation exists, but tweak the valuations slightly depending on the queries made. However, the details from the proof of Theorem 6.3.4 cannot be used directly since the existence of a connected strongly-proportional allocation is not solely dependent on the r-marks for $r \in \mathcal{T}$ for non-hungry agents (see the discussion at the beginning of Section 6.4), and the details from the proof of Theorem 6.3.6 cannot be used directly since Theorem 6.3.6 requires the entitlements to be generic.

Instead, we construct the following instance with $n \ge 3$ agents. The cake is divided into $2n-1$ parts. The odd parts (i.e., the 1st, 3rd, …, $(2n-1)$-th parts) are non-valuable to agents 1 to $n-1$, and worth $1/n$ each to agent n. The even parts (i.e., the 2nd, 4th, …, $(2n-2)$-th parts) are valuable to agents 1 to $n-1$, and non-valuable to agent n. For $i \in [n-1]$, agent i's first $n-2$ valuable parts (i.e., the 2nd, 4th, …, $(2n-4)$-th parts) are worth $a_i/(n-2)$ each to agent i for some carefully selected a_i, and the last valuable part (i.e., the $(2n-2)$-th part) is worth $1 - a_i$ to agent i. See Figure 6.3 for an illustration.

Consider a connected strongly-proportional allocation with equal entitlements. Agent n's piece has to include pieces from at least two consecutive odd parts in order for her value to

Agent 1	0	$a_1/(n-2)$	$\cdots$	0	$a_1/(n-2)$	0	$1-a_1$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Agent $n-1$	0	$a_{n-1}/(n-2)$	(total: $n-2$ identical copies)	0	$a_{n-1}/(n-2)$	0	$1-a_{n-1}$	0
Agent n	$1/n$	0		$1/n$	0	$1/n$	0	$1/n$

Figure 6.3: Construction of the cake used in the proof of Theorem 6.4.6.

be greater than $1/n$. By a clever choice of a_i for $i \in \{1, \ldots, n-1\}$, we force these two odd parts to be the *rightmost* odd parts. This leaves the remaining $2n-4$ parts for agents 1 to $n-1$. Removing all the non-valuable parts for these agents, the remaining valuable parts of the cake are worth a_i to agent $i \in \{1, \ldots, n-1\}$. Divide all valuations and entitlements by a_i for each $i \in \{1, \ldots, n-1\}$. Then, this is equivalent to a cake with value 1 to every agent such that each agent's entitlement is $w'_i = 1/na_i$. If we select the a_i's carefully such that $\sum_{i=1}^{n-1} w'_i = 1$ and the entitlements w'_i's are *generic*, then we can invoke Theorem 6.3.6 to show that the lower bound number of queries is in $\Omega(n \cdot 2^n)$.

Theorem 6.4.6. *Any algorithm that decides whether a connected strongly-proportional allocation exists for n agents with equal entitlements requires $\Omega(n \cdot 2^n)$ queries.*

Proof. Let M be a sufficiently large constant (particularly, $M \geq 2^n n^2$), and for each $i \in \{1, \ldots, n-1\}$, define $w'_i = \frac{M+2^{i-1}}{(n-1)M+2^{n-1}-1}$ and $a_i = \frac{1}{nw'_i}$. Note that $\sum_{i=1}^{n-1} w'_i = 1$.

Consider a cake with $2n-1$ parts as illustrated in Figure 6.3. We now show that the valuations on the cake are valid. It suffices to show that for each $i \in \{1, \ldots, n-1\}$, each of $a_i/(n-2)$ and $1-a_i$ is positive. It is clear that w'_i is positive, which means that a_i and hence $a_i/(n-2)$ are positive. We have $1-a_i = \frac{M-2^{n-1}+2^{i-1}n+1}{n(M+2^{i-1})}$ which is positive since $M > 2^{n-1}$. These show that the valuations are valid.

We show that any algorithm that makes fewer than $(n-1)(2^{n-3}-n+3) \in \Omega(n \cdot 2^n)$ queries may not be able to decide whether a connected strongly-proportional allocation exists. The proof idea is similar to that in the proofs of Theorems 6.3.4 and 6.3.6. We assume that the answer to every query made by the algorithm is consistent with the instance where the valuation of each agent is uniformly distributed in their valuable parts, which are the even parts for agents $i \in \{1, \ldots, n-1\}$ and the odd parts for agent n. We show that regardless of what the algorithm outputs as its answer, there are instances which contradict the answer.

Case 1: The algorithm outputs "true". Consider the instance where the valuation of each agent is uniformly distributed in their valuable parts. We show that a connected strongly-proportional allocation cannot exist in this instance.

Suppose on the contrary that a connected strongly-proportional allocation exists. Recall that agents have equal entitlements, which means that every agent receives a piece worth more than $1/n$. Agent n's piece has to include pieces from at least two consecutive odd parts in order for her value to be greater than $1/n$, which means that agent n's piece has to contain at least one of the valuable parts of agent $i \in \{1, \ldots, n-1\}$ completely.

We now show that for each $i \in \{1, \ldots, n-1\}$, we have $a_i/(n-2) > 1-a_i$. We have

$$\frac{a_i}{n-2} - (1-a_i) = \frac{M - 2^{i-1}n^2 + n(2^{n-1}+2^i-1)+1}{(n-2)n(M+2^{i-1})} > 0,$$

where the inequality holds because $M > 2^{i-1}n^2$. This shows that $a_i/(n-2) > 1 - a_i$, which implies that each of the left valuable parts is worth more than the rightmost valuable part for agent $i \in \{1, \ldots, n-1\}$.

For $i \in \{1, \ldots, n-1\}$, since $a_i/(n-2) > 1 - a_i$, no matter which valuable part(s) of agent i is given to agent n, each of the remaining $\leq n-2$ valuable parts is worth at most $a_i/(n-2)$ to agent i. Since the valuations are uniformly distributed within each valuable part, agent i receives more than $(1/n) \div (a_i/(n-2)) = (n-2)w_i'$ of a valuable part. Therefore, agents 1 to $n-1$ receive more than

$$\sum_{i=1}^{n-1}(n-2)w_i' = (n-2)\sum_{i=1}^{n-1} w_i' = n-2$$

valuable parts in total. This is not possible, and hence, no connected strongly-proportional allocation exists.

Case 2: The algorithm outputs "false". We shall now show by construction that the information provided to the algorithm is also consistent with an instance with a connected strongly-proportional allocation, resulting in a contradiction.

Let agent n receive the rightmost two consecutive odd parts, so that agent n receives more than $1/n$. This leaves the remaining $2n-4$ parts for agents 1 to $n-1$. Removing all the non-valuable parts for all agents $i \in \{1, \ldots, n-1\}$, the remaining valuable parts of the cake are worth a_i to agent i. Divide all valuations and entitlements by a_i for each $i \in \{1, \ldots, n-1\}$—note that this does not change the existence of a connected strongly-proportional allocation. Then, this is equivalent to a cake with value 1 to every agent in $\{1, \ldots, n-1\}$ such that agent i's entitlement is $1/na_i = w_i'$. Note that $\sum_{i=1}^{n-1} w_i' = 1$, so we have reduced the problem to finding a connected strongly-proportional allocation on a modified instance with $n-1$ hungry agents such that agent $i \in \{1, \ldots, n-1\}$ has an entitlement of w_i'.

We claim that the entitlements are generic. To see this, let $N_1, N_2 \subseteq \{1, \ldots, n-1\}$ such that $\sum_{i \in N_1} w_i' = \sum_{i \in N_2} w_i'$. Since the denominators of the w_i''s are equal to each other, we have $\sum_{i \in N_1}(M + 2^{i-1}) = \sum_{i \in N_2}(M + 2^{i-1})$. Since M is larger than $\sum_{i=1}^{n-1} 2^{i-1}$, we must have $|N_1| = |N_2|$, which implies that $\sum_{i \in N_1} 2^{i-1} = \sum_{i \in N_2} 2^{i-1}$. The only way this is possible is when $N_1 = N_2$, which proves that the entitlements are generic.

Since the $n-1$ agents are hungry in this modified instance, we can use the construction in the proof of Theorem 6.3.6 for agents $1, \ldots, n-1$, which shows that, with fewer than $(n-1)2^{n-3}$ queries, the answers are consistent with the existence of a connected strongly-proportional allocation. However, note that the marks of agent $i \in \{1, \ldots, n-1\}$ between every $a_i/(n-2)$ part in Figure 6.3 are already known, which translate to the $t/(n-2)$-marks for $t \in \{1, \ldots, n-3\}$. This means that a total of $(n-1)(n-3)$ marks are known, which requires at most the same number of queries. Therefore, with fewer than $(n-1)2^{n-3} - (n-1)(n-3) \in \Omega(n \cdot 2^n)$ queries, there exists an instance consistent with the information provided by the queries that admits a connected strongly-proportional allocation. This contradicts the output of the algorithm. □

The upper bound from Theorem 6.4.5 and the lower bound from Theorem 6.4.6 imply that the number of queries required to determine the existence of a connected strongly-proportional allocation is in $\Theta(n \cdot 2^n)$, even for agents with equal entitlements. The same tight bound also holds for *computing* such an allocation if it exists—this can be shown by modifying Algorithm 6.3 slightly by following the details in the second half of the proof of Theorem 6.4.4.

Theorem 6.4.7. *The number of queries required to decide the existence of a connected strongly-proportional allocation for n agents, or to compute such an allocation if it exists, is in $\Theta(n \cdot 2^n)$, even for agents with equal entitlements.*

6.5 Stronger than Strongly-Proportional

We have so far only considered allocations which are *strongly-proportional*—agents receive pieces with value strictly more than their entitlements. Strong proportionality does not guarantee that agents receive pieces beyond just a small crumb more than their proportional piece. It would indeed be useful if we can guarantee that agents receive a fixed positive amount more than their entitlements. This motivates us to consider an even stronger fairness notion: given some fixed value $z > 0$, can each agent i receive a piece with value more than $w_i + z$?

It is easy to adapt Algorithm 6.3 to this setting by replacing w_i in Line 6 of the algorithm with $w_i + z$—this gives an upper bound number of queries to determine the existence of such an allocation, or to compute such an allocation if it exists.

Theorem 6.5.1. *Let $\mathbf{w}$ be any vector of entitlements. Then, for any positive constant z, there exists an algorithm that decides whether a connected allocation exists for n agents in which each agent i receives a piece with value more than $w_i + z$ using at most $n \cdot 2^{n-1}$ queries.*

We now show a matching lower bound, even for hungry agents with equal entitlements. The proof is similar in spirit to the proof of Theorem 6.4.6.

Theorem 6.5.2. *Let n be given. Then, for any positive constant $z < \frac{1}{n(n-1)}$, any algorithm that decides whether a connected allocation exists for n hungry agents in which each agent receives a piece with value more than $1/n + z$ requires $\Omega(n \cdot 2^n)$ queries.*

Proof. Let $n \geq 3$ and $z \in (0, \frac{1}{n(n-1)})$ be given. Define ϵ such that

$$\epsilon = \min\left\{\frac{1}{n(n-1)} - z,\ \frac{nz}{n-1}\right\};$$

note that $\epsilon > 0$. Let M be a sufficiently large constant (to be decided later), and for each $i \in \{1, \ldots, n-1\}$, define $w'_i = \frac{M+2^{i-1}}{(n-1)M+2^{n-1}-1}$ and $a_i = \frac{1/n+z}{w'_i}$. Note that we have $\lim_{M\to\infty} w'_i = \frac{1}{n-1}$; therefore, we choose a value $M \geq 2^n$ such that $\frac{1}{n-1} - \epsilon < w'_i < \frac{1}{n-1} + \epsilon$ for all $i \in \{1, \ldots, n-1\}$. Note also that $\sum_{i=1}^{n-1} w'_i = 1$.

Consider a cake with two parts—the *left* part and the *right* part. The left part is worth a_i to agent $i \in \{1, \ldots, n-1\}$ and $1 - 1/n - z$ to agent n. The right part is worth $1 - a_i$ to agent $i \in \{1, \ldots, n-1\}$ and $1/n + z$ to agent n. See Figure 6.4 for an illustration.

Agent 1	a_1	$1 - a_1$
$\vdots$	$\vdots$	$\vdots$
Agent $n-1$	a_{n-1}	$1 - a_{n-1}$
Agent n	$1 - 1/n - z$	$1/n + z$

Figure 6.4: Construction of the cake used in the proof of Theorem 6.5.2.

We now show that the valuations on the cake are valid. This is clear for agent n's valuation since $0 < 1/n + z < 1$. It suffices to show that for $i \in \{1, \ldots, n-1\}$, each of a_i and $1 - a_i$ is less than 1. We have

$$\begin{aligned}
a_i &= \frac{\frac{1}{n} + z}{w_i'} \\
&< \frac{\frac{1}{n} + z}{\frac{1}{n-1} - \epsilon} && \left(\text{since } w_i' > \tfrac{1}{n-1} - \epsilon\right) \\
&\leq \frac{\frac{1}{n} + \frac{1}{n(n-1)} - \epsilon}{\frac{1}{n-1} - \epsilon} && \left(\text{since } \epsilon \leq \tfrac{1}{n(n-1)} - z\right) \\
&= \frac{\frac{1}{n-1} - \epsilon}{\frac{1}{n-1} - \epsilon} \\
&= 1.
\end{aligned}$$

Now, instead of showing $1 - a_i < 1$, we show an even stronger statement of $1 - a_i < 1/n + z$. We have

$$\begin{aligned}
1 - a_i &= 1 - \frac{\frac{1}{n} + z}{w_i'} \\
&< 1 - \frac{\frac{1}{n} + z}{\frac{1}{n-1} + \epsilon} && \left(\text{since } w_i' < \tfrac{1}{n-1} + \epsilon\right) \\
&= \frac{1 + (n-1)\epsilon}{1 + (n-1)\epsilon} - \frac{\frac{n-1}{n} + (n-1)z}{1 + (n-1)\epsilon} \\
&= \frac{\frac{1}{n} + (n-1)\epsilon - (n-1)z}{1 + (n-1)\epsilon}
\end{aligned}$$

and

$$\frac{1}{n} + z = \frac{(1 + (n-1)\epsilon)(\frac{1}{n} + z)}{1 + (n-1)\epsilon} = \frac{\frac{1}{n} + (n-1)(\frac{1}{n} + z)\epsilon + z}{1 + (n-1)\epsilon}.$$

Taking the difference between the two expressions, we have

$$\begin{aligned}
(1 - a_i) - \left(\frac{1}{n} + z\right) &< \frac{(n-1)(1 - \frac{1}{n} - z)\epsilon - nz}{1 + (n-1)\epsilon} \\
&< \frac{(n-1)\epsilon - nz}{1 + (n-1)\epsilon} && \left(\text{since } 1 - \tfrac{1}{n} - z < 1\right) \\
&\leq \frac{nz - nz}{1 + (n-1)\epsilon} && \left(\text{since } \epsilon \leq \tfrac{nz}{n-1}\right)
\end{aligned}$$

$$= 0,$$

showing indeed that $1 - a_i < 1/n + z$. These show that the valuations are valid.

We show that any algorithm that makes fewer than $(n-1)(2^{n-3}-1) \in \Omega(n \cdot 2^n)$ queries may not be able to decide whether a connected strongly-proportional allocation exists. The proof idea is similar to that in the proofs of Theorems 6.3.4 and 6.3.6. We assume that the answer to every query made by the algorithm is consistent with the instance where the valuation of each agent is uniformly distributed in each of the left and the right parts. We show that regardless of what the algorithm outputs as its answer, there are instances which contradict the answer.

Case 1: The algorithm outputs "true". Consider the instance where the valuation of each agent is uniformly distributed in each part. Note that all agents are hungry. We show that a connected allocation in which each agent receives a piece with value more than $1/n + z$ cannot exist in this instance.

Suppose on the contrary that such an allocation exists. Assume first that agent n receives the rightmost piece. Since the right part of the cake is worth $1/n+z$ to agent n, agent n's piece has to contain the whole of the right part. The remaining $n-1$ agents' pieces are contained in the left part. For $i \in \{1, \ldots, n-1\}$, agent i receives more than $(1/n + z)/a_i = w'_i$ of the left part. Therefore, agents 1 to $n-1$ receive more than $\sum_{i=1}^{n-1} w'_i = 1$ of the left part in total. This is not possible.

Therefore, some agent $j \in \{1, \ldots, n-1\}$ receives the rightmost piece. Since $1 - a_j < 1/n + z$, agent j's piece has to contain the whole of the right part. The remaining $n-1$ agents' pieces are contained in the left part. For $i \in \{1, \ldots, n-1\} \setminus \{j\}$, agent i receives more than $(1/n + z)/a_i = w'_i$ of the left part. Agent n receives more than

$$\frac{1/n + z}{1 - 1/n - z} > \frac{1/n + z}{1 - 1/n} = \frac{1}{n-1} + \frac{nz}{n-1} \geq \frac{1}{n-1} + \epsilon > w'_j$$

of the left part, where the second inequality holds since $\epsilon \leq \frac{nz}{n-1}$. Therefore, agents in $N \setminus \{j\}$ receive more than $\sum_{i=1}^{n-1} w'_i = 1$ of the left part in total. This is not possible, and hence, no such allocation exists.

Case 2: The algorithm outputs "false". We shall now show by construction that the information provided to the algorithm is also consistent with an instance with a connected allocation in which each agent receives a piece with value more than $1/n + z$, resulting in a contradiction.

Let agent n receive the right part of the cake, so that agent n receives a piece with value $1/n+z$. While the value of this piece is not more than $1/n+z$ yet, we will fix this later. Divide all valuations of the left part of the cake and entitlements by a_i for each $i \in \{1, \ldots, n-1\}$—note that this does not change the existence of such a connected allocation. Then, this is equivalent to a cake with value 1 to every agent $i \in \{1, \ldots, n-1\}$ such that agent i needs to receive a piece of cake with value more than $\frac{1/n+z}{a_i} = w'_i$. Note that $\sum_{i=1}^{n-1} w'_i = 1$, so we have reduced the problem to finding a connected allocation on a modified instance with $n-1$ hungry agents such that agent $i \in \{1, \ldots, n-1\}$ has an entitlement of w'_i.

We claim that the entitlements are generic. To see this, let $N_1, N_2 \subseteq \{1, \ldots, n-1\}$ such that $\sum_{i \in N_1} w'_i = \sum_{i \in N_2} w'_i$. Since the denominators of the w'_i's are equal to each other, we have $\sum_{i \in N_1}(M + 2^{i-1}) = \sum_{i \in N_2}(M + 2^{i-1})$. Since M is larger than $\sum_{i=1}^{n-1} 2^{i-1}$, we must have $|N_1| = |N_2|$, which implies that $\sum_{i \in N_1} 2^{i-1} = \sum_{i \in N_2} 2^{i-1}$. The only way this is possible is when $N_1 = N_2$, which proves that the entitlements are generic.

Since the $n-1$ agents are hungry in this modified instance, we can use the construction in the proof of Theorem 6.3.6 for agents $1, \ldots, n-1$, which shows that, with fewer than $(n-1)2^{n-3}$ queries, the answers are consistent with the existence of a connected strongly-proportional allocation. Note that the marks between the left and the right part of the cake are known. This means that a total of $n-1$ marks are known, which requires at most the same number of queries. Therefore, with fewer than $(n-1)2^{n-3} - (n-1) \in \Omega(n \cdot 2^n)$ queries, there exists an instance consistent with the information provided by the queries that admits a connected strongly-proportional allocation for agents in $\{1, \ldots, n-1\}$.

We now have a connected proportional allocation such that each agent in $\{1, \ldots, n-1\}$ receives a piece with value more than $1/n + z$ and agent n receives a piece with value exactly $1/n + z$. By a similar proof as that in Lemma 6.3.1, we can slightly move the boundary of agent n's piece such that every agent receives a piece with value more than $1/n + z$. This contradicts the output of the algorithm. □

Theorem 6.5.2 shows that no algorithm can decide whether there exists a connected allocation in which the *egalitarian value* is more than $1/n + z$—the egalitarian value is the smallest value of an agent's piece in an allocation. Aumann et al. (2012) proved a closely-related result, but in a different computational model. They assume that the agents' valuations are piecewise-constant and given explicitly to the algorithm. In this model, they prove that it is NP-hard to approximate the optimal egalitarian value to a factor better than 2. Specifically, they show a reduction from an instance of the NP-hard problem 3-dimensional matching (3DM) to a cake-cutting instance. They show that if the answer to the 3DM instance is "yes", then there exists an allocation where each agent gets value at least w; and if the answer is "no", then every allocation gives some agent value at most $w/2$. In their reduction, w is at least $4/n$. In contrast, we provide an unconditional exponential lower bound in the (harder) query model. Also, our result holds for a different range of possible $w = 1/n + z$ values.

Theorems 6.5.1 and 6.5.2 give a tight bound to the number of queries, even for hungry agents with equal entitlements.

Theorem 6.5.3. *Let n be given. Then, for any positive $z < \frac{1}{n(n-1)}$, the number of queries required to decide the existence of a connected allocation for n agents in which every agent receives a piece with value more than $w_i + z$ for entitlements $\mathbf{w}$, or to compute such an allocation if it exists, is in $\Theta(n \cdot 2^n)$, even for hungry agents with equal entitlements.*

Our results complete the picture on the query complexity of computing a connected allocation that guarantees each agent a piece with a certain value. For n hungry agents, the query complexity of computing a connected allocation in which each agent receives a piece with value at least $1/n$, more than $1/n$, and more than $1/n + z$ for some small $z > 0$ is $\Theta(n \log n)$, $\Theta(n^2)$, and $\Theta(n \cdot 2^n)$ respectively, if such an allocation exists.

6.6 Pies

We now consider a *pie*, where the resource is modeled by a circle instead of by an interval. We use $C = [0, 1)$ to represent the pie, but in contrast to the cake version, the endpoints 0 and 1 are "joined" together—they are considered the same point. Therefore, the piece $[0, a] \cup [b, 1)$ is also considered a connected piece for any $a, b \in C$ with $a \leq b$. Several papers have studied the special properties of pie-cutting (Stromquist, 2007; Thomson, 2007; Brams et al., 2008; Barbanel et al., 2009).

We show that the problem of deciding the existence of a connected strongly-proportional allocation of a pie is intractable.

Theorem 6.6.1. *No finite algorithm can decide the existence of a connected strongly-proportional allocation of a pie, even for hungry agents with equal entitlements.*

Proof. Suppose by way of contradiction that some finite algorithm decides the existence of a connected strongly-proportional allocation of a pie for n agents with equal entitlements. Assume that the information provided to the algorithm by eval and mark queries is consistent with that where every agent's valuation is uniformly distributed over the pie (in which case there is no connected strongly-proportional allocation of the pie), and so the algorithm should output "false". However, we shall now show that the information provided to the algorithm is also consistent with an instance with a connected strongly-proportional allocation. This means that the algorithm is not able to differentiate between the two, resulting in a contradiction.

Let P be the set of all points on the pie mentioned by the algorithm or by the queries—for example, if an $\textsc{Eval}_i(x, y)$ query is made by the algorithm, or if a $\textsc{Mark}_i(x, r)$ query is made by the algorithm and y is returned, then x and y are added to P. Since the algorithm is finite, P is finite. For $x \in C$, define $\bar{x} = \{x, x + 1/n, \ldots, x + (n-1)/n\} \subseteq C$ where all numbers in the set are modulo 1. (From now on, every point mentioned is modulo 1.) Since P is finite, there exists a point $x \in C$ such that $\bar{x} \cap P = \varnothing$. Fix x. Let $\epsilon \in (0, 1/n)$ be a number smaller than the distance between any element in $\bar{x}$ and any element in P.

Construct agent 1's valuation function such that $u_1([x, p]) = p - x$ for all $p \in P$, $u_1([x, x + 1/n + \epsilon]) = 1/n$, and its distribution between every two adjacent points in $P \cup \{x + 1/n + \epsilon\}$ is uniform within the respective intervals based on these valuations—note that this construction is valid and unique since these known r-marks (starting at the point x) are strictly increasing in r. All other $n - 1$ agents have valuation functions uniformly distributed over the pie. Note that all agents are hungry. By changing the axis to start at the point x, we see that the $1/n$-mark (starting at the point x) of agent 1 is at $x + 1/n + \epsilon$ while that of the other agents are at $x + 1/n$. Therefore, the $1/n$-mark of agent 1 is different from that of the other agents. By Theorem 6.3.2, there exists a connected strongly-proportional allocation starting from the point x. This means that the algorithm is not able to differentiate between the two instances. □

6.7 Conclusion

We have studied necessary and sufficient conditions for the existence of a connected strongly-proportional allocation on the interval cake (Theorems 6.3.2 and 6.4.4). We have shown that computing this condition requires $\Theta(n \cdot 2^n)$ queries even for agents with equal entitlements (Theorem 6.4.7) or hungry agents with generic entitlements (Theorem 6.3.7), and $\Theta(n^2)$ for hungry agents with equal entitlements (Theorem 6.3.5). The same bounds hold for the computation of such an allocation if it exists. We have also shown that for connected allocations where each agent receives a small value z more than their proportional share, the number of queries to decide the existence of such allocations is in $\Theta(n \cdot 2^n)$ (Theorem 6.5.3). Finally, we have shown that no finite algorithm can decide the existence of a connected strongly-proportional allocation of a pie (Theorem 6.6.1).

A natural question that arose from this chapter is whether there is an algorithm that (asymptotically) attains the lower bound in (6.1) for hungry agents with entitlements that are neither generic nor equal.

Additionally, this chapter can be extended in the following ways:

- **Chores.** *Chore-cutting* is a variant of cake-cutting in which agents have *negative* valuations for every piece of the cake.
- **Beyond the unit interval.** We can consider cakes with more complex topologies, such as graphical cakes (Bei and Suksompong, 2021), tangled cakes (Igarashi and Zwicker, 2024), and two-dimensional cakes (Segal-Halevi et al., 2017).
- **Envy-freeness.** It is known that, in every cake-cutting instance, a connected envy-free allocation exists (Stromquist, 1980; Su, 1999). What conditions are necessary and sufficient for the existence of a connected strongly-proportional allocation that is also envy-free?

We may also consider a weaker fairness notion of *proportionality* instead—we give a brief discussion in Appendix C.2.

Chapter 7

Approximate Envy-Freeness in Graphical Cake Cutting

7.1 Introduction

Cake cutting refers to the classic problem of fairly allocating a divisible resource such as land or advertising spaces—playfully modeled as a "cake"—among agents who may have different values for different parts of the resource (Robertson and Webb, 1998; Procaccia, 2013). The most common fairness criteria in this literature are *proportionality* and *envy-freeness*. Proportionality demands that if there are n agents among whom the cake is divided, then every agent should receive at least $1/n$ of her value for the entire cake. Envy-freeness, on the other hand, requires that no agent would rather have another agent's piece of cake than her own. Early work in cake cutting established that a proportional and envy-free allocation that assigns to each agent a connected piece of cake always exists, regardless of the number of agents or their valuations over the cake (Dubins and Spanier, 1961; Stromquist, 1980; Su, 1999).

Although existence results such as the aforementioned guarantees indicate that a high level of fairness can be achieved in cake cutting, they rely on a typical assumption in the literature that the cake is represented by an interval. This representation is appropriate when the resource corresponds to machine processing time or a single road, but becomes insufficient when one wishes to divide more complex resources such as networks. For example, one may wish to divide road networks, railway networks, or power cable networks among different companies for the purpose of construction or maintenance. In light of this observation, Bei and Suksompong (2021) introduced a more general model called *graphical cake cutting*, wherein the cake can be represented by any connected graph. With a graphical cake, a connected proportional allocation may no longer exist—see Figure 7.1 (left). Nevertheless, these authors showed that more than half of the proportionality guarantee can be retained: any graphical cake admits a connected allocation such that every agent receives at least $1/(2n-1)$ of her entire value.

The result of Bei and Suksompong (2021) demonstrates that approximate proportionality is attainable in graphical cake cutting. However, the allocation that their algorithm

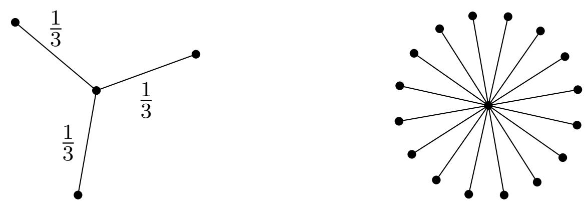


Figure 7.1: (Left) A star graph with three edges of equal length. Two agents with identical valuations distributed uniformly over the three edges cannot each receive a connected piece worth at least 1/2 of the whole cake at the same time. (Right) A star graph with many edges to be divided between two agents. If sharing of vertices is disallowed, then the agent who does not receive the center vertex will be restricted to at most one edge, and will incur envy equal to almost the value of the entire cake.

produces may lead to high *envy* between the agents. In particular, while each agent i is guaranteed $1/(2n-1)$ of her value, it is possible that the algorithm assigns the remaining $(2n-2)/(2n-1)$ of the value to another agent j from i's perspective, so that i envies j by almost the entire value of the cake (for large n) when measured additively, and by a factor linear in n when measured multiplicatively. Note that envy-freeness is a much more stringent benchmark than proportionality—for instance, although there exists a simple protocol for computing a connected proportional allocation of an interval cake (Dubins and Spanier, 1961), no finite protocol can compute a connected envy-free allocation of it (Stromquist, 2008), and even without the connectivity requirement, the only known envy-free protocol requires an enormous number of queries (Aziz and Mackenzie, 2016). The goal of this chapter is to investigate the existence of connected allocations of a graphical cake with low envy, as well as to design algorithms for computing such allocations.

7.1.1 Our Results

We assume that the cake is represented by the edges of a connected graph, and each edge can be subdivided into segments to be allocated to different agents. Each agent is to receive a connected piece, though we will also briefly explore relaxations of this constraint in Section 7.5. The whole cake must be allocated, and each agent's value for it is additive and normalized to 1; the value of each agent does not need to be uniform within each edge or across different edges. Following Bei and Suksompong (2021), we also assume that each vertex can be shared by multiple agents.[1] We consider both additive envy—for $\alpha \in [0,1]$, an allocation is *α-additive-EF* if no agent envies another agent by an *amount* of more than α—and multiplicative envy—for $\alpha \geq 1$, an allocation is *α-EF* if no agent envies another agent by a *factor* of more than α. An α-EF allocation is also $\left(\frac{\alpha-1}{\alpha+1}\right)$-additive-EF; we refer to Proposition 2.3.1 for details.

In Section 7.3, we consider agents with (possibly) non-identical valuations. We show that for any graph, there exists a 1/2-additive-EF allocation, and such an allocation can be computed by iteratively allocating to each agent a share that other agents do not value too highly. If the graph is a star, we present an algorithm that, for any $\epsilon > 0$, finds a $(3+\epsilon)$-EF

[1]Without this assumption, one cannot obtain nontrivial envy-free guarantees—see the caption of Figure 7.1 (right) for a brief discussion.

allocation (which is therefore nearly 1/2-additive-EF as well) by allowing agents to repeatedly relinquish their current share for a higher-value share, and allocating the remaining shares by following certain rules. Our two algorithms generalize ideas from algorithms for the interval cake by Goldberg et al. (2020) and Arunachaleswaran et al. (2019), respectively. We remark here that star graphs are of particular interest in graphical cake cutting because they constitute perhaps the most intuitive generalization of the well-studied interval cake, and therefore provide a natural platform for attempts to extend techniques and results from the interval-cake setting.[2]

Next, in Section 7.4, we demonstrate how the bounds for non-identical valuations can be improved in the case of identical valuations; this case captures scenarios in which there is an objective valuation among agents.[3] For arbitrary graphs, we devise an algorithm that computes a $(2+\epsilon)$-EF allocation (which is therefore nearly 1/3-additive-EF). Our algorithm is inspired by the work of Chu et al. (2010) on partitioning edges of a graph (see Section 7.1.2), and involves repeatedly adjusting the shares along a path from the minimum share to the maximum share so that the shares become more balanced in value. For star graphs, we provide a simpler algorithm that returns a 2-EF allocation using a bag-filling idea. As we discuss at the start of Section 7.2, an approximate proportionality result of Bei and Suksompong (2021) implies that both of our guarantees in this section are (essentially) tight.

Finally, in Section 7.5, we explore the fairness guarantees when each agent can receive more than one connected piece. We introduce the notion of *path similarity number* to discuss the relationship between connected interval cake cutting and (non-connected) graphical cake cutting.

Our results in Sections 7.3 and 7.4 are summarized in Table 7.1. All of our algorithms can be implemented in the standard cake-cutting model of Robertson and Webb (1998) in time polynomial in n, the size of the graph, and, if applicable, $1/\epsilon$.

	general graphs	**star graphs**
non-identical valuations	1/2-additive-EF (Thm. 7.3.1)	$(3+\epsilon)$-EF (Thm. 7.3.2)
identical valuations	$(2+\epsilon)$-EF (Thm. 7.4.2)	2-EF (Thm. 7.4.6)

Table 7.1: Summary of results in Sections 7.3 and 7.4.

7.1.2 Further Related Work

Cake cutting is a topic of constant interest for researchers in mathematics, economics, and computer science alike. For an overview of its intriguing history, we refer to the books by Brams and Taylor (1996) and Robertson and Webb (1998), as well as the book chapter by Procaccia (2016).

The cake-cutting literature traditionally assumes that the cake is given by an interval, and connectivity of the cake allocation is often desired in order to avoid giving agents a "union of

[2]Star graphs (and path graphs) are also often studied in the context of indivisible items (see Section 7.1.2). In graphical cake cutting, all path graphs are equivalent to the classic interval cake, which is why the role of star graphs is further highlighted.

[3]We discuss further motivation for investigating this case at the beginning of Section 7.4.

crumbs" (Stromquist, 1980, 2008; Su, 1999; Bei et al., 2012; Cechlárová and Pillárová, 2012; Cechlárová et al., 2013; Aumann and Dombb, 2015; Arunachaleswaran et al., 2019; Goldberg et al., 2020; Segal-Halevi and Suksompong, 2021, 2023; Barman and Kulkarni, 2023).[4] Besides Bei and Suksompong (2021), a few authors have recently addressed the division of a graphical cake. Igarashi and Zwicker (2024) focused on envy-freeness but made the crucial assumption that vertices cannot be shared between agents—as discussed in the caption of Figure 7.1 (right), with their assumption, one cannot obtain nontrivial guarantees even for star graphs and identical valuations. Deligkas et al. (2022) explored the complexity of deciding whether an envy-free allocation exists for a given instance (and, if so, finding one), both when vertices can and cannot be shared, but did not consider approximate envy-freeness. Elkind et al. (2021) investigated another fairness notion called maximin share fairness in graphical cake cutting.

Several recent papers have examined connectivity constraints in the allocation of *indivisible items* represented by vertices of a graph (Bouveret et al., 2017; Igarashi and Peters, 2019; Suksompong, 2019; Lonc and Truszczynski, 2020; Deligkas et al., 2021; Bei et al., 2022, 2024; Bilò et al., 2022; Caragiannis et al., 2022; Gahlawat and Zehavi, 2023). In particular, Caragiannis et al. (2022) assumed that some vertices can be shared by different agents; this assumption allowed them to circumvent the strong imbalance in the case of star graphs. A number of authors considered the problem of dividing *edges* of a graph, where, unlike in graphical cake cutting, each edge is treated as an indivisible object (Wu et al., 2007; Chu et al., 2010, 2013).

Another line of work also combines cake cutting and graphs, but the graph represents the acquaintance relation among agents (Abebe et al., 2017; Bei et al., 2017, 2020; Ghalme et al., 2023; Tucker-Foltz, 2023). Despite the superficial similarity, this model is very different from graphical cake cutting, and there are no implications between results on the two models.

7.2 Preliminaries

Refer to the preliminaries in Sections 2.1 and 2.3. We now describe other preliminaries specific to this chapter.

Let $G = (V, E)$ be a connected undirected graph representing the cake. Each edge $e \in E$, isomorphic to the interval $[0, 1]$ of the real numbers, is denoted by $e = [v_1, v_2] = [v_2, v_1]$ where $v_1, v_2 \in V$ are the endpoints of edge e. If x_1 and x_2 are points *on* an edge, then the segment between them is denoted by $[x_1, x_2]$ or $[x_2, x_1]$—we sometimes call it an *interval*. We shall restrict our attention to closed intervals only.

Two intervals of a cake G are considered *disjoint* if their intersection is a finite set of points. A *share* of a cake G is a finite union of pairwise disjoint (closed) intervals of G—where the intervals may belong to different edges—such that it is connected, i.e., for any two points in the share, there exists a path between the two points that only traverses the share. As with intervals, two shares of a cake G are considered *disjoint* if their intersection is a finite

[4] While connectivity is the most frequently studied constraint in cake cutting, other constraints, such as geometric and separation constraints, have also been explored (Suksompong, 2021).

set of points. A share is *unallocated* if its intersection with every agent's share is a finite set of points.

An *instance* of graphical cake cutting consists of a graph G, a set of agents N, and their utility functions $(u_i)_{i \in N}$.

Bei and Suksompong (2021) demonstrated that for every $n \geq 2$, there exists an instance in which no allocation is α-proportional for any $\alpha < 2 - 1/n$, even for identical valuations and star graphs. By Proposition 2.3.1, one cannot obtain a better guarantee than 2-EF for such instances.

We now state a useful lemma about the existence of a share that has sufficiently high value for one agent and, at the same time, not exceedingly high value for other agents.

Lemma 7.2.1. *Let H be a connected subgraph of a graphical cake, and suppose that H is worth β_0 to some agent in a subset $N' \subseteq N$. Then, for any positive $\beta \leq \beta_0$ and any vertex r of H, there exists an algorithm, running in time polynomial in n and the size of H, that finds a partition of H into two (connected) shares such that the first share is worth at least β to some agent in N' and less than 2β to every agent in N', and the second share contains the vertex r.*

Bei and Suksompong (2021, Lemma 4.9) made this claim for the special case where all agents have the same value for H and no vertex r is specified. We will use Lemma 7.2.1 as a subroutine in IterativeDivide (Algorithm 7.1) and BalancePath (Algorithm 7.4); IterativeDivide considers the case where different agents may have different values for H, while BalancePath requires the condition on the vertex r in order to maintain connectivity along the minimum-maximum path. We shall use $\text{Divide}(H, N', \beta, r)$ to denote the ordered pair of the two corresponding shares as described in the lemma. The idea behind the proof of Lemma 7.2.1 is similar to that of the special case shown by Bei and Suksompong (2021): we convert H into a tree rooted at the vertex r by removing cycles iteratively—keeping the edges and duplicating the vertices if necessary—then traverse the tree from r until a vertex v with a subtree of an appropriate size is reached, and finally identify some connected subgraph of the subtree as the first share while assigning the remaining portion as the second share. The full details, including the pseudocode, are given in Appendix D.

7.3 Possibly Non-Identical Valuations

In this section, we allow agents to have different valuations. For arbitrary graphs, we present an algorithm that computes an approximately envy-free allocation of a graphical cake when measured additively. For the case where the graph is a star, we give an algorithm that finds an allocation wherein the envy is bounded by a multiplicative factor of roughly 3.

7.3.1 General Graphs

A priori, it is not even clear whether there exists a constant $\alpha < 1$ independent of n such that an α-additive-EF allocation always exists. We now describe the algorithm, Iterative-Divide (Algorithm 7.1), which finds a 1/2-additive-EF allocation for arbitrary graphs and

Algorithm 7.1 ITERATIVEDIVIDE(G, N).

Input: Graph G, set of agents $N = \{1, \dots, n\}$.
Output: Allocation $(A_1, \dots, A_n)$.
Initialization: $r \leftarrow$ any vertex of G; $H_n \leftarrow G$; $N' \leftarrow N$.

for $i = 1, \dots, n-1$ **do**
  $\beta_i \leftarrow 1/4$
  if there exists $i' \in N'$ such that $u_{i'}(H_n) \geq \beta_i$ **then**
    $(H_i, H_n) \leftarrow$ DIVIDE(H_n, N', β_i, r)
    $i^* \leftarrow$ any agent in N' who values H_i at least β_i
  else
    $H_i \leftarrow \varnothing$
    $i^* \leftarrow$ any agent in N'
  end if
  $A_{i^*} \leftarrow H_i$
  $N' \leftarrow N' \setminus \{i^*\}$
end for
$A_j \leftarrow H_n$, where j is the remaining agent in N'
return $(A_1, \dots, A_n)$

non-identical valuations, using ideas similar to the algorithm by Goldberg et al. (2020) for computing a 1/3-additive-EF allocation of an interval cake. Choose any arbitrary vertex r of G, and start with the entire graph G and all agents in contention. If there is only one agent remaining, allocate the remaining graph to that agent. If the remaining graph is worth less than $\beta = 1/4$ to every remaining agent,[5] allocate an empty graph to any one of the remaining agents and remove this agent. Otherwise, apply the algorithm DIVIDE on the remaining graph and the remaining agents with threshold β. Allocate the first share to any agent who values that share at least β, and remove this agent along with her share. Repeat the procedure with the remaining graph until the whole graph is allocated. We claim that the resulting allocation is indeed 1/2-additive-EF.

Theorem 7.3.1. *Given an instance of graphical cake cutting, there exists an algorithm that computes a* $1/2$*-additive-EF allocation in time polynomial in* n *and the size of* G.

Proof. We claim that the algorithm ITERATIVEDIVIDE (Algorithm 7.1) satisfies the condition. It is clear that the algorithm can be implemented in polynomial time; it remains to check that the allocation returned by the algorithm is 1/2-additive-EF. Let $i \in N$, and let $N_0 \subseteq N$ be the subset of agents who were allocated shares that correspond to the first share of some DIVIDE procedure called by ITERATIVEDIVIDE. If $i \in N_0$, then agent i receives a share worth at least $\beta = 1/4$ to her by Lemma 7.2.1, so every other agent receives a share worth at most $1 - 1/4 = 3/4$ to agent i, and agent i's envy is at most $3/4 - 1/4 = 1/2$. Else, $i \notin N_0$, and every agent in N_0 receives a share worth less than $2\beta = 1/2$ to agent i by Lemma 7.2.1, while every agent in $N \setminus N_0$ receives a share worth less than $\beta = 1/4 < 1/2$ to agent i, so agent i's envy is again at most $1/2$. □

While an additive envy of $1/2$ can be seen as high, the left example of Figure 7.1 shows that

[5]While the value of β is the same for all iterations here, we write β_i in the pseudocode because we will later consider a generalization in which β can be different for different iterations.

an envy of 1/3 is inevitable. Moreover, even for an interval cake, the (roughly) 1/4-additive approximation of Barman and Kulkarni (2023) is the current best as far as polynomial-time computability is concerned.

Although IterativeDivide guarantees that the envy between each pair of agents is at most 1/2, it is possible that some agents receive an empty share from the algorithm. In the remainder of this chapter, we present algorithms that find approximately envy-free allocations up to constant multiplicative factors for star graphs as well as for agents with identical valuations. Any such allocation ensures positive value for every agent and, by Proposition 2.3.1, is also approximately envy-free when measured additively.

7.3.2 Star Graphs

The case of star graphs presents a natural generalization of the canonical interval cake and, as can be seen in Figure 7.1, already highlights some of the challenges that graphical cake cutting poses. For this class of graphs, we devise an algorithm that, for any constant $\epsilon > 0$, computes a $(3+\epsilon)$-EF allocation in polynomial time. The algorithm consists of four phases. It starts with an empty partial allocation and finds a small star of "stubs" near the center vertex (Phase 1). It then repeatedly finds an unallocated share worth slightly more than some agent's share, and allows that agent to relinquish her existing share for this new share—care must be taken to ensure that other agents do not have too much value for this new share (Phase 2). This new share could be a segment of an edge (Phase 2a) or a union of multiple complete edges (Phase 2b). This phase is repeated until there are no more unallocated shares suitable for agents to trade with. Finally, the unallocated shares are appended to the agents' existing shares (Phases 3 and 4). See Figure 7.2 for an illustration of each phase. We remark that Phases 2a and 3 of our algorithm are adapted from the algorithm of Arunachaleswaran et al. (2019) for finding a $(2+\epsilon)$-EF allocation of an interval cake.

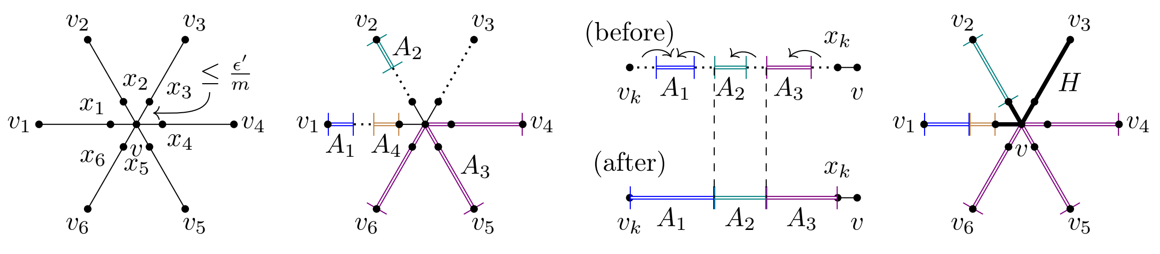


Figure 7.2: (a) The points x_k are found, where $[x_k, v]$ is worth at most ϵ'/m to every agent. (b) The unallocated intervals (dotted lines) are the ones to be considered in Phase 2a. (c) The unallocated intervals (dotted lines) are appended leftwards in v_k's direction, except for the one containing v_k which is appended rightwards. (d) The remaining unallocated portion H (bold lines) is a share connected by v.

Let $G = (V, E)$ be a star graph centered at vertex v with $m \geq 2$ edges. Label the other vertices v_k and the edges $e_k = [v_k, v]$ for $k \in \{1, \ldots, m\}$. Fix any $\epsilon \in (0, 1)$.

Phase 1: Preparation. Define $\epsilon' = \frac{\epsilon}{16nm}$. Initialize an empty partial allocation $\mathcal{A} = (A_1, \ldots, A_n)$. For each edge e_k, find a point $x_k \in [v_k, v]$ such that the segment $[x_k, v]$ is worth at most ϵ'/m to all agents. Define $e_k^1 = [v_k, x_k]$ and $e_k^2 = [x_k, v]$ for $k \in \{1, \ldots, m\}$, and let E^1 and E^2 be the sets containing all e_k^1's and all e_k^2's, respectively. Note that E^2 is worth at most ϵ' to every agent.

Phase 2: Increase agents' shares incrementally. If there is a segment $e_k^1 \in E^1$ such that some unallocated interval within e_k^1 is worth at least $u_i(A_i) + \epsilon'$ to some agent i, **go to Phase 2a**. Otherwise, consider the segments in E^1 that are entirely unallocated. If the union of these segments is worth at least $u_i(A_i) + \epsilon'$ to some agent i, **go to Phase 2b**. Otherwise, Phase 2 ends; **go to Phase 3**.

- **Phase 2a: Allocate a subinterval of some e_k^1.** Pick an unallocated interval $I \subseteq e_k^1$ that is worth at least $u_i(A_i) + \epsilon'$ to some agent i, and assume without loss of generality that it cannot be extended in either direction without overlapping an allocated share or e_k^2. Suppose that $I = [a, b]$, where a is closer to v_k than b is. If $a = v_k$, find the point $z \in I$ closest to a such that $[a, z]$ is worth exactly $u_{i^*}(A_{i^*}) + \epsilon'$ to some agent i^*, and let $A_{i^*} = [a, z]$, i.e., agent i^* relinquishes her existing share for this new share. Else, $a \neq v_k$; find the point $z \in I$ closest to b such that $[z, b]$ is worth exactly $u_{i^*}(A_{i^*}) + \epsilon'$ to some agent i^*, and let $A_{i^*} = [z, b]$. **Repeat Phase 2.**

- **Phase 2b: Allocate multiple edges in E.** Let K_0 be the set of all indices k such that the entire segment e_k^1 is unallocated. Initialize $K = \varnothing$, and add the indices from K_0 to K one by one until $\{e_k^1 \mid k \in K\}$ is worth at least $u_{i^*}(A_{i^*}) + \epsilon'$ to some agent i^*. Let A_{i^*} be the union of e_k over all $k \in K$, i.e., agent i^* relinquishes her existing share for this new share. Note that this new share is connected by the center vertex v. **Repeat Phase 2.**

Phase 3: Append unallocated subintervals within e_k^1. Let $N_1 \subseteq N$ consist of all agents who last received a subinterval of some e_k^1 via Phase 2a, and $N_2 \subseteq N$ consist of all agents who last received two or more complete edges in E via Phase 2b (we will show later that, in fact, $N_1 \cup N_2 = N$). For each e_k^1 of which some agent from N_1 is allocated a subinterval, and for each unallocated interval $I = [a, b] \subseteq e_k^1$ that cannot be extended in either direction without overlapping an allocated share or E^2, where a is closer to v_k than b is, append I to the share of the agent who is allocated the point a (i.e., append towards v_k's direction). The only time this is not possible is when $a = v_k$, in which case we append to the share of the agent who is allocated the point b.

Phase 4: Append H. Consider the remaining unallocated portion H of the graph. Note that for each $k \in \{1, \ldots, m\}$, we have $H \cap e_k = \{v\}$ or e_k^2 or e_k—this means that H is connected by the center vertex v. If N_2 is nonempty, append H to the share of an arbitrary agent in N_2. Else, if some segment e_k^1 is allocated to at least two agents, give H to the agent who has been allocated the point x_k. Otherwise, we know that every agent is allocated exactly one segment in E^1—give H to the agent who traded her share last in Phase 2 (in particular, Phase 2a).

We claim that this algorithm yields a $(3+\epsilon)$-EF allocation. By Proposition 2.3.1, such an allocation is roughly 1/2-additive-EF and (by taking $\epsilon = 1/n$) 3-proportional as well.

Theorem 7.3.2. *Given an instance of graphical cake cutting consisting of a star graph with m edges, there exists an algorithm that, for any $\epsilon > 0$, computes a $(3+\epsilon)$-EF allocation in time polynomial in n, m, and $1/\epsilon$.*

For the sake of exposition, we shall introduce notations to differentiate the partial allocations at different stages of the algorithm. For an integer $t \geq 0$, let $\mathcal{P}^t = (P_1^t, \ldots, P_n^t)$ be the partial allocation after t iterations of Phase 2, and let $\mathcal{P} = (P_1, \ldots, P_n)$ be the partial allocation at the start of Phase 3 (we show in Lemma 7.3.3 that it is a valid partial allocation). For any share P, let $\widehat{P} = \bigcup_{k=1}^{m}(P \cap e_k^1)$. The final allocation (i.e., after Phase 4) shall be denoted $\mathcal{A} = (A_1, \ldots, A_n)$. We establish the approximate envy-freeness of $\mathcal{A}$ via a series of intermediate results.

Lemma 7.3.3. *$\mathcal{P}$ is a valid partial allocation, and is equal to $\mathcal{P}^t$ for some $t \leq \frac{16n^2m}{\epsilon}$.*

Proof. To show that $\mathcal{P}$ is a valid partial allocation, we prove by induction that $\mathcal{P}^t$ is a valid partial allocation for every t. In particular, we check that each P_i^t is connected, and the agents' shares in $\mathcal{P}^t$ are pairwise disjoint. Clearly, the empty partial allocation, $\mathcal{P}^0$, is a valid partial allocation. Now, assume that $\mathcal{P}^t$ is a valid partial allocation; we will prove the validity of $\mathcal{P}^{t+1}$. At the $(t+1)^{\text{th}}$ iteration of Phase 2, some agent i^* trades her share in either Phase 2a or 2b, while all other agents' shares remain unchanged, so we only need to check that the share of agent i^* is connected and disjoint from other agents' shares.

- If agent i^* trades her share in Phase 2a, then she receives a connected subinterval of some e_k^1; furthermore, this subinterval is disjoint from other agents' shares since it is a subset of some unallocated interval $I = [a, b]$.
- If agent i^* trades her share in Phase 2b, then she receives a collection of edges e_k which are connected by the vertex v; furthermore, for any $k \in \{1, \ldots, m\}$, if e_k^1 is unallocated, then e_k^2 is unallocated as well, so the e_k's received by agent i^* are not allocated to any other agent.

Hence, $\mathcal{P}^{t+1}$ is a valid partial allocation. This completes the induction.

In each iteration of Phase 2, some agent increases the value of her share by at least ϵ'. Since the value of each agent's share starts from 0 and cannot exceed 1, the total number of increments is at most $1/\epsilon'$ for each agent. As there are n agents, the total number of iterations of Phase 2 is at most $n/\epsilon' = 16n^2m/\epsilon$. □

Lemma 7.3.4. *Fix any $i \in N$.*

- *For any $j \in N_1$, we have $u_i(P_j) \leq u_i(P_i) + \epsilon'$.*
- *For any $j \in N_2$, we have $u_i(\widehat{P}_j) \leq 2(u_i(P_i) + \epsilon')$.*

Proof. We prove that the statements are true for each $\mathcal{P}^t$ by induction on t. Let N_1^t and N_2^t be the sets of agents whose share in $\mathcal{P}^t$ was last obtained via Phase 2a and Phase 2b, respectively. Note that $N_1^0 = N_2^0 = \varnothing$, and if agent i^* obtains her share via Phase 2a in the t^{th} iteration of Phase 2, then $N_1^t = N_1^{t-1} \cup \{i^*\}$ and $N_2^t = N_2^{t-1} \setminus \{i^*\}$; an analogous statement holds if i^* obtains her share via Phase 2b. For the induction, we need to prove the following two statements.

- For any $j \in N_1^t$, we have $u_i(P_j^t) \leq u_i(P_i^t) + \epsilon'$.
- For any $j \in N_2^t$, we have $u_i(\widehat{P}_j^t) \leq 2(u_i(P_i^t) + \epsilon')$.

The statements are clearly true for the empty partial allocation $\mathcal{P}^0$, as all shares have zero value. Now, assume that the statements are true for $\mathcal{P}^t$; we shall prove the same for $\mathcal{P}^{t+1}$. Only the share of one agent i^* has changed, so we can focus on the case where either $i = i^*$ or $j = i^*$. Since the statements trivially hold for $i = j$, we may assume that $i \neq j$.

- If $i = i^*$, then we have $P_j^{t+1} = P_j^t$ and $u_i(P_i^{t+1}) \geq u_i(P_i^t)$, so both statements hold for $t + 1$.
- If $j = i^*$, then agent j trades her share in either Phase 2a or Phase 2b in the $(t+1)^{\text{th}}$ iteration of Phase 2.
  - If $j \in N_1^{t+1}$, then j trades in Phase 2a. By our procedure in Phase 2a, agent j's share is not worth more than $u_i(P_i^t) + \epsilon' = u_i(P_i^{t+1}) + \epsilon'$ to agent i; otherwise agent i would have gotten a strict subinterval of P_j^{t+1} instead. Therefore, the first statement holds for $t + 1$.
  - If $j \in N_2^{t+1}$, then j trades in Phase 2b. Recall that the set K in Phase 2b was formed by adding indices from K_0 one by one. Let K_1 be the subset of K without the last index added. Then each of $\{e_k^1 \mid k \in K_1\}$ and $\{e_k^1 \mid k \in K \setminus K_1\}$ is worth less than $u_i(P_i^t) + \epsilon' = u_i(P_i^{t+1}) + \epsilon'$ to agent i, so their union, $\widehat{P}_j^{t+1}$, is worth less than $2(u_i(P_i^{t+1}) + \epsilon')$ to agent i. Therefore, the second statement holds for $t + 1$.

This completes the induction. By Lemma 7.3.3, $\mathcal{P} = \mathcal{P}^t$ for some t, and so the statements in Lemma 7.3.4 hold. ☐

Lemma 7.3.5. *Every agent receives a share in $\mathcal{P}$ worth at least $\frac{1}{4nm}$ to her.*

Proof. Suppose by way of contradiction that some agent i receives a share worth less than $\frac{1}{4nm}$ to her. Since $\epsilon < 1$ and $\epsilon' = \frac{\epsilon}{16nm}$, we have $u_i(P_i) + \epsilon' < \frac{1}{4nm} + \epsilon' < \frac{5}{16nm}$. Every part of the graph G can be classified into one of the following three cases.

- **Case 1: Within E^1 and within an agent's share.**
  By Lemma 7.3.4, for each j, the value of $\widehat{P}_j$ is at most $2(u_i(P_i) + \epsilon') < \frac{5}{8nm}$ to agent i. (If $j \notin N_1 \cup N_2$, then P_j is empty.) As there are n agents, the union of these $\widehat{P}_j$'s is worth at most $n \cdot \left(\frac{5}{8nm}\right) < \frac{5}{8}$ to agent i.
- **Case 2: Within an unallocated subinterval of some segment in E^1.**
  For each $k \in \{1, \ldots, m\}$, if n_k agents are allocated some subinterval of segment $e_k^1 \in E^1$, then there are at most $n_k + 1$ unallocated subintervals on the same segment. Therefore, altogether there are at most $n + m$ unallocated subintervals within all the segments in

E^1. Since Phase 2 terminated, each of these subintervals is worth less than $u_i(P_i)+\epsilon' < \frac{5}{16nm}$ to agent i. Therefore, the total value of these subintervals to agent i is less than $(n+m)\left(\frac{5}{16nm}\right) \leq \frac{5}{16}$, where the inequality holds because $n, m \geq 2$.

- **Case 3: Within E^2.**
  By definition, E^2 is worth at most $\epsilon' < \frac{1}{16}$ to agent i.

The whole cake is thus worth less than $\frac{5}{8}+\frac{5}{16}+\frac{1}{16} = 1$ to agent i, which is a contradiction. □

Lemma 7.3.5 implies that no agent receives an empty share in $\mathcal{P}$, that is, $N_1 \cup N_2 = N$. With this lemma in hand, we are now ready to prove Theorem 7.3.2.

Proof of Theorem 7.3.2. Without loss of generality, we may assume that $\epsilon \in (0,1)$. The running time claim holds because each iteration of each phase runs in time polynomial in n and m, and the number of iterations of Phase 2 is polynomial in n, m, and $1/\epsilon$ by Lemma 7.3.3.

Fix any $i, j \in N$; we shall first show that A_j is worth at most $3u_i(P_i) + 4\epsilon'$ to agent i. To this end, we consider three cases for j. Recall the definition of H from Phase 4 of the algorithm.

- **Case 1: $j \in N_1$ and A_j does not contain H.**
  By Lemma 7.3.4, we have $u_i(P_j) \leq u_i(P_i) + \epsilon'$. Note that at the start of Phase 3, every unallocated subinterval of any e_k^1 is worth less than $u_i(P_i) + \epsilon'$ to agent i; otherwise Phase 2 would have continued. Since agent j is allocated at most two such subintervals, we have $u_i(A_j) \leq 3(u_i(P_i) + \epsilon') \leq 3u_i(P_i) + 4\epsilon'$.

- **Case 2: $j \in N_1$ and A_j contains H.**
  Since $j \in N_1$, P_j is a subinterval of some e_k^1. Let $P_j = [a, b]$, where a is closer to v_k than b is. We claim that in this case, unlike in Case 1, agent j receives at most one unallocated subinterval of e_k^1 in Phase 3. Note that since A_j contains H, no agent held a subinterval of $[b, x_k]$ at the start of Phase 3.

  - If e_k^1 is allocated to at least two agents during Phase 3, then some other agent held a subinterval of $[v_k, a]$ at the start of Phase 3—let such a subinterval closest to a be $[y, z]$, where $v_k \leq y < z \leq a$. The unallocated interval $[z, a]$ at the start of Phase 3 (if it is nonempty) is appended to the share of the agent who held $[y, z]$ at the start of Phase 3. Thus, agent j receives only the unallocated subinterval $[b, x_k]$ of e_k^1 in Phase 3.
  - If e_k^1 is allocated only to agent j during Phase 3, then since A_j contains H, by the description of how H is allocated in Phase 4, agent j was the last agent who traded her share in Phase 2, in particular, Phase 2a. This means that either e_k^1 was entirely unallocated just before agent j received a share from it, or it was allocated only to j at that point. Hence, j's share is of the form $[a, b] = [v_k, b]$ or $[a, b] = [a, x_k]$. It follows that j receives only one unallocated subinterval $[b, x_k]$ or $[v_k, a]$ during Phase 3.

In total, A_j consists of P_j (worth at most $u_i(P_i) + \epsilon'$ to agent i, by Lemma 7.3.4), at most one unallocated subinterval of some e_k^1 (worth at most $u_i(P_i) + \epsilon'$ to agent i, as in Case 1), and H. Now, H is a union of the unallocated segments in E^1 and a subset of E^2. The unallocated segments in E^1 together are worth less than $u_i(P_i) + \epsilon'$ to agent i—otherwise Phase 2 (in particular, Phase 2b) would have continued—and any subset of E^2 is worth at most ϵ' to agent i, so H is worth at most $u_i(P_i) + 2\epsilon'$ to agent i. As a consequence, A_j is worth at most $3u_i(P_i) + 4\epsilon'$ to agent i.

- **Case 3:** $j \in N_2$.
  By Lemma 7.3.4, we have $u_i(\widehat{P}_j) \leq 2(u_i(P_i) + \epsilon')$. The remaining portion $A_j \setminus \widehat{P}_j$ is a subset of the union of the unallocated segments in E^1 and the segments of E^2, which is worth at most $u_i(P_i) + 2\epsilon'$ to agent i as detailed in the last paragraph of Case 2. This gives $u_i(A_j) \leq 3u_i(P_i) + 4\epsilon'$.

In summary, we have $u_i(A_j) \leq 3u_i(P_i) + 4\epsilon'$ in all cases. Now, $\epsilon' = \frac{\epsilon}{16nm}$ by definition and $u_i(P_i) \geq \frac{1}{4nm}$ by Lemma 7.3.5, which implies that $4\epsilon' \leq \epsilon u_i(P_i)$. It follows that

$$u_i(A_j) \leq 3u_i(P_i) + 4\epsilon' \leq 3u_i(P_i) + \epsilon u_i(P_i) = (3 + \epsilon)u_i(P_i) \leq (3 + \epsilon)u_i(A_i).$$

Since $i, j \in N$ were arbitrarily selected, the allocation $\mathcal{A}$ is $(3 + \epsilon)$-EF, as desired. □

7.4 Identical Valuations

In this section, we focus on the case where the valuation functions of all agents are identical. While this case is uninteresting for interval cake cutting since a fully envy-free allocation can be trivially found, it becomes highly nontrivial when graphs are involved (see, for example, the left of Figure 7.1). Indeed, a number of works on dividing edges or vertices of a graph can be interpreted as dealing with the identical-valuation setting (Wu et al., 2007; Chu et al., 2010; Caragiannis et al., 2022). Moreover, this setting captures scenarios where there is an objective measure across agents, for example, when a town wants to divide the responsibility of maintaining its streets among contractors based on the lengths or numbers of residents on the streets.

As we mentioned in Section 7.2, an α-EF allocation is not guaranteed to exist for any $\alpha < 2$, even with identical valuations and star graphs. We will show in this section that, for arbitrary graphs and any $\epsilon > 0$, it is possible to find an allocation that is $(2 + \epsilon)$-EF, which means that the approximation factor of 2 is essentially tight.

To this end, we first discuss how we can find a 4-EF allocation using a variation of the IterativeDivide algorithm that we saw in Section 7.3. This 4-EF allocation will later be used as an input to an algorithm that computes a $(2 + \epsilon)$-EF allocation. For star graphs, we also describe a simpler method for computing a 2-EF allocation.

Let us denote by u the common valuation function of the agents, and define $\max(\mathcal{A}) = \max_{i \in N} u(A_i)$ and $\min(\mathcal{A}) = \min_{i \in N} u(A_i)$ for any allocation $\mathcal{A} = (A_1, \ldots, A_n)$.

7.4.1 4-EF

In IterativeDivide, we used the threshold $\beta = 1/4$ in every call to Divide so as to allocate a share worth at least $1/4$ to some agent, which results in a $1/2$-additive-EF allocation. Even with identical valuations, each iteration of Divide is unpredictable in the sense that the recipient could receive a share worth anywhere between β and 2β. If β is chosen to be more than $1/(2n-2)$ and the first $n-1$ agents all take shares of value close to 2β, then the last agent will be left effectively empty-handed. In contrast, if β is chosen to be at most $1/(2n-2)$ and the first $n-1$ agents all take shares of value only β, then the last agent will receive a share of value at least $1/2$, which leads to an envy factor linear in n.

To resolve this problem, let us consider using an *adaptive* threshold that takes the values of the previous shares into account. If the previous agents took large shares, then the threshold β is reduced appropriately for the current agent, and vice versa. Without loss of generality, assume that for each $i \in \{1, \ldots, n-1\}$, agent i is the one who takes the first share generated by the i^{th} iteration of Divide. By choosing $\beta_i = \frac{1}{2}(\frac{2i}{2n-1} - \sum_{j=1}^{i-1} u(A_j))$ to be the threshold for the i^{th} iteration of Divide, we claim that the resulting allocation is 4-EF.[6] Along the way, we shall see that the allocation is also $(2-1/n)$-proportional.

Theorem 7.4.1. *Given an instance of graphical cake cutting consisting of n agents with identical valuations, there exists an algorithm that computes a 4-EF and $(2-\frac{1}{n})$-proportional allocation in time polynomial in n and the size of G.*

Proof. We consider the algorithm IterativeDivide (Algorithm 7.1)—without loss of generality, assume that the shares are allocated to the agents in ascending order of indices, i.e., $i^* = i$ for all $i \in \{1, \ldots, n-1\}$ in the algorithm—and substitute β_i with $\frac{1}{2}(\frac{2i}{2n-1} - \sum_{j=1}^{i-1} u(A_j))$ (instead of $1/4$). We claim that this algorithm satisfies the condition of the theorem. It is clear that the running time is polynomial in n and the size of G, so it remains to check that the envy-freeness and proportionality claims are valid. Let $(A_1, \ldots, A_n) = \text{IterativeDivide}(G, N)$, and let $\xi = \frac{1}{2n-1}$. By induction, we shall prove the following statements for $i \in \{1, \ldots, n-1\}$:

(i) $\xi \leq u(A_i) < (4 - 2^{-(i-2)})\xi$;

(ii) $(2i - 2 + 2^{-(i-1)})\xi \leq \sum_{j=1}^{i} u(A_j) < 2i\xi$.

For the base case $i = 1$, we have $\beta_1 = \xi \leq 1 = u(G)$, which by Lemma 7.2.1 means that $\xi \leq u(A_1) < 2\xi$, proving the two statements together.

For the inductive step, fix $i \in \{2, \ldots, n-1\}$, and assume that the two statements hold for $i-1$. We first check that the **if**-condition in Line 3 of IterativeDivide is satisfied. To this end, we have to check that β_i is positive and is at most the value of the remaining graph $1 - \sum_{j=1}^{i-1} u(A_j)$. It follows from (ii) for $i-1$ that $\sum_{j=1}^{i-1} u(A_j) < 2(i-1)\xi$, so the former claim is true by $\beta_i = \frac{1}{2}(2i\xi - \sum_{j=1}^{i-1} u(A_j)) > \xi > 0$ and the latter claim is true by $2\beta_i + \sum_{j=1}^{i-1} u(A_j) = 2i\xi < 1$. The call to Divide in the following line is hence valid. Therefore, by Lemma 7.2.1, we have $\beta_i \leq u(A_i) < 2\beta_i$. We now prove the two inductive statements for i.

[6] Even better, the allocation is actually $(4 - 1/2^{n-3})$-EF, as shown in the proof of Theorem 7.4.1.

- By definition, $2\beta_i = 2i\xi - \sum_{j=1}^{i-1} u(A_j)$, and by the inductive hypothesis,

$$(2(i-1) - 2 + 2^{-(i-2)})\xi \leq \sum_{j=1}^{i-1} u(A_j) < 2(i-1)\xi.$$

  Putting these together gives $2\xi < 2\beta_i \leq (4 - 2^{-(i-2)})\xi$. Thus,

$$\xi < \beta_i \leq u(A_i) < 2\beta_i \leq (4 - 2^{-(i-2)})\xi,$$

  which proves (i).

- Now, subtracting $\beta_i \leq u(A_i) < 2\beta_i$ from $2\beta_i$ gives $0 < 2\beta_i - u(A_i) \leq \beta_i$. Then,

$$2i\xi - \sum_{j=1}^{i} u(A_j) = \left(2i\xi - \sum_{j=1}^{i-1} u(A_j)\right) - u(A_i) = 2\beta_i - u(A_i);$$

  combining this with the previous inequality yields $0 < 2i\xi - \sum_{j=1}^{i} u(A_j) \leq \beta_i$. Subtracting this from $2i\xi$ gives $2i\xi - \beta_i \leq \sum_{j=1}^{i} u(A_j) < 2i\xi$. Finally, combining this with the statement from the previous bullet point that $2\beta_i \leq (4 - 2^{-(i-2)})\xi$, that is, $\beta_i \leq (2 - 2^{-(i-1)})\xi$, we get

$$2i\xi - (2 - 2^{-(i-1)})\xi \leq \sum_{j=1}^{i} u(A_j) < 2i\xi,$$

  which proves (ii).

This completes the induction.

By (ii) for $i = n-1$, we have

$$(2(n-1) - 2 + 2^{-(n-2)})\xi \leq \sum_{j=1}^{n-1} u(A_j) < 2(n-1)\xi.$$

Combining this with

$$u(A_n) = 1 - \sum_{j=1}^{n-1} u(A_j) = (2n-1)\xi - \sum_{j=1}^{n-1} u(A_j),$$

we get $\xi < u(A_n) \leq (3 - 2^{-(n-2)})\xi$. Together with (i), we see that the minimum value across all $u(A_i)$'s is at least ξ and the maximum value is at most $(4 - 2^{-(n-3)})\xi$. This shows that the allocation is $(4 - 2^{-(n-3)})$-EF, which is also 4-EF. Additionally, we have $u(A_i) \geq \xi = \frac{1}{2n-1} = \frac{1}{(2-1/n)n}$ for all i, and so the allocation is $(2 - \frac{1}{n})$-proportional. $\square$

As mentioned in Section 7.2, an α-proportional allocation is not guaranteed to exist for any $\alpha < 2 - 1/n$, so the algorithm in Theorem 7.4.1 attains the optimal proportionality approximation. In terms of envy-freeness, one may consider adjusting the values of β_i in IterativeDivide so as to obtain an allocation with a better approximation factor than 4.

While this might be possible, it seems unlikely that this approach could lead to 2-EF, given that the guarantee from Lemma 7.2.1 already has a multiplicative gap of 2. This motivates us to devise another algorithm that reduces the factor to arbitrarily close to 2.

7.4.2 $(2+\epsilon)$-EF

Let us first define a *minimum-maximum path* of an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ as a list $\mathcal{P} = (P_1, \ldots, P_d)$ (where $d \leq n$) satisfying the following conditions:

- For each $i \in \{1, \ldots, d\}$, $P_i = A_j$ for some $j \in \{1, \ldots, n\}$,
- $P_i \neq P_j$ for $1 \leq i < j \leq d$,
- For each $i \in \{1, \ldots, d-1\}$, there exists at least one point belonging to both P_i and P_{i+1}, and
- $u(P_1) = \min(\mathcal{A})$ and $u(P_d) = \max(\mathcal{A})$.

Intuitively, a minimum-maximum path is a list of shares that chains a minimum-valued one to a maximum-valued one in the underlying graph without crossing any share more than once. Such a list can be found in polynomial time: we locate a minimum-valued share and a maximum-valued share of $\mathcal{A}$, find a path through the graph that connects both shares, and identify the shares corresponding to this path. If some share A_i appears more than once, we repeatedly remove the part between the two occurrences of A_i (including one of these occurrences). Let us use MinMaxPath($\mathcal{A}$) to denote an arbitrary minimum-maximum path of $\mathcal{A}$.

We now describe the algorithm, RecursiveBalance (Algorithm 7.2), which finds a $(2+\epsilon)$-EF allocation for any given $\epsilon > 0$. We employ similar ideas as the ones used by Chu et al. (2010)—in their work, they partition the (indivisible) edges of a graph, whereas we have to account for the divisibility of the edges. Assume without loss of generality that $\epsilon \in (0, 1)$. Given an allocation $\mathcal{A}$, RecursiveBalance repeatedly replaces $\mathcal{A}$ with the allocation Balance($\mathcal{A}, \epsilon$), then terminates when $\mathcal{A}$ is $(2+\epsilon)$-EF. The algorithm Balance (Algorithm 7.3) finds a minimum-maximum path $\mathcal{P}$ of $\mathcal{A}$, and replaces the shares in $\mathcal{A}$ that appear in $\mathcal{P}$ by BalancePath($\mathcal{P}, \epsilon$). Note that the order of shares in $\mathcal{A}$ does not affect the fairness properties due to the identical valuation across all agents.

Algorithm 7.2 RecursiveBalance($\mathcal{A}, \epsilon$).

Input: Allocation $\mathcal{A} = (A_1, \ldots, A_n)$, $\epsilon \in (0, 1)$.
Output: Allocation $\mathcal{A} = (A_1, \ldots, A_n)$.

```
while max(A)/min(A) > 2 + ε do
    A ← Balance(A, ε)
end while
return A
```

Now, BalancePath (Algorithm 7.4) does the bulk of the work. This algorithm adjusts the shares in $\mathcal{P} = (P_1, \ldots, P_d)$ so that their values meet certain criteria. Let $\gamma = u(P_d)$ and

Algorithm 7.3 BALANCE($\mathcal{A}, \epsilon$).

Input: Allocation $\mathcal{A} = (A_1, \dots, A_n)$, $\epsilon \in (0, 1)$.
Output: Allocation $\mathcal{A} = (A_1, \dots, A_n)$.

```
1: 𝒫 ← MinMaxPath(𝒜)
2: 𝒜 ← (𝒜 \ 𝒫) ∪ BalancePath(𝒫, ϵ)
3: return 𝒜
```

$\widehat{P}_1 = P_1$.[7] For each i from 1 to $d-1$, the algorithm does one of the following unless it is terminated prematurely via Case 1 or Case 2.

- **Case 1: The value of $\widehat{P}_i$ is at least $\frac{\gamma}{2+\epsilon}$.**
  Set $P'_i = \widehat{P}_i$ and $P'_j = P_j$ for all $j \in \{i+1, \dots, d\}$. Terminate the algorithm by returning $(P'_1, \dots, P'_d)$.

- **Case 2: The value of $\widehat{P}_i$ is less than $\frac{\gamma}{2+\epsilon}$ and the value of $\widehat{P}_i \cup P_{i+1}$ is less than $\frac{2\gamma}{2+\epsilon}$.**
  Set $P'_i = \widehat{P}_i \cup P_{i+1}$. Set P'_{i+1} and P'_d to be the respective outputs of DIVIDE$(P_d, N, \frac{\gamma}{3}, r)$, where r is any point in P_d; note that this call to DIVIDE is valid because P_d has value γ. Set $P'_j = P_j$ for all $j \in \{i+2, \dots, d-1\}$. Terminate the algorithm by returning $(P'_1, \dots, P'_d)$.

- **Case 3: The value of $\widehat{P}_i$ is less than $\frac{\gamma}{2+\epsilon}$ and the value of $\widehat{P}_i \cup P_{i+1}$ is at least $\frac{2\gamma}{2+\epsilon}$.**
  Consider the graph $P^* = \widehat{P}_i \cup P_{i+1}$. Set P'_i and$\widehat{P}_{i+1}$ to be the respective outputs of DIVIDE$(P^*, N, \frac{\gamma}{2+\epsilon}, r)$, where r is any point belonging to both P_{i+1} and P_{i+2} (unless $i = d-1$, in which case r is any point in P_d); note that this call to DIVIDE is valid because P^* has value at least $\frac{2\gamma}{2+\epsilon}$. The choice of r ensures that, if $i < d-1$, $\widehat{P}_{i+1}$ and P_{i+2} share at least one point. Continue with the next i by incrementing it by 1.

If the algorithm still has not terminated after $i = d-1$, set $P'_d = \widehat{P}_d$ and return $(P'_1, \dots, P'_d)$.

We claim that the algorithm RECURSIVEBALANCE terminates in polynomial time if it receives a 4-EF allocation as input (provided by Theorem 7.4.1), and upon termination the algorithm returns a $(2+\epsilon)$-EF allocation.

Theorem 7.4.2. *Given an instance of graphical cake cutting consisting of n agents with identical valuations, there exists an algorithm that, for any $\epsilon > 0$, computes a $(2+\epsilon)$-EF allocation in time polynomial in n, $1/\epsilon$, and the size of G.*

To establish the proof of Theorem 7.4.2, we will work "inside-out": establish properties of BALANCEPATH (Algorithm 7.4), BALANCE (Algorithm 7.3), and RECURSIVEBALANCE (Algorithm 7.2) in this order. Throughout the proofs, we will assume that the inputs of the algorithms are *pseudo* 4-*EF*—the definition is given below.

For a parameter $\alpha \geq 1$, we say that an allocation $\mathcal{A}$ is *pseudo α-EF* if $\min(\mathcal{A}) > 0$ and $\min(\mathcal{A}') \geq \max(\mathcal{A}')/\alpha$, where $\mathcal{A}'$ is defined to be an allocation after removing one share of the

[7] We reuse P_i for P'_i and $\widehat{P}_i$ in the pseudocode for simplicity; however, we differentiate them in the main text for clarity.

Algorithm 7.4 BALANCEPATH$(\mathcal{P}, \epsilon)$.

Input: List of shares $\mathcal{P} = (P_1, \ldots, P_d)$, $\epsilon \in (0, 1)$.
Output: List of d shares $(P_1, \ldots, P_d)$.
Initialization: $\gamma \leftarrow u(P_d)$.

for $i = 1, \ldots, d-1$ **do**
  if $u(P_i) \geq \frac{\gamma}{2+\epsilon}$ **then**
    return $(P_1, \ldots, P_d)$
  end if
  $P^* \leftarrow P_i \cup P_{i+1}$
  if $u(P^*) < \frac{2\gamma}{2+\epsilon}$ **then**
    $P_i \leftarrow P^*$
    $r \leftarrow$ any vertex in P_d
    $(P_{i+1}, P_d) \leftarrow$ DIVIDE $\left(P_d, N, \frac{\gamma}{3}, r\right)$
    return $(P_1, \ldots, P_d)$
  end if
  if $i = d-1$ **then**
    $r \leftarrow$ any vertex in P_d
  else
    $r \leftarrow$ any vertex in $P_{i+1} \cap P_{i+2}$
  end if
  $(P_i, P_{i+1}) \leftarrow$ DIVIDE $\left(P^*, N, \frac{\gamma}{2+\epsilon}, r\right)$
end for
return $(P_1, \ldots, P_d)$

lowest value in $\mathcal{A}$ (if there is more than one such share, the definition is independent of which of those shares is removed). In other words, the concept of "pseudo α-EF" ignores the effect of one share with the lowest value. We apply the analogous definition to a minimum-maximum path $\mathcal{P}$. Note that an α-EF allocation is also pseudo α-EF. For any minimum-maximum path $\mathcal{P}$ of $\mathcal{A}$, $\mathcal{P}$ is pseudo α-EF if $\mathcal{A}$ is, and $\mathcal{P}$ is α-EF if and only if $\mathcal{A}$ is.

First, we establish properties satisfied by the output of BALANCEPATH.

Lemma 7.4.3. *Let $\epsilon \in (0, 1)$, and let $\mathcal{P} = (P_1, \ldots, P_d)$ be a minimum-maximum path of some allocation $\mathcal{A}$ that is pseudo 4-EF but not $(2+\epsilon)$-EF. Let $\gamma = max(\mathcal{P})$. Then, the following statements regarding $\mathcal{P}$ hold:*

- $0 < u(P_1) < \frac{\gamma}{2+\epsilon}$,
- $\frac{\gamma}{4} \leq u(P_j) \leq \gamma$ *for all* $j \in \{2, \ldots, d-1\}$, *and*
- $u(P_d) = \gamma$.

Moreover, if $\mathcal{P}' = (P'_1, \ldots, P'_d)$ is the output of BALANCEPATH$(\mathcal{P}, \epsilon)$, then at least one of the following three cases holds:

- *Case 1: There exists $i \in \{2, \ldots, d-1\}$ such that*
  - $\frac{\gamma}{2+\epsilon} \leq u(P'_j) < \frac{2\gamma}{2+\epsilon}$ *for all* $j \in \{1, \ldots, i-1\}$,
  - $\frac{\gamma}{2+\epsilon} \leq u(P'_i) < u(P_i)$, *and*
  - $u(P'_j) = u(P_j)$ *for all* $j \in \{i+1, \ldots, d\}$.

- *Case 2: There exists $i \in \{1, \dots, d-2\}$ such that*
  - $\frac{\gamma}{4} \le u(P'_j) < \frac{2\gamma}{2+\epsilon}$ *for all* $j \in \{1, \dots, i+1\} \cup \{d\}$*, and*
  - $u(P'_j) = u(P_j)$ *for all* $j \in \{i+2, \dots, d-1\}$.
- *Case 3:*
  - $\frac{\gamma}{2+\epsilon} \le u(P'_j) < \frac{2\gamma}{2+\epsilon}$ *for all* $j \in \{1, \dots, d-1\}$*, and*
  - $0 < u(P'_d) < u(P_d) = \gamma$.

Proof. First, we prove the statements regarding $\mathcal{P}$. If $\mathcal{A}$ (and hence $\mathcal{P}$) is not $(2+\epsilon)$-EF, then the smallest share, P_1, has value less than $\frac{\gamma}{2+\epsilon}$. Since $\mathcal{A}$ (and hence $\mathcal{P}$) is pseudo 4-EF, P_1 must have positive value and all other P_j's must have value at least $\frac{\gamma}{4}$ and at most γ. Finally, $u(P_d) = \max(\mathcal{P}) = \gamma$.

Next, we prove the results related to $\mathcal{P}'$. Cases 1, 2 and 3 correspond to the output of BalancePath in Lines 3, 10, and 19, respectively.[8] For each $i \in \{1, \dots, d-1\}$, if the value of $\widehat{P}_i$ is less than $\frac{\gamma}{2+\epsilon}$ and the value of $\widehat{P}_i \cup P_{i+1}$ is at least $\frac{2\gamma}{2+\epsilon}$, then $\frac{\gamma}{2+\epsilon} \le u(P'_i) < \frac{2\gamma}{2+\epsilon}$ by Divide$(P^*, N, \frac{\gamma}{2+\epsilon}, r)$ and Lemma 7.2.1. Hence,

$$u(\widehat{P}_{i+1}) = u(\widehat{P}_i \cup P_{i+1}) - u(P'_i) > 0,$$

and since $u(\widehat{P}_i) < \frac{\gamma}{2+\epsilon} \le u(P'_i)$, we have

$$u(\widehat{P}_{i+1}) = u(P_{i+1}) - (u(P'_i) - u(\widehat{P}_i)) < u(P_{i+1}). \tag{7.1}$$

Putting the two bounds on $u(\widehat{P}_{i+1})$ together, we get $0 < u(\widehat{P}_{i+1}) < u(P_{i+1}) \le \gamma$.

If there is some smallest $i \in \{2, \dots, d-1\}$ such that the value of $\widehat{P}_i$ is at least $\frac{\gamma}{2+\epsilon}$ (note that this is not possible for $i = 1$ since $u(\widehat{P}_1) = u(P_1) < \frac{\gamma}{2+\epsilon}$), then we have $\frac{\gamma}{2+\epsilon} \le u(P'_j) < \frac{2\gamma}{2+\epsilon}$ for all $j \in \{1, \dots, i-1\}$ by Lemma 7.2.1, and $\frac{\gamma}{2+\epsilon} \le u(P'_i) < u(P_i)$ by (7.1) and the definition of i (since the two early termination conditions were not triggered for $i-1$ and $P'_i = \widehat{P}_i$). The remaining shares are unchanged, i.e., $u(P'_j) = u(P_j)$ for all $j \in \{i+1, \dots, d\}$. This corresponds to Case 1.

Else, if there is some smallest $i \in \{1, \dots, d-2\}$ such that the value of $\widehat{P}_i \cup P_{i+1}$ is less than $\frac{2\gamma}{2+\epsilon}$ (note that this is not possible for $i = d-1$ since $u(\widehat{P}_{d-1} \cup P_d) \ge u(P_d) = \gamma > \frac{2\gamma}{2+\epsilon}$), then we have

$$\frac{\gamma}{4} \le \frac{\gamma}{2+\epsilon} \le u(P'_j) < \frac{2\gamma}{2+\epsilon}$$

for all $j \in \{1, \dots, i-1\}$ by Lemma 7.2.1. Now, $\frac{\gamma}{4} \le u(P_{i+1}) \le u(\widehat{P}_i \cup P_{i+1}) < \frac{2\gamma}{2+\epsilon}$, where the first inequality follows from the first paragraph of this proof. Since P'_i is set to $\widehat{P}_i \cup P_{i+1}$, we have

$$\frac{\gamma}{4} \le u(P'_i) < \frac{2\gamma}{2+\epsilon}.$$

[8] Please note the notational differences between the main text and the pseudocode. We refer to the notations in the main text.

Also, since $(P'_{i+1}, P'_d) = \textsc{Divide}(P_d, N, \frac{\gamma}{3}, r)$, it holds that

$$\frac{\gamma}{4} \leq \frac{\gamma}{3} \leq u(P'_j) \leq \frac{2\gamma}{3} < \frac{2\gamma}{2+\epsilon}$$

for $j \in \{i+1, d\}$ by Lemma 7.2.1 and the fact that $u(P_d) = \gamma$. The remaining shares are unchanged, i.e., $u(P'_j) = u(P_j)$ for all $j \in \{i+2, \ldots, d-1\}$. This corresponds to Case 2.

Finally, suppose that the value of $\widehat{P}_i$ is less than $\frac{\gamma}{2+\epsilon}$ and the value of $\widehat{P}_i \cup P_{i+1}$ is at least $\frac{2\gamma}{2+\epsilon}$ for all $i \in \{1, \ldots, d-1\}$. By Lemma 7.2.1, $\frac{\gamma}{2+\epsilon} \leq u(P'_j) < \frac{2\gamma}{2+\epsilon}$ for all $j \in \{1, \ldots, d-1\}$. Moreover, since

$$\frac{2\gamma}{2+\epsilon} \leq u(\widehat{P}_{d-1} \cup P_d) \leq u(\widehat{P}_{d-1}) + u(P_d) < \frac{\gamma}{2+\epsilon} + \gamma$$

and $\frac{\gamma}{2+\epsilon} \leq u(P'_{d-1}) < \frac{2\gamma}{2+\epsilon}$, and $u(P'_d) = u(\widehat{P}_{d-1} \cup P_d) - u(P'_{d-1})$, we have $0 < u(P'_d) < \gamma = u(P_d)$. This corresponds to Case 3. □

Having analyzed the output of BalancePath, we next establish properties satisfied by the output of Balance. For an allocation $\mathcal{A}$, let $\mathcal{I}(\mathcal{A}) = \left[\frac{\max(\mathcal{A})}{2+\epsilon}, \max(\mathcal{A})\right]$ and $\mathcal{J}(\mathcal{A}) = \left[\frac{2\max(\mathcal{A})}{2+\epsilon}, \max(\mathcal{A})\right]$. Let $\mathcal{N}(\mathcal{A}, I)$ be the number of shares in $\mathcal{A}$ having values in the interval I, and let $\mathcal{N}(\mathcal{A}) = \mathcal{N}(\mathcal{A}, \mathcal{I}(\mathcal{A}))$.

Lemma 7.4.4. *Let $\epsilon \in (0,1)$, and let $\mathcal{A} = (A_1, \ldots, A_n)$ be an allocation that is pseudo 4-EF but not $(2+\epsilon)$-EF. Let $\mathcal{A}' = (A'_1, \ldots, A'_n)$ be the output of Balance$(\mathcal{A}, \epsilon)$. Without loss of generality, assume that both $\mathcal{A}$ and $\mathcal{A}'$ have the shares arranged in ascending order of values. Then $\mathcal{A}'$ is pseudo 4-EF, and at least one of the following two cases holds:*

- *Case (i): $\mathcal{N}(\mathcal{A}', \mathcal{I}(\mathcal{A})) > \mathcal{N}(\mathcal{A})$, and for all j such that $u(A'_j) \in \mathcal{J}(\mathcal{A})$, we have $u(A'_j) \leq u(A_j)$.*
- *Case (ii): For all j such that $u(A'_j) \in \mathcal{J}(\mathcal{A})$, we have $u(A'_j) \leq u(A_{j-1})$.*

Proof. Since $\mathcal{A}$ is pseudo 4-EF but not $(2+\epsilon)$-EF, we can apply Lemma 7.4.3 on its minimum-maximum path denoted by $\mathcal{P}$. Note that the values of the shares in $\mathcal{A} \setminus \mathcal{P}$ remain unchanged, while all shares in $\mathcal{P}'$ have values in the range $[\frac{\gamma}{4}, \gamma]$, except possibly P'_d which has value in the range $(0, \gamma]$. This means that $\mathcal{A}'$ is pseudo 4-EF.

We now show that at least one of the two cases in the lemma statement holds. First, suppose that either $\mathcal{P}'$ falls under Case 1 of Lemma 7.4.3, or $\mathcal{P}'$ falls under Case 3 with the additional condition that $\frac{\gamma}{2+\epsilon} \leq u(P'_d) < \gamma$. There exists $i \in \{2, \ldots, d\}$ such that only $P_1, \ldots, P_i$ are changed to $P'_1, \ldots, P'_i$, while the rest of the shares remain unchanged. Let $\widehat{\mathcal{P}} = (P_1, \ldots, P_i)$ and $\widehat{\mathcal{P}}' = (P'_1, \ldots, P'_i)$. We have $u(P_1) \notin \mathcal{I}(\mathcal{A})$ because $\mathcal{A}$ is not $(2+\epsilon)$-EF, so $\mathcal{N}(\widehat{\mathcal{P}}, \mathcal{I}(\mathcal{A})) \leq i-1$, while $u(P'_j) \in \mathcal{I}(\mathcal{A})$ for all $j \in \{1, \ldots, i\}$, so $\mathcal{N}(\widehat{\mathcal{P}}', \mathcal{I}(\mathcal{A})) = i$. As the rest of the shares in $\mathcal{P}$ and $\mathcal{A}$ remain unchanged, this implies $\mathcal{N}(\mathcal{A}', \mathcal{I}(\mathcal{A})) > \mathcal{N}(\mathcal{A})$. Furthermore, the only share in $\widehat{\mathcal{P}}'$ with a value that is potentially in $\mathcal{J}(\mathcal{A})$ is P'_i, but since $u(P'_i) < u(P_i)$, we have $u(A'_j) \leq u(A_j)$ for all j such that $u(A'_j) \in \mathcal{J}(\mathcal{A})$. This corresponds to Case (i).

Next, suppose that either $\mathcal{P}'$ falls under Case 2, or $\mathcal{P}'$ falls under Case 3 with the additional condition that $0 < u(P'_d) < \frac{\gamma}{2+\epsilon}$. There exists $i \in \{1, \dots, d-2\}$ such that only $P_1, \dots, P_{i+1}$ and P_d are changed to $P'_1, \dots, P'_{i+1}$ and P'_d. Then we have $u(P'_j) \notin \mathcal{J}(\mathcal{A})$ for all $j \in \{1, \dots, i+1\} \cup \{d\}$. Hence, for j such that $u(P'_j) \in \mathcal{J}(\mathcal{A})$, it holds that $j \in \{i+2, \dots, d-1\}$ and $u(P_j) = u(P'_j) \in \mathcal{J}(\mathcal{A})$. Moreover, $u(P_d) = \max(\mathcal{A}) \in \mathcal{J}(\mathcal{A})$. Since both $\mathcal{A}$ and $\mathcal{A}'$ have the shares arranged in ascending order of values and the values of the shares in $\mathcal{A} \setminus \mathcal{P}$ remain unchanged, it follows that $u(A'_j) \le u(A_{j-1})$ for all j such that $u(A'_j) \in \mathcal{J}(\mathcal{A})$. This corresponds to Case (ii). □

By leveraging Lemma 7.4.4, we can bound the number of calls to Balance in RecursiveBalance.

Lemma 7.4.5. *Given an instance of graphical cake cutting consisting of n agents with identical valuations, a 4-EF allocation $\mathcal{A}$ of the cake, and any $\epsilon \in (0,1)$, the algorithm RecursiveBalance terminates after at most $\frac{5n^2}{\epsilon}$ calls to Balance.*

Proof. Let $\mathcal{A}^0 = \mathcal{A}$, and $\mathcal{A}^{t+1} = \textsc{Balance}(\mathcal{A}^t, \epsilon)$ for all $t \ge 0$. If there is some t such that $\mathcal{A}^t$ is $(2+\epsilon)$-EF, then the algorithm RecursiveBalance terminates. By Lemma 7.4.4, every $\mathcal{A}^t$ is pseudo 4-EF, and if $\mathcal{A}^t$ is not $(2+\epsilon)$-EF, then either Case (i) or Case (ii) holds when $\mathcal{A}^t$ is used as an input to Balance, with output $\mathcal{A}^{t+1}$. As a consequence of Lemma 7.4.4, we have $\max(\mathcal{A}^{t+1}) \le \max(\mathcal{A}^t)$. For an allocation $\mathcal{B}$, let us say that $\mathcal{B}$ *follows Case (i)* (resp., *Case (ii)*) if the pair $(\mathcal{B}, \mathcal{B}')$ fulfills the conditions of Case (i) (resp., Case (ii)), where $\mathcal{B}' = \textsc{Balance}(\mathcal{B}, \epsilon)$.

We claim that RecursiveBalance terminates if there are n consecutive allocations following Case (i). Consider some $\mathcal{A}^t$ and assume that for every $k \in \{0, \dots, n-1\}$, $\mathcal{A}^{t+k}$ follows Case (i) and is not $(2+\epsilon)$-EF. For each k, we have

$$\mathcal{N}(\mathcal{A}^{t+k+1}, \mathcal{I}(\mathcal{A}^{t+k})) > \mathcal{N}(\mathcal{A}^{t+k})$$

by Lemma 7.4.4, and

$$\mathcal{N}(\mathcal{A}^{t+k+1}) \ge \mathcal{N}(\mathcal{A}^{t+k+1}, \mathcal{I}(\mathcal{A}^{t+k}))$$

holds due to the fact that $\max(\mathcal{A}^{t+k+1}) \le \max(\mathcal{A}^{t+k})$. This gives the chain

$$\mathcal{N}(\mathcal{A}^t) < \mathcal{N}(\mathcal{A}^{t+1}) < \cdots < \mathcal{N}(\mathcal{A}^{t+n}).$$

Since each $\mathcal{N}(\mathcal{A}^{t+k})$ is an integer bounded by 0 and n, the chain forces $\mathcal{N}(\mathcal{A}^{t+k}) = k$ for all $k \in \{0, \dots, n\}$. In particular, $\mathcal{N}(\mathcal{A}^{t+n}) = n$, which implies that $\mathcal{A}^{t+n}$ is $(2+\epsilon)$-EF, and the algorithm terminates.

Let $\mathcal{A}^t$ and $k > 0$ be given. For each $i \in \{0, \dots, k\}$, consider the number of shares the allocation $\mathcal{A}^{t+i}$ has in the interval $\mathcal{J}(\mathcal{A}^t)$, i.e., consider $z_i = \mathcal{N}(\mathcal{A}^{t+i}, \mathcal{J}(\mathcal{A}^t))$. By Lemma 7.4.4, if $\mathcal{A}^{t+i}$ follows Case (i), then for all j such that $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^{t+i})$, we have $u(A_j^{t+i+1}) \le u(A_j^{t+i})$. Note that if $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^t)$, then it must also be that $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^{t+i})$, because $\max(\mathcal{A}^t) \ge \max(\mathcal{A}^{t+i}) \ge \max(\mathcal{A}^{t+i+1})$. Hence, if $\mathcal{A}^{t+i}$ follows Case (i), then for all j such that $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^t)$, we have $u(A_j^{t+i+1}) \le u(A_j^{t+i})$. This implies that

$z_{i+1} \le z_i$ in this case. Now, consider the case where $z_i > 0$ and $\mathcal{A}^{t+i}$ follows Case (ii). This means that for all j such that $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^{t+i})$, we have $u(A_j^{t+i+1}) \le u(A_{j-1}^{t+i})$. Note again that if $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^t)$, then it must also be that $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^{t+i})$, because $\max(\mathcal{A}^t) \ge \max(\mathcal{A}^{t+i}) \ge \max(\mathcal{A}^{t+i+1})$. Therefore, for all j such that $u(A_j^{t+i+1}) \in \mathcal{J}(\mathcal{A}^t)$, we have $u(A_j^{t+i+1}) \le u(A_{j-1}^{t+i})$. Combined with the assumption that $z_i > 0$, it follows that $z_{i+1} < z_i$ in this case. Hence, if at least n allocations among $\mathcal{A}^t, \mathcal{A}^{t+1}, \ldots, \mathcal{A}^{t+k-1}$ follow Case (ii), then $z_k = 0$ and consequently, $\max(\mathcal{A}^{t+k}) \le \frac{2}{2+\epsilon}\max(\mathcal{A}^t)$. Similarly, if at least qn allocations among $\mathcal{A}^t, \mathcal{A}^{t+1}, \ldots, \mathcal{A}^{t+k-1}$ follow Case (ii) for some positive integer q, then $\max(\mathcal{A}^{t+k}) \le \left(\frac{2}{2+\epsilon}\right)^q \max(\mathcal{A}^t)$.

We are now ready to show that RecursiveBalance terminates after at most qn^2 calls to Balance, where $q = \lfloor 5/\epsilon \rfloor$. Suppose on the contrary that the algorithm still has not terminated at $\mathcal{A}^{qn^2}$. If fewer than qn allocations among $\mathcal{A}^0, \mathcal{A}^1, \ldots, \mathcal{A}^{qn^2-1}$ follow Case (ii), then by a counting argument, there must be n consecutive allocations in this sequence that follow Case (i), and hence the algorithm would have terminated. Therefore, at least qn allocations among $\mathcal{A}^0, \mathcal{A}^1, \ldots, \mathcal{A}^{qn^2-1}$ follow Case (ii), so by the previous paragraph, $\max(\mathcal{A}^{qn^2}) \le \left(\frac{2}{2+\epsilon}\right)^q \max(\mathcal{A}^0)$. Since $\mathcal{A}^0$ is 4-EF, we must have $\max(\mathcal{A}^0) \le \frac{4}{n+3}$ (otherwise, every share has value more than $\frac{1}{n+3}$ and the total value of all n shares is more than $\frac{4}{n+3} + (n-1)\cdot\frac{1}{n+3} = 1$, a contradiction). Now, the function $f(x) = (1+x)^{1/x}$ is decreasing in $(0,\infty)$—to see this, note that $\ln f(x) = \frac{1}{x}\ln(1+x) = \frac{\ln(1+x)-\ln 1}{x}$, which is the slope of the line through $(1, \ln 1)$ and $(1+x, \ln(1+x))$, and must be decreasing because the ln function is concave. Hence, $(1+\frac{\epsilon}{2})^{2/\epsilon} = f(\epsilon/2) \ge f(1) = 2$, and so

$$\left(\frac{2+\epsilon}{2}\right)^q = \left(1+\frac{\epsilon}{2}\right)^q \ge \left(1+\frac{\epsilon}{2}\right)^{4/\epsilon} \ge 2^2 = 4,$$

which implies that $\left(\frac{2}{2+\epsilon}\right)^q \le \frac{1}{4}$. It follows that

$$\max(\mathcal{A}^{qn^2}) \le \left(\frac{1}{4}\right)\left(\frac{4}{n+3}\right) = \frac{1}{n+3} \le \frac{2}{2n-1}.$$

However, this means that $\mathcal{A}^{qn^2}$ is 2-EF (otherwise, some share has value less than $\frac{1}{2n-1}$ and the total value of all n shares is less than $\frac{1}{2n-1} + (n-1)\cdot\frac{2}{2n-1} = 1$, a contradiction) and hence $(2+\epsilon)$-EF. This is a contradiction since the algorithm should have terminated at $\mathcal{A}^{qn^2}$. $\square$

We now use Lemma 7.4.5 to establish Theorem 7.4.2.

Proof of Theorem 7.4.2. Without loss of generality, we may assume that $\epsilon \in (0,1)$. Apply IterativeDivide to obtain a 4-EF allocation (Theorem 7.4.1), then apply Recursive-Balance on this allocation to obtain a $(2+\epsilon)$-EF allocation. IterativeDivide and each iteration of Balance run in time polynomial in n, $1/\epsilon$, and the size of G, and the number of iterations of Balance is polynomial in n and $1/\epsilon$ by Lemma 7.4.5. The claimed overall running time follows. $\square$

7.4.3 Star Graphs

Although RecursiveBalance provides a nearly 2-EF allocation, the algorithm is rather complex and involves many steps. Moreover, it seems difficult to improve the guarantee to *exactly* 2-EF via the algorithm, as the balancing of shares may be too slow and we may not reach a 2-EF allocation in finite time.[9] We show next that, for the case of star graphs, it is possible to obtain a 2-EF allocation using a simpler "bag-filling" approach.

Let G be a star graph centered at vertex v. While there exists an edge $[u, v]$ in G with value at least $1/n$, find a point $x \in [u, v]$ such that $[u, x]$ has value $1/n$, and allocate $[u, x]$ to one of the agents. Remove this agent from consideration and remove the allocated segment from G; note that G remains a star graph. Repeat this process on the remaining graph and agents until every edge has value less than $1/n$. See Figure 7.3 for an illustration.

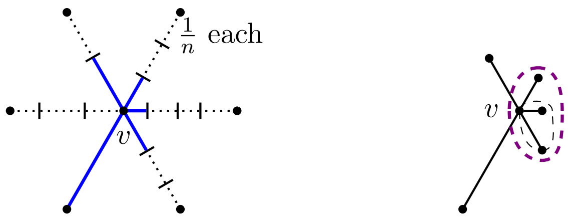


Figure 7.3: (Left) Segments of value $1/n$ each (dotted lines) are allocated to the agents. The remaining stubs (solid lines) have value less than $1/n$ each. (Right) Two groups of the lowest value are merged together repeatedly. The figure shows three final groups: one group of three stubs on the right and two groups of one stub each on the left.

At this point, we are left with a star of "stubs"—edges with value less than $1/n$ each—and $k \in \{0, \ldots, n\}$ agents who are not allocated any share yet. If $k = 0$ we are done,[10] so assume that $k \geq 1$. The total value of all stubs is exactly k/n, so there are more than k stubs. Make each stub a separate "group", then repeatedly merge two groups of the lowest value until there are exactly k groups. Assign these k groups to the k agents; note that each group is connected by the vertex v. We claim that the resulting allocation is 2-EF.

Theorem 7.4.6. *Given an instance of graphical cake cutting consisting of a star graph with m edges and n agents with identical valuations, there exists an algorithm that computes a 2-EF allocation in time polynomial in n and m.*

Proof. The running time claim is clear.

If no agent remains after assigning segments of value $1/n$ (i.e., $k = 0$), then the allocation is envy-free and hence 2-EF.

Assume that $k \geq 1$. The total value of the star of stubs is k/n, so we cannot end the process with k groups of value less than $1/n$ each. Therefore, at some point, a newly-merged group has value at least $1/n$. Two groups of the lowest value are always the ones selected

[9] For example, agents with shares of 1 unit and $2+\epsilon$ units respectively may repeatedly exchange and adjust their shares with each other (through multiple calls to Algorithm 7.4) but may never reach a multiplicative envy factor of 2.

[10] There may be a star of stubs of value 0 left; in that case we append it to the last agent's share so that connectivity is retained.

for the merge, so the value of each newly-merged group is at least that of any previously-merged group. Let g^* be the final group formed by merging two (now extinct) groups g_1 and g_2, and assume without loss of generality that $u(g_1) \leq u(g_2)$. Not only does g^* have the maximum value across all *groups*, but the fact that $u(g^*) \geq 1/n$ implies that it also has the maximum value across all *shares* (including the $1/n$-value segments from the initial process). Furthermore, it follows that $u(g^*) \leq 2/n$; otherwise, we would have $u(g_2) > 1/n$, in which case each of the k groups would have value greater than $1/n$ and the sum of their values could not be exactly k/n.

We shall prove that the allocation is 2-EF. Since g^* has the maximum value across all shares, it suffices to prove that every agent receives a share worth at least $u(g^*)/2$. Let agent i be given. If agent i receives some group g, then g must have value at least $u(g_2)$ (otherwise g would have been chosen for the final merge with g_1 instead of g_2). This means that $u(g) \geq (u(g_1) + u(g_2))/2 = u(g^*)/2$. Else, agent i receives a $1/n$-value segment; then the segment has value at least $u(g^*)/2$ since $u(g^*) \leq 2/n$. It follows that the allocation is 2-EF, as desired. □

7.5 Beyond One Connected Piece

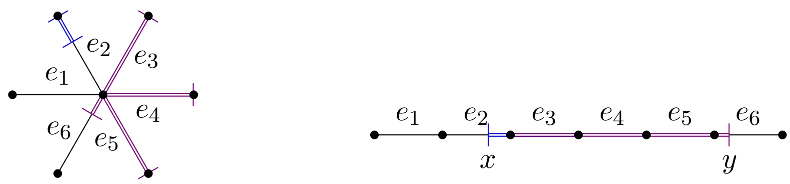
In previous sections, we made the crucial assumption that each agent necessarily receives a connected piece of the graphical cake. We now relax this assumption and explore improvements in envy-freeness guarantees if each agent is allowed to receive a small number of connected pieces. As mentioned in Section 7.1.2, extensive research has been done on (approximate) envy-free connected allocations of an *interval cake*; it will be interesting to draw connections between these results and graphical cake cutting.

As an example, consider a star graph with edges e_k for $k \in \{1, \dots, m\}$. Rearrange the edges to form a path graph with the edges e_1 to e_m going from left to right; the farthest end of each edge from the center vertex in the star graph is oriented towards the *right* on the path graph. Note that any segment along the path graph corresponds to at most two connected pieces in the star graph; see Figure 7.4 for an illustration.[11] It follows that if a connected α-EF (resp., α-additive-EF) allocation of an interval cake can be found in polynomial time, then there also exists a polynomial-time algorithm that finds an α-EF (resp., α-additive-EF) allocation of the star cake where each agent receives at most *two* connected pieces.

We now consider an arbitrary graph G. Let ϕ be a bijection of the edges of G onto the edges of a path graph with the same number of edges as G, where the orientation of the edges along the path can be chosen. By an abuse of notation, let $\phi(G)$ be the corresponding path graph. Define the *path similarity number of G associated with ϕ*, denoted by $\mathrm{PSN}(G, \phi)$, as the smallest number k such that any segment along $\phi(G)$ corresponds to at most k connected pieces in G. Any results pertaining to connected allocations of an interval cake[12] can be directly applied to allocations of G in which each agent receives at most $\mathrm{PSN}(G, \phi)$ connected

[11] A segment can start and end at points within an edge and span across different edges—it need not start and end at vertices.

[12] For instance, Barman and Kulkarni (2023) presented a polynomial-time algorithm that computes a (roughly) 1/4-additive-EF and 2-EF allocation of an interval cake.



(a) Star graph (b) Path graph

Figure 7.4: (a) A star graph with six edges and (b) its corresponding path graph. A segment $[x, y]$ along the path graph (double lines) corresponds to at most two connected pieces in the star graph.

pieces.

Proposition 7.5.1. *Let G be a graph and ϕ be a bijection of the edges of G onto $\phi(G)$ with orientation such that ϕ can be found in polynomial time. If there exists a polynomial-time algorithm that computes a connected α-EF allocation of an interval cake, then there exists a polynomial-time algorithm that computes an α-EF allocation of a graphical cake G where each agent receives at most $PSN(G, \phi)$ connected pieces. An analogous statement holds for α-additive-EF.*

To complement Proposition 7.5.1, we provide upper bounds of $\mathrm{PSN}(G, \phi)$ where ϕ can be computed in polynomial time.

Theorem 7.5.2. *Let G be a tree of height h. Then there exists a bijection ϕ that can be computed in polynomial time such that $PSN(G, \phi) \leq h + 1$.*

Proof. Define $\phi(G)$ as follows: arrange the edges from left to right according to their appearance in a depth-first search of G, and orient their directions so that the farthest point of each edge from the root vertex in G appears towards the *right* of the edge in $\phi(G)$. Clearly, ϕ can be computed in polynomial time. We claim that any segment along $\phi(G)$ corresponds to at most $h + 1$ connected pieces in G.

Let $[x, y]$ be a segment along $\phi(G)$, and consider its corresponding piece(s) in G—see Figure 7.5 for an illustration. If $[x, y]$ corresponds to a path in G, then this path is a connected piece in G. Otherwise, as we traverse $[x, y]$ from left to right in $\phi(G)$, the corresponding traversal in G will eventually return to some ancestor vertex before another branch in G is searched; this process may be repeated. Therefore, there exists some point in $[x, y]$ that corresponds to a vertex v in G such that $[x, y]$ corresponds to a subgraph of the subtree rooted at v (for example, in Figure 7.5, v is the root of the tree G). Let $z_1 \in [x, y]$ be the leftmost point that corresponds to vertex v (in Figure 7.5, only one point in $[x, y]$ corresponds to v). Note that $[z_1, y]$ corresponds to a connected piece in G—this is because z_1 corresponds to the root vertex of the subtree and a depth-first search from z_1 to y ensures that the subgraph is connected.

By repeating this process on $[x, z_i]$, we can find a sequence of points $z_1, \ldots, z_k$ in $[x, y]$ such that $[x, z_k], [z_k, z_{k-1}], \ldots, [z_2, z_1], [z_1, y]$ each corresponds to a connected piece in G—note that there are at most $k + 1$ connected pieces. Furthermore, $z_1, \ldots, z_k$ correspond to a chain of vertices with an ancestor-descendant relationship, i.e., z_i is an ancestor of z_j for

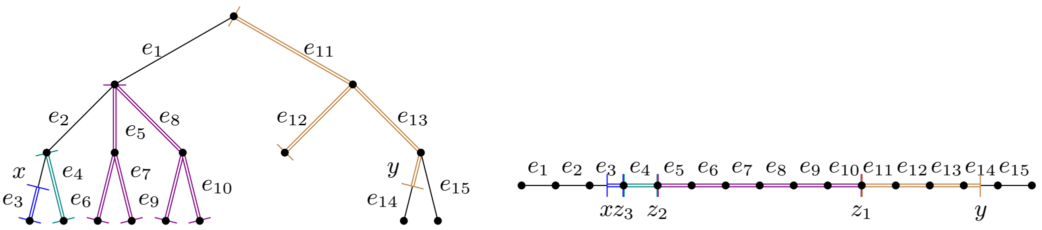


Figure 7.5: (a) A tree G of height 3 and (b) its corresponding path graph $\phi(G)$ based on a depth-first search of G. A segment $[x, y]$ along the path graph (double lines) corresponds to at most *four* connected pieces in the tree.

$i < j$ in the corresponding graph G. Since none of the z_i's correspond to leaf vertices, the chain has at most h vertices, i.e., $k \leq h$. Therefore, there are at most $h + 1$ connected pieces in G. □

Theorem 7.5.3. *Let G be a connected graph, and let d be the diameter of a spanning tree of G with the minimum diameter. Then there exists a bijection ϕ that can be computed in polynomial time such that $PSN(G, \phi) \leq \lceil d/2 \rceil + 2$.*

Proof. A spanning tree T of G with the minimum diameter, and a vertex r that minimizes its maximum distance to any vertex in T, can be found in polynomial time (Hassin and Tamir, 1995).[13] Let T be rooted at r. We claim that the height of T is at most $\lceil d/2 \rceil$. To see this, let $v_1, \dots, v_k$ be the children of r, and assume for contradiction that some node v is of distance greater than $\lceil d/2 \rceil$ from r; without loss of generality, suppose v is a descendant of v_1. If there exists a descendant w of some node in $\{v_2, \dots, v_k\}$ of distance at least $\lceil d/2 \rceil$ from r, then the path from v to w must pass through r since T is a tree, and the distance between v and w is greater than $\lceil d/2 \rceil + \lceil d/2 \rceil \geq d$, contradicting the fact that d is the diameter of T. Therefore, every descendant of $v_2, \dots, v_k$ must be of distance less than $\lceil d/2 \rceil$ from r. It follows that r is not a node that minimizes its maximum distance to any vertex in T, because v_1 reduces this quantity by one. This contradicts the definition of r. Hence, the height of T is at most $\lceil d/2 \rceil$.

Now, construct the tree T' as follows: start with all vertices and edges in T and with root vertex r; then for each edge $e = [w, v]$ in $G \setminus T$, add a new leaf vertex v' as a child of w in T', where the edge $[w, v']$ in T' corresponds to the edge e in G. If v appears multiple times across different edges in $G \setminus T$, then a *new* leaf vertex is created each time. The constructed graph T' is a tree with corresponding edges in G, and has height at most one more than the height of T; furthermore, any connected piece in T' corresponds to a connected piece in G. Let ϕ' be the corresponding bijection of the edges from G to T'. By Theorem 7.5.2 applied to T' with height $h \leq \lceil d/2 \rceil + 1$, there exists a bijection ϕ'' that can be computed in polynomial time such that $\mathrm{PSN}(T', \phi'') \leq \lceil d/2 \rceil + 2$. Then, we have $\mathrm{PSN}(G, \phi'' \circ \phi') \leq \mathrm{PSN}(T', \phi'') \leq \lceil d/2 \rceil + 2$, as desired. □

[13] r is known as an *absolute* 1-*center of* G (Hassin and Tamir, 1995).

7.6 Conclusion

In this chapter, we have studied the existence and computation of approximately envy-free allocations in graphical cake cutting. For general graphs, we devised polynomial-time algorithms for computing a $1/2$-additive-envy-free allocation in the case of non-identical valuations and a $(2+\epsilon)$-envy-free allocation in the case of identical valuations. For star graphs, our efficient algorithms provide a multiplicative envy factor of $3+\epsilon$ for non-identical valuations and 2 for identical valuations. Our bounds in the case of identical valuations are (essentially) tight. We also explored envy-freeness guarantees when the connectivity assumption is relaxed, through the notion of path similarity number.

An interesting question left open by this chapter is whether a connected allocation with a constant multiplicative approximation of envy-freeness can be guaranteed for general graphs and non-identical valuations—the techniques that we developed for star graphs (Section 7.3.2) do not seem sufficient for answering this question. If a non-connected allocation is allowed, then tightening the bounds for the path similarity number (Section 7.5) will reduce the number of connected pieces each agent receives to achieve the same envy-freeness approximation. Another intriguing direction for non-connected allocations is whether we can obtain improved envy bounds using a different approach than converting the graph into a path.

Bibliography

Rediet Abebe, Jon Kleinberg, and David C. Parkes. Fair division via social comparison. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 281–289, 2017.

Hannaneh Akrami and Jugal Garg. Breaking the 3/4 barrier for approximate maximin share. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 74–91, 2024.

Hannaneh Akrami, Noga Alon, Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, and Ruta Mehta. EFX: A simpler approach and an (almost) optimal guarantee via rainbow cycle number. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, page 61, 2023.

Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence*, 322:103965, 2023a.

Georgios Amanatidis, Georgios Birmpas, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. Round-robin beyond additive agents: Existence and fairness of approximate equilibria. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, pages 67–87, 2023b.

Eshwar Ram Arunachaleswaran, Siddharth Barman, Rachitesh Kumar, and Nidhi Rathi. Fair and efficient cake division with connected pieces. In *Proceedings of the 15th Conference on Web and Internet Economics (WINE)*, pages 57–70, 2019. Extended version available at *Computing Research Repository (CoRR)*, abs/1907.11019v4.

Yonatan Aumann and Yair Dombb. The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation*, 3(4):23:1–23:16, 2015.

Yonatan Aumann, Yair Dombb, and Avinatan Hassidim. Computing socially-efficient cake divisions. *Computing Research Repository (CoRR)*, abs/1205.3982, 2012.

Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–427, 2016.

Haris Aziz, Ildikó Schlotter, and Toby Walsh. Control of fair division. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 67–73, 2016.

Haris Aziz, Péter Biró, Jérôme Lang, Julien Lesca, and Jérôme Monnot. Efficient reallocation under additive and responsive preferences. *Theoretical Computer Science*, 790:1–15, 2019.

Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi-Agent Systems*, 36(1):3:1–3:21, 2022.

Haris Aziz, Warut Suksompong, Zhaohong Sun, and Toby Walsh. Fairness concepts for indivisible items with externalities. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5472–5480, 2023.

Julius B. Barbanel. Game-theoretic algorithms for fair and strongly fair cake division with entitlements. *Colloquium Mathematicae*, 69(1):59–73, 1996a.

Julius B. Barbanel. Super envy-free cake division and independence of measures. *Journal of Mathematical Analysis and Applications*, 197(1):54–60, 1996b.

Julius B. Barbanel, Steven J. Brams, and Walter Stromquist. Cutting a pie is not a piece of cake. *American Mathematical Monthly*, 116(6):496–514, 2009.

Siddharth Barman and Pooja Kulkarni. Approximation algorithms for envy-free cake division with connected pieces. In *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 16:1–16:19, 2023.

Xiaohui Bei and Warut Suksompong. Dividing a graphical cake. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5159–5166, 2021.

Xiaohui Bei, Ning Chen, Xia Hua, Biaoshuai Tao, and Endong Yang. Optimal proportional cake cutting with connected pieces. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1263–1269, 2012.

Xiaohui Bei, Youming Qiao, and Shengyu Zhang. Networked fairness in cake cutting. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3632–3638, 2017.

Xiaohui Bei, Xiaoming Sun, Hao Wu, Jialin Zhang, Zhijie Zhang, and Wei Zi. Cake cutting on graphs: A discrete and bounded proportional protocol. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2114–2123, 2020.

Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. Fair division of mixed divisible and indivisible goods. *Artificial Intelligence*, 293:103436, 2021.

Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. The price of connectivity in fair division. *SIAM Journal on Discrete Mathematics*, 36(2):1156–1186, 2022.

Xiaohui Bei, Alexander Lam, Xinhang Lu, and Warut Suksompong. Welfare loss in connected resource allocation. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2660–2668, 2024.

Benjamin Aram Berendsohn, Simona Boyadzhiyska, and László Kozma. Fixed-point cycles and approximate EFX allocations. In *Proceedings of the 47th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 17:1–17:13, 2022.

Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. On approximate envy-freeness for indivisible chores and mixed resources. In *Proceedings of the 24th International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 1:1–1:23, 2021.

Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S. Zwicker. Almost envy-free allocations with connected bundles. *Games and Economic Behavior*, 131:197–221, 2022.

Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 91–97, 2018.

Niclas Boehmer, Robert Bredereck, Klaus Heeger, Dušan Knop, and Junjie Luo. Multivariate algorithmics for eliminating envy by donating goods. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 127–135, 2022.

Marthe Bonamy, Nicolas Bousquet, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Arnaud Mary, Moritz Mühlenthaler, and Kunihiro Wasa. The perfect matching reconfiguration problem. In *Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 80:1–80:14, 2019.

Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 135–141, 2017.

Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

Steven J. Brams, Michael A. Jones, and Christian Klamler. Proportional pie-cutting. *International Journal of Game Theory*, 36:353–367, 2008.

Simina Brânzei, Ariel D. Procaccia, and Jie Zhang. Externalities in cake cutting. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 55–61, 2013.

Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

Eric Budish, Gérard P. Cachon, Judd B. Kessler, and Abraham Othman. Course Match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, 65(2):314–336, 2017.

Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation*, 7(3):12:1–12:32, 2019.

Ioannis Caragiannis, Evi Micha, and Nisarg Shah. A little charity guarantees fair connected graph partitioning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 4908–4916, 2022.

Katarína Cechlárová and Eva Pillárová. On the computability of equitable divisions. *Discrete Optimization*, 9(4):249–257, 2012.

Katarína Cechlárová, Jozef Doboš, and Eva Pillárová. On the existence of equitable cake divisions. *Information Sciences*, 228:239–245, 2013.

Guillaume Chèze. Envy-free cake cutting: A polynomial number of queries with high probability. *Computing Research Repository (CoRR)*, abs/2005.01982, 2020.

An-Chiang Chu, Bang Ye Wu, Hung-Lung Wang, and Kun-Mao Chao. A tight bound on the min-ratio edge-partitioning problem of a tree. *Discrete Applied Mathematics*, 158(14): 1471–1478, 2010.

An-Chiang Chu, Bang Ye Wu, and Kun-Mao Chao. A linear-time algorithm for finding an edge-partition with max-min ratio at most two. *Discrete Applied Mathematics*, 161(7–8): 932–943, 2013.

Logan Crew, Bhargav Narayanan, and Sophie Spirkl. Disproportionate division. *Bulletin of the London Mathematical Society*, 52(5):885–890, 2020.

Ágnes Cseh and Tamás Fleiner. The complexity of cake cutting with unequal shares. *ACM Transactions on Algorithms (TALG)*, 16(3):1–21, 2020.

Argyrios Deligkas, Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. The parameterized complexity of connected fair division. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 139–145, 2021.

Argyrios Deligkas, Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. The complexity of envy-free graph cutting. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 237–243, 2022.

Britta Dorn, Ronald de Haan, and Ildikó Schlotter. Obtaining a proportional allocation by deleting items. *Algorithmica*, 83(5):1559–1603, 2021.

Lester E. Dubins and Edwin H. Spanier. How to cut a cake fairly. *American Mathematical Monthly*, 68(1):1–17, 1961.

Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. Graphical cake cutting via maximin share. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 161–167, 2021.

Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. Mind the gap: Cake cutting with separation. *Artificial Intelligence*, 313:103783, 2022.

Shimon Even and Azaria Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3): 285–296, 1984.

Rupert Freeman, Evi Micha, and Nisarg Shah. Two-sided matching meets fair division. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 203–209, 2021.

Harmender Gahlawat and Meirav Zehavi. Parameterized complexity of incomplete connected fair division. In *Proceedings of the 43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 14:1–14:18, 2023.

Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

Ganesh Ghalme, Xin Huang, Yuka Machino, and Nidhi Rathi. A discrete and bounded locally envy-free cake cutting protocol on trees. In *Proceedings of the 19th International Conference on Web and Internet Economics (WINE)*, pages 310–328, 2023.

Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Beyond additive valuations. *Artificial Intelligence*, 303:103633, 2022.

Paul W. Goldberg, Alexandros Hollender, and Warut Suksompong. Contiguous cake cutting: Hardness results and approximation algorithms. *Journal of Artificial Intelligence Research*, 69:109–141, 2020.

Jonathan Goldman and Ariel D. Procaccia. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges*, 13(2):41–46, 2014.

Laurent Gourvès, Julien Lesca, and Anaëlle Wilczynski. Object allocation via swaps along a social network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 213–219, 2017.

Hao Guo, Weidong Li, and Bin Deng. A survey on fair allocation of chores. *Mathematics*, 11 (16), 2023.

Jiatong Han and Warut Suksompong. Fast & Fair: A collaborative platform for fair division applications. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 23796–23798, 2024.

Refael Hassin and Arie Tamir. On the minimum diameter spanning tree problem. *Information Processing Letters*, 53(2):109–111, 1995.

Ian Holyer. The NP-completeness of some edge-partition problems. *SIAM Journal on Computing*, 10(4):713–717, 1981.

Sen Huang and Mingyu Xiao. Object reachability via swaps under strict and weak preferences. *Autonomous Agents and Multi-Agent Systems*, 34(2):51:1–51:33, 2020.

Halvard Hummel and Magnus Lie Hetland. Maximin shares under cardinality constraints. In *Proceedings of the 19th European Conference on Multi-Agent Systems (EUMAS)*, pages 188–206, 2022.

Ayumi Igarashi and Dominik Peters. Pareto-optimal allocation of indivisible goods with connectivity constraints. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 2045–2052, 2019.

Ayumi Igarashi and Tomohiko Yokoyama. Kajibuntan: A house chore division app. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, pages 16449–16451, 2023.

Ayumi Igarashi and William S. Zwicker. Fair division of graphs and of tangled cakes. *Mathematical Programming*, 203(1):931–975, 2024.

Ayumi Igarashi, Yasushi Kawase, Warut Suksompong, and Hanna Sumita. Fair division with two-sided preferences. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2756–2764, 2023.

Ayumi Igarashi, Naoyuki Kamiyama, Warut Suksompong, and Sheung Man Yuen. Reachability of fair allocations via sequential exchanges. *Algorithmica*, 86(12):3653–3683, 2024.

Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12–14):1054–1065, 2011.

Takehiro Ito, Yuni Iwamasa, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. Reforming an envy-free matching. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5084–5091, 2022.

Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. On reachable assignments under dichotomous preferences. *Theoretical Computer Science*, 979:114196, 2023.

Zsuzsanna Jankó and Attila Joó. Cutting a cake for infinitely many guests. *The Electronic Journal of Combinatorics*, 29, 2022.

Zsuzsanna Jankó, Attila Joó, Erel Segal-Halevi, and Sheung Man Yuen. On connected strongly-proportional cake-cutting. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pages 3356–3363, 2024.

Matthew Johnson, Dieter Kratsch, Stefan Kratsch, Viresh Patel, and Daniël Paulusma. Finding shortest paths between graph colourings. *Algorithmica*, 75(2):295–321, 2016.

David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM*, 64(2):8:1–8:27, 2018.

Fu Li, C. Gregory Plaxton, and Vaibhav B. Sinha. Object allocation over a network of objects: Mobile agents with strict preferences. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1578–1580, 2021.

Richard Jay Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

Shengxin Liu, Xinhang Lu, Mashbat Suzuki, and Toby Walsh. Mixed fair division: A survey. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20):22641–22649, 2024.

Zbigniew Lonc and Miroslaw Truszczynski. Maximin share allocations on cycles. *Journal of Artificial Intelligence Research*, 69:613–655, 2020.

Pasin Manurangsi and Warut Suksompong. Differentially private fair division. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5814–5822, 2023.

Carl Menger. *Principles of Economics.* Ludwig von Mises Institute, 1871. ISBN 9781610163606.

Hervé Moulin. *Fair Division and Collective Welfare.* MIT Press, 2003.

Hervé Moulin. Fair division in the internet age. *Annual Review of Economics*, 11:407–441, 2019.

Luis Müller and Matthias Bentert. On reachable assignments in cycles. In *Proceedings of the 7th International Conference on Algorithmic Decision Theory (ADT)*, pages 273–288, 2021.

Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52:1–52:25, 2018.

Svetlana Obraztsova, Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. On swap-distance geometry of voting rules. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 383–390, 2013.

Svetlana Obraztsova, Edith Elkind, and Piotr Faliszewski. On swap convexity of voting rules. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1910–1917, 2020.

Hoon Oh, Ariel D. Procaccia, and Warut Suksompong. Fairly allocating many goods with few queries. *SIAM Journal on Discrete Mathematics*, 35(2):788–813, 2021.

Benjamin Plaut and Tim Roughgarden. Communication complexity of discrete fair division. *SIAM Journal on Computing*, 49(1):206–243, 2020.

Ariel D. Procaccia. Cake cutting: Not just child's play. *Communications of the ACM*, 56(7): 78–87, 2013.

Ariel D. Procaccia. Cake cutting algorithms. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 13, pages 311–329. Cambridge University Press, 2016.

Kenneth Rebman. How to get (at least) a fair share of the cake. *Mathematical Plums (Edited by R. Honsberger), The Mathematical Association of America*, pages 22–37, 1979.

Jack Robertson and William Webb. *Cake-Cutting Algorithms: Be Fair if You Can.* Peters/CRC Press, 1998.

Walter John Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

Erel Segal-Halevi. Cake-cutting with different entitlements: How many cuts are needed? *Journal of Mathematical Analysis and Applications*, 480(1):123382, 2019.

Erel Segal-Halevi. Redividing the cake. *Autonomous Agents and Multi-Agent Systems*, 36(1): 14:1–14:36, 2022.

Erel Segal-Halevi and Warut Suksompong. How to cut a cake fairly: A generalization to groups. *American Mathematical Monthly*, 128(1):79–83, 2021.

Erel Segal-Halevi and Warut Suksompong. Cutting a cake fairly for groups revisited. *American Mathematical Monthly*, 130(3):203–213, 2023.

Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. Fair and square: Cake-cutting in two dimensions. *Journal of Mathematical Economics*, 70:1–28, 2017.

Hila Shoshan, Noam Hazon, and Erel Segal-Halevi. Efficient nearly-fair division with capacity constraints. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 206–214, 2023.

Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948.

Walter Stromquist. How to cut a cake fairly. *American Mathematical Monthly*, 87(8):640–644, 1980.

Walter Stromquist. A pie that can't be cut fairly (revised for DSP). In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2007.

Walter Stromquist. Envy-free cake divisions cannot be found by finite protocols. *The Electronic Journal of Combinatorics*, 15:#R11, 2008.

Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly*, 106(10):930–942, 1999.

Warut Suksompong. Fairly allocating contiguous blocks of indivisible items. *Discrete Applied Mathematics*, 260:227–236, 2019.

Warut Suksompong. Constraints in fair division. *ACM SIGecom Exchanges*, 19(2):46–61, 2021.

Warut Suksompong. A characterization of maximum Nash welfare for indivisible goods. *Economics Letters*, 222:110956, 2023.

William Thomson. Children crying at birthday parties. Why? *Economic Theory*, 31(3): 501–521, 2007.

William Thomson. Introduction to the theory of fair allocation. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 11, pages 261–283. Cambridge University Press, 2016.

Jamie Tucker-Foltz. Thou shalt covet the average of thy neighbors' cakes. *Information Processing Letters*, 180:106341, 2023.

Staal A. Vinterbo. A note on the hardness of the k-ambiguity problem. Technical Report DSG-TR-2002-006, Decision Systems Group/Harvard Medical School, May 2002.

Toby Walsh. Fair division: The computer scientist's perspective. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4966–4972, 2020.

William A. Webb. An algorithm for super envy-free cake division. *Journal of Mathematical Analysis and Applications*, 239(1):175–179, 1999.

Gerhard J. Woeginger and Jiří Sgall. On the complexity of cake cutting. *Discrete Optimization*, 4(2):213–220, 2007.

Douglas R. Woodall. A note on the cake-division problem. *Journal of Combinatorial Theory, Series A*, 42(2):300–301, 1986.

Bang Ye Wu, Hung-Lung Wang, Shih Ta Kuan, and Kun-Mao Chao. On the uniform edge-partition of a tree. *Discrete Applied Mathematics*, 155(10):1213–1223, 2007.

Xiaowei Wu, Bo Li, and Jiarui Gan. Budget-feasible maximum Nash social welfare allocation is almost envy-free. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 465–471, 2021.

Sheung Man Yuen and Warut Suksompong. Extending the characterization of maximum Nash welfare. *Economics Letters*, 224:111030, 2023.

Sheung Man Yuen and Warut Suksompong. Approximate envy-freeness in graphical cake cutting. *Discrete Applied Mathematics*, 357:112–131, 2024.

Sheung Man Yuen, Ayumi Igarashi, Naoyuki Kamiyama, and Warut Suksompong. Reforming an unfair allocation by exchanging goods. Under submission, 2024.

Appendix A

NP-Hardness of Directed Triangle Partition in Chapter 4

We prove Lemma 4.4.2 by reducing the well-known NP-hard problem 3SAT to Directed Triangle Partition.

- **3SAT.** Given a set of variables $Y = \{y_1, \ldots, y_q\}$ and a set of clauses $C = \{c_1, \ldots, c_r\}$ where each clause is a disjunction of three literals, i.e., $c_j = \ell_{j,1} \vee \ell_{j,2} \vee \ell_{j,3}$, and each literal is either a variable (i.e., $\ell_{j,k} = y_i$) or its negation (i.e., $\ell_{j,k} = \overline{y_i}$), determine whether there exists an assignment to the variables in Y such that every clause in C is satisfied.

- **Directed Triangle Partition.** Given a directed graph $\widetilde{G} = (V, E)$ with no directed cycles of length 1 or 2, determine whether there is a partition of the edges into triangles (i.e., directed cycles of length 3).

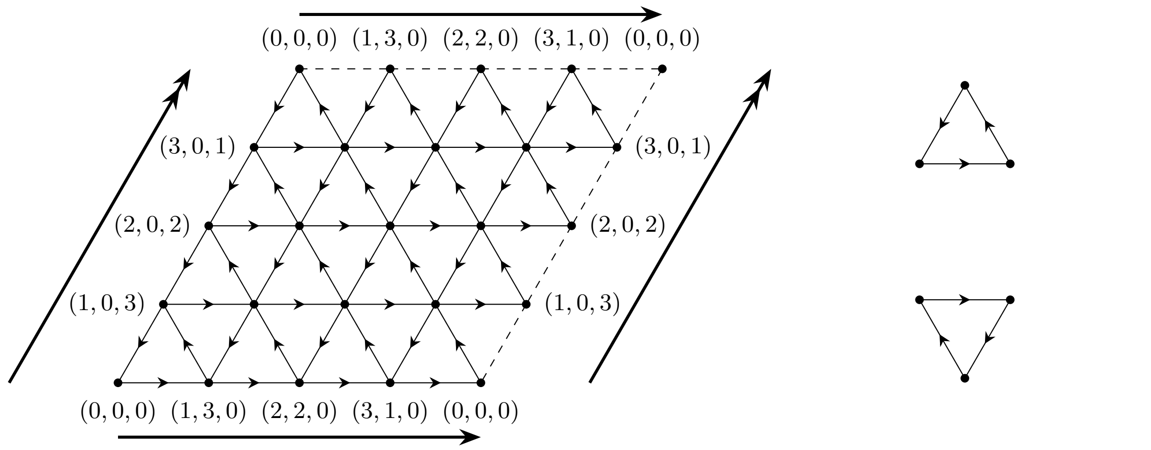
The reduction works by constructing a graph for each variable and each literal that can be edge-partitioned into triangles in exactly two ways—one representing "true" and the other representing "false"—and joining these graphs together in special ways to restrict the truth values that they represent. This idea is similar to that used by Holyer (1981) in his proof of the corresponding result for *undirected* graphs.

Define the directed graph $H_p = (V_p, E_p)$ for each positive integer p as follows:

$$
\begin{aligned}
V_p &= \{(a_1, a_2, a_3) \in \mathbb{Z}_p^3 \mid a_1 + a_2 + a_3 \equiv 0\}, \\
E_p &= \{(a_1, a_2, a_3) \to (b_1, b_2, b_3) \mid \exists (i, j, k) \in \{(1,2,3), (2,3,1), (3,1,2)\} \text{ such that} \\
&\qquad\qquad b_i \equiv a_i,\ b_j \equiv a_j + 1,\ \text{and } b_k \equiv a_k - 1\},
\end{aligned}
$$

where all equivalences are modulo p. There are only two types of triangles in H_p: *T-triangles* of the form $(a_1, a_2, a_3) \to (a_1 + 1, a_2 - 1, a_3) \to (a_1 + 1, a_2, a_3 - 1) \to (a_1, a_2, a_3)$, and *$F$-triangles* of the form $(a_1, a_2, a_3) \to (a_1 + 1, a_2 - 1, a_3) \to (a_1, a_2 - 1, a_3 + 1) \to (a_1, a_2, a_3)$. See Figure A.1 for an illustration. Note that each vertex has an indegree and an outdegree of 3.

Two triangles are called *neighbors* if they share a common edge. A *patch* is a triangle

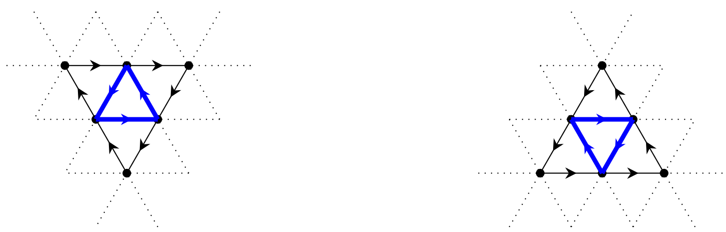


(a) The graph H_4. The opposite sides wrap around.

(b) A T-triangle (top) and an F-triangle (bottom).

Figure A.1: The graph H_4 and an example of a T-triangle and an F-triangle.

together with its neighbors. The set of edges of the triangle is called the *center* of the patch, and the set of edges of a patch that do not belong to the center is called the *exterior* of the patch. A *T-patch* (resp., *F-patch*) is a patch in which the center is a T-triangle (resp., F-triangle). See Figure A.2 for an illustration. Two patches P^1 and P^2 are *non-interfering* if the distance between any vertex in P^1 and any vertex in P^2 is at least (say) 10 on H_p, where distance is measured along a shortest path. We shall also require patches to be of distance at least 10 from the vertex $\mathbf{0} = (0, 0, 0)$.



(a) A T-patch.

(b) An F-patch.

Figure A.2: A T-patch and an F-patch. The center T-triangle and F-triangle are denoted by bold lines, and the exteriors are denoted by non-bold solid lines. The dotted lines are not part of the patches.

Consider the graph H_p with non-interfering patches and with some edges of the patches removed. Suppose there is an edge-partition of the resulting graph into triangles. The vertex $\mathbf{0}$ has an indegree and an outdegree of 3, so any edge-partition into triangles requires $\mathbf{0}$ to belong to exactly *three* triangles. The only ways to have these three triangles are when they are all T-triangles or all F-triangles. Then, all neighboring vertices to $\mathbf{0}$ belong to triangles of the same type. By a similar argument, the neighboring vertices must each belong to exactly three triangles of the same type. This cascades through the whole graph (except possibly at the patches), and therefore, we see that an edge-partition of H_p into triangles necessarily consists only of T-triangles or only of F-triangles (except possibly at the patches).

Let H_p^1, H_p^2, and H_p^3 be three copies of H_p, and let P_F^k be an F-patch on H_p^k for each

$k \in \{1, 2, 3\}$. We say that we apply an *F-F-F join* on (H_p^1, H_p^2, H_p^3) if we remove the patches P_F^1, P_F^2, and P_F^3 on the respective copies and replace them by *one* copy of the vertices of an F-patch P_F and *one* copy of the exterior of P_F. See Figure A.3 for an illustration. We claim that any edge-partition of this new graph into triangles results in *exactly one* of H_p^1, H_p^2, and H_p^3 being partitioned into F-triangles (and the other two into T-triangles). To see this, consider an edge $x = \mathbf{a} \to \mathbf{b}$ belonging to the exterior of P_F. Since x belongs to a triangle, we consider the candidates for the third vertex of the triangle. There are only three such candidates: $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$, which are parallel vertices on H_p^1, H_p^2, and H_p^3, respectively.

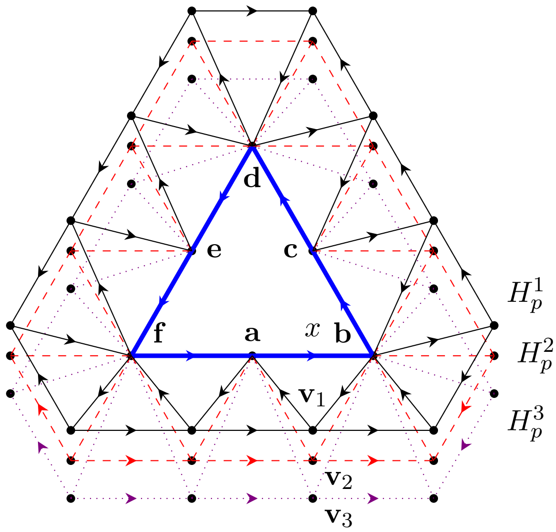
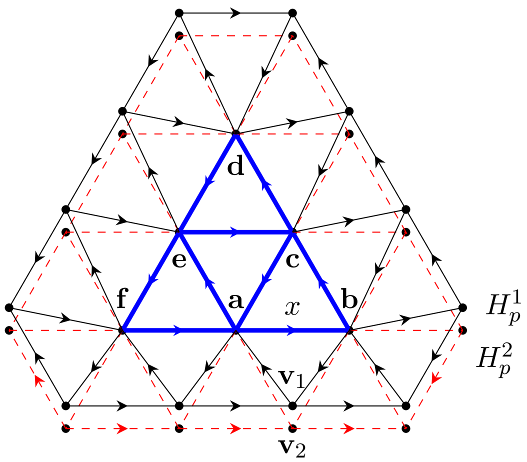


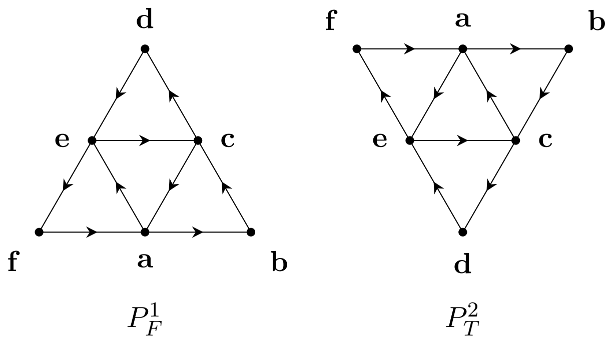
Figure A.3: An F-F-F join on H_p^1 (solid lines), H_p^2 (dashed lines) and H_p^3 (dotted lines). All three graphs share the exterior of the patch P_F (bold lines).

Assume without loss of generality that the triangle is $\mathbf{a} \to \mathbf{b} \to \mathbf{v}_1 \to \mathbf{a}$ (note that this is an F-triangle), and consider the triangles containing the vertex $\mathbf{v}_1$. Since $\mathbf{v}_1$ already has an F-triangle, the other triangles containing $\mathbf{v}_1$ can only be F-triangles, and the cascading effect implies that $\mathbf{0}$ in H_p^1, and all other vertices in H_p^1 (except possibly at P_F), belong to F-triangles. Note that this implies that each edge in the exterior of P_F is combined with edges in H_p^1 to form F-triangles. On the other hand, the indegree and the outdegree of both $\mathbf{v}_2$ and $\mathbf{v}_3$ are 3, so there must be exactly three triangles containing $\mathbf{v}_2$ and $\mathbf{v}_3$, respectively. Since x is used by the edges $\mathbf{b} \to \mathbf{v}_1$ and $\mathbf{v}_1 \to \mathbf{a}$ in H_p^1, it cannot be used by the edges $\mathbf{b} \to \mathbf{v}_2$ and $\mathbf{v}_2 \to \mathbf{a}$ in H_p^2 or $\mathbf{b} \to \mathbf{v}_3$ and $\mathbf{v}_3 \to \mathbf{a}$ in H_p^3 to form the respective F-triangles. As such, the triangles containing $\mathbf{v}_2$ and $\mathbf{v}_3$ must all be T-triangles, and the cascading effect implies that $\mathbf{0}$ in H_p^2 and H_p^3, and all other vertices in H_p^2 and H_p^3 (except at P_F), belong to T-triangles. It can be verified that these edge-partitions into triangles are indeed valid.

Let H_p^1 and H_p^2 be two copies of H_p, and let P_F^k be an F-patch on H_p^k for each $k \in \{1, 2\}$. We say that we apply an *F-F join* on (H_p^1, H_p^2) if we remove the patches P_F^1 and P_F^2 on the respective copies and replace them by *one* copy of an F-patch P_F (note that this construction is slightly different from the F-F-F join, as the center of the patch is also included here). See Figure A.4(a) for an illustration. We claim that any edge-partition of this new graph into triangles results in *at least one* of H_p^1 and H_p^2 being partitioned into T-triangles. Similar to the proof for the F-F-F join, consider an edge $x = \mathbf{a} \to \mathbf{b}$ belonging to the exterior of P_F. There are now *three* possible candidates for the third vertex of the triangle containing x: $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{c}$, where $\mathbf{v}_1$ and $\mathbf{v}_2$ are parallel vertices on H_p^1 and H_p^2, respectively, and $\mathbf{c}$ is a vertex in the center of P_F. If the third vertex of the triangle is $\mathbf{v}_1$ or $\mathbf{v}_2$, then the same proof for the F-F-F join can be used to conclude that exactly one of H_p^1 and H_p^2 is partitioned into T-triangles. Otherwise, the third vertex is $\mathbf{c}$. In this case, the other edges in the center of the F-patch can only belong to T-triangles, which implies that both H_p^1 and H_p^2 can also only be partitioned into T-triangles. This shows that at least one of H_p^1 and H_p^2 is partitioned into T-triangles. It can be verified that these edge-partitions into triangles are indeed valid.



(a) An F-F join.



(b) An F-T join.

Figure A.4: (a) An F-F join on H_p^1 (solid lines) and H_p^2 (dashed lines). Both graphs share the patch P_F (bold lines). (b) The association of vertices between P_F^1 and P_T^2 in an F-T join on H_p^1 and H_p^2—note the labelling of vertices.

Let H_p^1 and H_p^2 be two copies of H_p, and let P_F^1 be an F-patch on H_p^1 and P_T^2 be a T-patch on H_p^2. We say that we apply an F-T *join* on (H_p^1, H_p^2) if we remove the patches P_F^1 and P_T^2 on the respective copies and replace them by one copy of an F-patch P_F—here, the replacement of P_T^2 by P_F is "mirrored". See Figure A.4(b) for an illustration, where the mirroring is across the edge $\mathbf{e} \to \mathbf{c}$. We claim that any edge-partition of this new graph into triangles results in H_p^1 being partitioned into T-triangles or H_p^2 being partitioned into F-triangles (or both). The proof is similar to that of the F-F join, except that we reverse the argument regarding H_p^2 due to the "mirror" effect on P_T^2.

We are now ready to prove our result.

Lemma 4.4.2. *Directed Triangle Partition is NP-hard.*

Proof. We shall reduce 3SAT to Directed Triangle Partition. Recall that in an instance of 3SAT, we are given a set of variables $Y = \{y_1, \ldots, y_q\}$ and a set of clauses $C = \{c_1, \ldots, c_r\}$ where each clause is a disjunction of three literals, i.e., $c_j = \ell_{j,1} \vee \ell_{j,2} \vee \ell_{j,3}$, and each literal is either a variable (i.e., $\ell_{j,k} = y_i$) or its negation (i.e., $\ell_{j,k} = \overline{y_i}$).

Choose p large enough so that there are at least $3r$ T-patches and $3r$ F-patches in H_p that are pairwise non-interfering (say, $p = 100r$). Assign to each variable y_i a separate copy of the graph Y_i isomorphic to H_p, and assign to each literal $\ell_{j,k}$ a separate copy of the graph $L_{j,k}$ isomorphic to H_p. For each j, apply an F-F-F join on $(L_{j,1}, L_{j,2}, L_{j,3})$ via any F-patch in $L_{j,k}$. For each (j,k), if the literal $\ell_{j,k}$ corresponds to the variable y_i, apply an F-F join on $(L_{j,k}, Y_i)$ via any unused F-patches, and if the literal $\ell_{j,k}$ corresponds to $\overline{y_i}$, apply an F-T join on $(L_{j,k}, Y_i)$ via any unused F-patch and T-patch.

Let $\widetilde{G} = (V, E)$ denote the constructed graph. This construction can be done in time polynomial in the size of the 3SAT instance. Note that $\widetilde{G}$ is a directed graph with no cycles of length 1 or 2. We claim that there exists a satisfying assignment in the 3SAT instance if and only if $\widetilde{G}$ can be edge-partitioned into triangles.

Suppose that there exists a partition of the edges of $\widetilde{G}$ into triangles. Consider one such partition, and assign y_i as true if and only if Y_i is partitioned into T-triangles. For each j, note that $L_{j,k}$ is partitioned into F-triangles for some $k \in \{1,2,3\}$ due to the F-F-F join—we claim that the corresponding literal $\ell_{j,k}$ is satisfied. If $\ell_{j,k} = y_i$ for some i, then Y_i must be partitioned into T-triangles by the F-F join on $(L_{j,k}, Y_i)$, which means that $\ell_{j,k} = y_i$ is true. If $\ell_{j,k} = \overline{y_i}$ for some i, then Y_i must be partitioned into F-triangles by the F-T join on $(L_{j,k}, Y_i)$, which means that y_i is false and $\ell_{j,k} = \overline{y_i}$ is true. In both cases, we see that the literal $\ell_{j,k}$ is satisfied.

Conversely, suppose there exists a satisfying assignment in the 3SAT instance, and consider any satisfying assignment. For each i, if y_i is true, partition Y_i into T-triangles; else, partition Y_i into F-triangles. For each j, at least one of the literals in c_j is true; pick any one of them, say $\ell_{j,k}$, and partition $L_{j,k}$ into F-triangles, and partition the other two $L_{j,k'}$ into T-triangles. We now verify that the edge-partition is a valid partition by checking that the restrictions caused by the joins are not violated. For each j, consider the F-F-F join on $(L_{j,1}, L_{j,2}, L_{j,3})$—since one $L_{j,k}$ is edge-partitioned into F-triangles and the other two into T-triangles, the requirement on the F-F-F join is satisfied. Now, for each (j,k), if the literal $\ell_{j,k}$ corresponds to the variable y_i, then the join on $(L_{j,k}, Y_i)$ is F-F, and at least one of $L_{j,k}$ and Y_i is partitioned into T-triangles (otherwise, if both are partitioned into F-triangles, then y_i is false and $\ell_{j,k}$ is true, which is not possible). On the other hand, if the literal $\ell_{j,k}$ corresponds to $\overline{y_i}$, then the join on $(L_{j,k}, Y_i)$ is F-T, and $L_{j,k}$ is partitioned into T-triangles or Y_i is partitioned into F-triangles, or both (otherwise, if $L_{j,k}$ is partitioned into F-triangles and Y_i is partitioned into T-triangles, then both $\overline{y_i} = \ell_{j,k}$ and y_i are true, which is not possible). Therefore, the edges of $\widetilde{G}$ can be partitioned into triangles. □

Appendix B

Other Results from Chapter 5

B.1 Worst-Case Bounds for Identical Utilities

We continue the discussion from Section 5.5 on worst-case bounds, and focus on identical utilities in this section.

Given n and s, let $f_{\mathrm{id}}(n,s)$ be the smallest integer such that for every instance with n agents with *identical* utilities and ns goods and every s-balanced allocation $\mathcal{A}$ in the instance, there exists an EF1 allocation that can be reached from $\mathcal{A}$ using at most $f_{\mathrm{id}}(n,s)$ exchanges.

A tight bound for two agents is an immediate consequence of our previous results.

Theorem B.1.1. *Let s be a positive integer. Then, $f_{\mathrm{id}}(2,s) = \lfloor s/2 \rfloor$.*

Proof. The lower bound follows from Theorem 5.5.1, while the upper bound follows from Theorem 5.5.2. □

For three or more agents, we conjecture that $f_{\mathrm{id}}(n,s)$ is roughly $sn/4$, like $f_{\mathrm{id,bin}}(n,s)$. However, proving this turns out to be surprisingly challenging. We shall present a result using a slightly weaker fairness notion in the case of three agents.

We say that agent i is *weak-EF1 towards* agent j in an allocation $\mathcal{A} = (A_1, \dots, A_n)$ if $u_i(A_i) \geq u_i(A_j) - \max_{g \in G} u_i(g)$; note the condition $g \in G$ as opposed to $g \in A_j$ for EF1. An allocation $\mathcal{A}$ is *weak-EF1* if every agent is weak-EF1 towards every other agent in $\mathcal{A}$. Weak-EF1 is the fairness notion originally considered by Lipton et al. (2004) (although their algorithm satisfies EF1), and weak-EF1 and EF1 are equivalent when the utilities are binary. Since we consider identical utilities, we use u instead of u_i. Without loss of generality, we may divide all utilities by $\max_{g \in G} u(g)$. Then, the utility of each good is in $[0,1]$, and the condition for agent i to be weak-EF1 towards agent j is $u(A_i) \geq u(A_j) - 1$.

Given n and s, let $\tilde{f}_{\mathrm{id}}(n,s)$ be the smallest integer such that for every instance with n agents with identical utilities and ns goods, and every s-balanced allocation $\mathcal{A}$ in the instance, there exists a weak-EF1 allocation that can be reached from $\mathcal{A}$ using at most $\tilde{f}_{\mathrm{id}}(n,s)$ exchanges. We shall determine the value of $\tilde{f}_{\mathrm{id}}(3,s)$.

We describe an algorithm $\mathfrak{A}$ that performs a sequence of exchanges of goods starting from an initial allocation $\mathcal{A}^0$. For each t starting from 0, we begin with the allocation $\mathcal{A}^t = (A_1^t, \dots, A_n^t)$. If $\mathcal{A}^t$ is weak-EF1, then we are done and the algorithm terminates.

Otherwise, we perform an exchange of goods between two agents to reach the allocation $\mathcal{A}^{t+1} = (A_1^{t+1}, \ldots, A_n^{t+1})$. For each agent k, let g_k^t and h_k^t be a good of the highest utility and a good of the lowest utility in agent k's bundle, A_k^t, respectively. Let i_t be an agent with the most valuable bundle, i.e., $i_t = \arg\max_{k \in N} u(A_k^t)$, and j_t be an agent with the least valuable bundle, i.e., $j_t = \arg\min_{k \in N} u(A_k^t)$; we may resolve ties arbitrarily. Note that agent j_t is not weak-EF1 towards agent i_t—otherwise, $\mathcal{A}^t$ is weak-EF1—and hence $i_t \neq j_t$. We then exchange $g_{i_t}^t$ with $h_{j_t}^t$ to form $\mathcal{A}^{t+1}$, i.e., $A_{i_t}^{t+1} = (A_{i_t}^t \setminus \{g_{i_t}^t\}) \cup \{h_{j_t}^t\}$, $A_{j_t}^{t+1} = (A_{j_t}^t \setminus \{h_{j_t}^t\}) \cup \{g_{i_t}^t\}$, and $A_k^{t+1} = A_k^t$ for all $k \in N \setminus \{i_t, j_t\}$. Subsequently, we increment t by 1 and repeat the procedure.

To establish our result, we prove a series of lemmas on properties of this algorithm.

Lemma B.1.2. *Let $\mathcal{A}^t$ be an allocation which is not weak-EF1. Then, $u(g_{i_t}^t) > u(h_{j_t}^t)$.*

Proof. If $u(g_{i_t}^t) \leq u(h_{j_t}^t)$, then $u(A_{i_t}^t) - 1 \leq u(A_{i_t}^t) \leq s \cdot u(g_{i_t}^t) \leq s \cdot u(h_{j_t}^t) \leq u(A_{j_t}^t)$, so agent j_t is weak-EF1 towards agent i_t, and therefore $\mathcal{A}^t$ is weak-EF1, a contradiction. Hence, $u(g_{i_t}^t) > u(h_{j_t}^t)$. □

Lemma B.1.3. *Let $\mathcal{A}^t$ be an allocation which is not weak-EF1. Then, in $\mathcal{A}^{t+1}$,*

- *agent i_t is weak-EF1 towards every agent; and*
- *every agent is weak-EF1 towards agent j_t.*

Proof. Let $k \in N \setminus \{i_t, j_t\}$. Note that $u(A_k^{t+1}) = u(A_k^t)$.

Since $i_t = \arg\max_{\ell \in N} u(A_\ell^t)$, we have

$$u(A_{i_t}^{t+1}) = u((A_{i_t}^t \setminus \{g_{i_t}^t\}) \cup \{h_{j_t}^t\}) \geq u(A_{i_t}^t \setminus \{g_{i_t}^t\}) \geq u(A_{i_t}^t) - 1 \geq u(A_k^t) - 1 = u(A_k^{t+1}) - 1,$$

showing that agent i_t is weak-EF1 towards agent k.

Similarly, since $j_t = \arg\min_{\ell \in N} u(A_\ell^t)$, we have

$$u(A_k^{t+1}) = u(A_k^t) \geq u(A_{j_t}^t) = u((A_{j_t}^{t+1} \cup \{h_{j_t}^t\}) \setminus \{g_{i_t}^t\}) \geq u(A_{j_t}^{t+1} \setminus \{g_{i_t}^t\}) \geq u(A_{j_t}^{t+1}) - 1,$$

showing that agent k is weak-EF1 towards agent j_t.

Finally, since agent j_t is not weak-EF1 towards agent i_t, we have $u(A_{j_t}^t) < u(A_{i_t}^t) - 1$. Thus,

$$\begin{aligned}
u(A_{i_t}^{t+1}) &= u((A_{i_t}^t \setminus \{g_{i_t}^t\}) \cup \{h_{j_t}^t\}) \\
&\geq u(A_{i_t}^t \setminus \{g_{i_t}^t\}) \\
&\geq u(A_{i_t}^t) - 1 \\
&> u(A_{j_t}^t) \\
&= u((A_{j_t}^{t+1} \cup \{h_{j_t}^t\}) \setminus \{g_{i_t}^t\}) \\
&\geq u(A_{j_t}^{t+1} \setminus \{g_{i_t}^t\}) \\
&\geq u(A_{j_t}^{t+1}) - 1,
\end{aligned}$$

showing that agent i_t is weak-EF1 towards agent j_t. □

For each $t \geq 0$, call i_t a *strong agent* and j_t a *weak agent*. Let $I^0 = J^0 = \varnothing$, and for each $t \geq 0$, let $I^{t+1} = I^t \cup \{i_t\}$ be the set of strong agents up to round t, and $J^{t+1} = J^t \cup \{j_t\}$ be the set of weak agents up to round t.

Lemma B.1.4. *Let $t \geq 0$ be given such that $\mathcal{A}^0, \ldots, \mathcal{A}^t$ are not weak-EF1. Then, $I^{t+1} \cap J^{t+1} = \varnothing$.*

Proof. Suppose on the contrary that there exists an agent k such that $k \in I^{t+1} \cap J^{t+1}$. Let t_p be the smallest index such that $k \in I^{t_p+1}$, and t_q be the smallest index such that $k \in J^{t_q+1}$. Then, we have $k = i_{t_p} = j_{t_q}$. Note that $t_p \neq t_q$, since $i_{t'} \neq j_{t'}$ for all t'.

Suppose first that $t_p < t_q$. We show by induction that agent k is weak-EF1 towards every agent in $\mathcal{A}^{t'+1}$ for all $t' \in \{t_p, \ldots, t\}$. The base case of $t' = t_p$ is true by Lemma B.1.3 since $k = i_{t_p}$. For the inductive step, suppose that agent k is weak-EF1 towards every agent in $\mathcal{A}^{t'+1}$ for some $t' \in \{t_p, \ldots, t-1\}$. Then, agent k cannot be $j_{t'+1}$. If agent k is $i_{t'+1}$, then agent k is weak-EF1 towards every agent in $\mathcal{A}^{t'+2}$ by Lemma B.1.3, making the inductive statement true. If agent k is not $i_{t'+1}$, then agent k does not take part in the exchange going from $\mathcal{A}^{t'+1}$ to $\mathcal{A}^{t'+2}$.

- Agent k is weak-EF1 towards agent $i_{t'+1}$ in $\mathcal{A}^{t'+2}$ since agent k is weak-EF1 towards $i_{t'+1}$ in $\mathcal{A}^{t'+1}$ by the inductive hypothesis, and agent $i_{t'+1}$'s utility of her own bundle decreases after the exchange by Lemma B.1.2.
- Agent k is weak-EF1 towards agent $j_{t'+1}$ in $\mathcal{A}^{t'+2}$ by Lemma B.1.3.
- Agent k is weak-EF1 towards every other agent in $\mathcal{A}^{t'+2}$ since their bundles did not change from $\mathcal{A}^{t'+1}$.

Overall, these show that agent k is weak-EF1 towards every agent in $\mathcal{A}^{t'+2}$, proving the inductive statement. Since agent k is weak-EF1 towards every agent in $\mathcal{A}^{t'+1}$ for all $t' \in \{t_p, \ldots, t\}$, agent k can never be j_{t_q}. This shows that $t_p < t_q$ is false.

Therefore, we must have $t_p > t_q$. The argument for this case is similar to that for the previous case. We show by induction that every agent is weak-EF1 towards agent k in $\mathcal{A}^{t'+1}$ for all $t' \in \{t_q, \ldots, t\}$. The base case of $t' = t_q$ is true by Lemma B.1.3 since $k = j_{t_q}$. For the inductive step, suppose that every agent is weak-EF1 towards agent k in $\mathcal{A}^{t'+1}$ for some $t' \in \{t_q, \ldots, t-1\}$. Then, agent k cannot be $i_{t'+1}$. If agent k is $j_{t'+1}$, then every agent is weak-EF1 towards agent k in $\mathcal{A}^{t'+2}$ by Lemma B.1.3, making the inductive statement true. If agent k is not $j_{t'+1}$, then agent k does not take part in the exchange going from $\mathcal{A}^{t'+1}$ to $\mathcal{A}^{t'+2}$.

- Agent $i_{t'+1}$ is weak-EF1 towards agent k in $\mathcal{A}^{t'+2}$ by Lemma B.1.3.
- Agent $j_{t'+1}$ is weak-EF1 towards agent k in $\mathcal{A}^{t'+2}$ since agent $j_{t'+1}$ is weak-EF1 towards agent k in $\mathcal{A}^{t'+1}$ by the inductive hypothesis, and agent $j_{t'+1}$'s utility of her own bundle increases after the exchange by Lemma B.1.2.
- Every other agent is weak-EF1 towards agent k in $\mathcal{A}^{t'+2}$ since their bundles did not change from $\mathcal{A}^{t'+1}$.

Overall, these show that every agent is weak-EF1 towards agent k in $\mathcal{A}^{t'+2}$, proving the inductive statement. Since every agent is weak-EF1 towards agent k in $\mathcal{A}^{t'+1}$ for all $t' \in \{t_q, \ldots, t\}$, agent k can never be i_{t_p}. This yields the desired contradiction. $\square$

Lemma B.1.5. *Let $t \geq 0$ be given such that $\mathcal{A}^0, \ldots, \mathcal{A}^t$ are not weak-EF1. Then,*

- *for any $i \in I^{t+1}$, $u(A_i^0) \geq \cdots \geq u(A_i^{t+1})$; and*
- *for any $j \in J^{t+1}$, $u(A_j^0) \leq \cdots \leq u(A_j^{t+1})$.*

Proof. At every time step $t' \in \{0, \ldots, t\}$, an agent $i \in I^{t+1}$ cannot be a weak agent by Lemma B.1.4. Therefore, agent i either takes part in the exchange from $\mathcal{A}^{t'}$ to $\mathcal{A}^{t'+1}$ as a strong agent $i_{t'}$ or does not take part in the exchange. The utility of agent i's bundle either decreases in the former case due to Lemma B.1.2 or remains the same in the latter case. An analogous argument holds for $j \in J^{t+1}$. $\square$

Lemma B.1.6. *Each good is not exchanged more than once in algorithm $\mathfrak{A}$.*

Proof. Suppose on the contrary that some good g is exchanged more than once. We first consider the case where g is in a strong agent's bundle in $\mathcal{A}^0$ and is exchanged for the first time at round t, i.e., $g = g_{i_t}^t$. After its first exchange, the good is now with agent j_t. By Lemma B.1.4, $j_t \notin I^{t'}$ for any $t' > t$. Since the good is exchanged again, it must be that $g = h_{j_{t'}}^{t'}$ for some $t' > t$, where $j_{t'} = j_t$. Then, we have

$$\begin{aligned}
u(A_{i_{t'}}^t) - 1 &\geq u(A_{i_{t'}}^{t'}) - 1 && \text{(by Lemma B.1.5 on } i_{t'} \in I^{t'+1}\text{)} \\
&> u(A_{j_{t'}}^{t'}) && \text{(since } j_{t'} \text{ is not weak-EF1 towards } i_{t'}\text{)} \\
&\geq s \cdot u(g) && \text{(since } g \text{ is the least valuable good in } A_{j_{t'}}^{t'}\text{)} \\
&\geq u(A_{i_t}^t) && \text{(since } g \text{ is the most valuable good in } A_{i_t}^t\text{)} \\
&> u(A_{i_t}^t) - 1,
\end{aligned}$$

which means that agent j_t should have exchanged goods with agent $i_{t'}$ at round t instead of with agent i_t. This contradiction shows that a good in a strong agent's bundle in $\mathcal{A}^0$ cannot be exchanged more than once.

Analogously, we now consider the case where g is in a weak agent's bundle in $\mathcal{A}^0$ and is exchanged for the first time at round t, i.e., $g = h_{j_t}^t$. After its first exchange, the good is now with agent i_t. By Lemma B.1.4, $i_t \notin J^{t'}$ for any $t' > t$. Since the good is exchanged again, it must be that $g = g_{i_{t'}}^{t'}$ for some $t' > t$, where $i_{t'} = i_t$. Then, we have

$$\begin{aligned}
u(A_{j_{t'}}^t) &\leq u(A_{j_{t'}}^{t'}) && \text{(by Lemma B.1.5 on } j_{t'} \in J^{t'+1}\text{)} \\
&< u(A_{i_{t'}}^{t'}) - 1 && \text{(since } j_{t'} \text{ is not weak-EF1 towards } i_{t'}\text{)} \\
&< u(A_{i_{t'}}^{t'}) \\
&\leq s \cdot u(g) && \text{(since } g \text{ is the most valuable good in } A_{i_{t'}}^{t'}\text{)} \\
&\leq u(A_{j_t}^t), && \text{(since } g \text{ is the least valuable good in } A_{j_t}^t\text{)}
\end{aligned}$$

which means that agent i_t should have exchanged goods with agent $j_{t'}$ at round t instead of with agent j_t. This contradiction shows that a good in a weak agent's bundle in $\mathcal{A}^0$ also cannot be exchanged more than once. $\square$

Lemma B.1.7. *Algorithm $\mathfrak{A}$ terminates in finite time.*

Proof. Since each good is not exchanged more than once by Lemma B.1.6, at most $\lfloor m/2 \rfloor$ pairs of goods can be exchanged, and the algorithm terminates by round $\lfloor m/2 \rfloor$. $\square$

Since the algorithm terminates in finite time by Lemma B.1.7, there exists $T \geq 0$ such that $\mathcal{A}^0, \ldots, \mathcal{A}^T$ are not weak-EF1 but $\mathcal{A}^{T+1}$ is weak-EF1. Let $I = I^{T+1}$ be the set of strong agents and $J = J^{T+1}$ be the set of weak agents. By Lemma B.1.4, I and J are disjoint sets of agents. Therefore, at each round $t \in \{0, \ldots, T\}$ of the algorithm, some agent $i_t \in I$ exchanges a good with some agent $j_t \in J$.

We derive a bound on the number of steps that $\mathfrak{A}$ takes in the case of two agents.

Lemma B.1.8. *For $n = 2$ agents with s goods each, algorithm $\mathfrak{A}$ terminates after at most $\lfloor s/2 \rfloor$ rounds.*

Proof. The statement is clear when $s = 1$, so we assume that $s \geq 2$. Suppose on the contrary that after $T = \lfloor s/2 \rfloor$ rounds, the allocation $\mathcal{A}^T$ is still not weak-EF1. Without loss of generality, assume that $1 \in I$ and $2 \in J$. Then, the most valuable $\lfloor s/2 \rfloor$ goods from agent 1's bundle A_1^0 are exchanged with the least valuable $\lfloor s/2 \rfloor$ goods from agent 2's bundle A_2^0 to reach $\mathcal{A}^T$. Let $B_1 \subseteq A_1^0$ and $B_2 \subseteq A_2^0$ be the sets of goods from the respective bundles that are exchanged between the two agents, and let $C_1 = A_1^0 \setminus B_1$ and $C_2 = A_2^0 \setminus B_2$. Note that all these sets are disjoint by Lemma B.1.6. Let g be any arbitrary good in C_1, and let $C_1' = C_1 \setminus \{g\}$. We have $|B_1| = |B_2| = \lfloor s/2 \rfloor$, $|C_1| = |C_2| = \lceil s/2 \rceil$, and $|C_1'| \leq \lfloor s/2 \rfloor$.

Now, $u(B_1) \geq u(C_1')$ since the goods with the highest values from A_1^0 are exchanged and B_1 has at least as many goods as C_1'. Also, $u(C_2) \geq u(B_2)$ since the goods with the lowest values from A_2^0 are exchanged and C_2 has at least as many goods as B_2. Therefore, we have

$$u(A_2^T) = u(B_1 \cup C_2) \geq u(C_1' \cup B_2) = u(A_1^T \setminus \{g\}) \geq u(A_1^T) - 1,$$

which shows that agent 2 is weak-EF1 towards agent 1 in $\mathcal{A}^T$. On the other hand, agent 1 is also weak-EF1 towards agent 2 in $\mathcal{A}^T$ due to Lemma B.1.3 applied on $\mathcal{A}^{T-1}$. This shows that $\mathcal{A}^T$ is weak-EF1, a contradiction. $\square$

We now come to our main lemma, which bounds the number of steps that $\mathfrak{A}$ takes for three agents. For convenience of the analysis, we focus on the case where s is divisible by 3.

Lemma B.1.9. *Let s be a positive integer divisible by 3. For $n = 3$ agents with s goods each, algorithm $\mathfrak{A}$ terminates after at most $2s/3$ rounds.*

Proof. Suppose on the contrary that after $T = 2s/3$ rounds, the allocation $\mathcal{A}^T$ is still not weak-EF1. Note that $T > 0$, so $I^T, J^T \neq \varnothing$. If $|I^T| = |J^T| = 1$, then after at most $\lfloor s/2 \rfloor$ rounds, the agent $i \in I^T$ and the agent $j \in J^T$ are weak-EF1 towards each other by Lemma B.1.8, while the agent $k \in N \setminus \{i, j\}$ is weak-EF1 towards everyone and vice versa since agent k does not partake in the exchanges. Since $\lfloor s/2 \rfloor \le 2s/3$, the allocation $\mathcal{A}^{T'}$ is weak-EF1 for some $T' \le 2s/3$, contradicting our assumption. Therefore, we must have $I^T \cup J^T = N$.

Case 1: $|I^T| = 1$. We consider the allocation $\mathcal{A}^T$ relative to $\mathcal{A}^0$. Without loss of generality, let $1 \in I^T$. Let $B_{1,2} \subseteq A_1^0$ and $B_2 \subseteq A_2^0$ be the sets of the goods in the respective bundles that are exchanged between agents 1 and 2, $B_{1,3} \subseteq A_1^0$ and $B_3 \subseteq A_3^0$ be the sets of the goods in the respective bundles that are exchanged between agents 1 and 3, and let $C_1 = A_1^0 \setminus (B_{1,2} \cup B_{1,3})$, $C_2 = A_2^0 \setminus B_2$, and $C_3 = A_3^0 \setminus B_3$. Note that all these sets are disjoint by Lemma B.1.6. Let $x = |B_{1,2}| = |B_2|$ and $y = |B_{1,3}| = |B_3|$. We have $x + y = 2s/3$, $|C_2| = s - x$, $|C_3| = s - y$, and $|C_1| = s - x - y = s/3$. Without loss of generality, let $x \ge y$. Note that $x \le s/2$, since otherwise agent 2 will be weak-EF1 towards agent 1 by Lemma B.1.8 and does not need to exchange more goods with agent 1.

In $\mathcal{A}^0$, we have $A_1^0 = B_{1,2} \cup B_{1,3} \cup C_1$, $A_2^0 = B_2 \cup C_2$, and $A_3^0 = B_3 \cup C_3$. In $\mathcal{A}^T$, we have $A_1^T = B_2 \cup B_3 \cup C_1$, $A_2^T = B_{1,2} \cup C_2$, and $A_3^T = B_{1,3} \cup C_3$. Since the algorithm always exchanges the most valuable goods from agent 1's bundle and the least valuable goods from agent 2's and agent 3's bundles, we have $u(B_{1,2})/x \ge u(C_1)/(s/3)$, $u(B_{1,3})/y \ge u(C_1)/(s/3)$, $u(C_2)/(s-x) \ge u(B_2)/x$, and $u(C_3)/(s-y) \ge u(B_3)/y$. By Lemma B.1.2, we have $u(B_{1,2}) \ge u(B_2)$ and $u(B_{1,3}) \ge u(B_3)$.

Since $x \ge y$, we have $s/3 \le x \le s/2$ and hence $s/6 \le y \le s/3$. Let $\alpha = (6x-s)/(3x+s) = 2 - 3s/(3x+s) = 2 - s/(s-y)$. Since $s/3 \le x \le s/2$, we have $1/2 \le \alpha \le 4/5$.

Let $\alpha_2 = \alpha(s-x)/x$ and $\alpha_3 = (1-\alpha)(s-y)/y$. Since $1/3 \le x/s \le 1/2$, we have $\alpha \ge x/s$, which implies that $\alpha_2 = \alpha(s-x)/x = \alpha s/x - \alpha \ge 1 - \alpha$. On the other hand, the derivative of $\alpha_2 = \alpha(s-x)/x$ with respect to x is

$$\begin{aligned}
\left(\frac{6x-s}{3x+s}\right)\left(-\frac{s}{x^2}\right) + \left(\frac{9s}{(3x+s)^2}\right)\left(\frac{s-x}{x}\right) &= \left(\frac{s}{x(3x+s)}\right)\left(\frac{s-6x}{x} + \frac{9s-9x}{3x+s}\right) \\
&= \left(\frac{s}{x(3x+s)}\right)\left(\frac{(s+9x)(s-3x)}{x(3x+s)}\right).
\end{aligned}$$

When the derivative of α_2 with respect to x is equal to 0, we get $x = -s/9$ or $x = s/3$. It can be verified that α_2 attains a local maximum at $x = s/3$. For $x \in [s/3, s/2]$, the maximum value of α_2 is hence equal to 1 at $x = s/3$. Together, we have $1 - \alpha \le \alpha_2 \le 1$.

Now,

$$\alpha_3 = (1-\alpha)\frac{s-y}{y} = \left(1 - \left(2 - \frac{s}{s-y}\right)\right)\frac{s-y}{y} = -\frac{s-y}{y} + \frac{s}{y} = 1.$$

We shall show that $\alpha u(A_2^T) + (1-\alpha)u(A_3^T) \ge u(A_1^T)$. We have

$$\begin{aligned}
&\alpha u(A_2^T) + (1-\alpha)u(A_3^T) \\
&= \alpha(u(B_{1,2}) + u(C_2)) + (1-\alpha)(u(B_{1,3}) + u(C_3)) \\
&\ge \alpha u(B_{1,2}) + \frac{\alpha(s-x)}{x}u(B_2) + (1-\alpha)u(B_{1,3}) + \frac{(1-\alpha)(s-y)}{y}u(B_3) \\
&= \alpha u(B_{1,2}) + \alpha_2 u(B_2) + (1-\alpha)u(B_{1,3}) + \alpha_3 u(B_3)
\end{aligned}$$

$$
\begin{aligned}
&= (\alpha+\alpha_2-1)u(B_{1,2}) + (1-\alpha_2)u(B_{1,2}) + \alpha_2 u(B_2) + (1-\alpha)u(B_{1,3}) + u(B_3) \\
&\geq (\alpha+\alpha_2-1)u(B_{1,2}) + (1-\alpha_2)u(B_2) + \alpha_2 u(B_2) + (1-\alpha)u(B_{1,3}) + u(B_3) \\
&= (\alpha+\alpha_2-1)u(B_{1,2}) + u(B_2) + (1-\alpha)u(B_{1,3}) + u(B_3) \\
&\geq (\alpha+\alpha_2-1)\frac{3x}{s}u(C_1) + u(B_2) + (1-\alpha)\frac{3y}{s}u(C_1) + u(B_3) \\
&= \frac{3}{s}\left((\alpha+\alpha_2)x - x + (1-\alpha)y\right)u(C_1) + u(B_2) + u(B_3).
\end{aligned}
$$

Since $\alpha_2 x = \alpha(s-x)$ implies $(\alpha+\alpha_2)x = \alpha s$ and $y = \alpha_3 y = (1-\alpha)(s-y)$ implies $(1-\alpha)y = (1-\alpha)s - y$, the expression $(\alpha+\alpha_2)x - x + (1-\alpha)y$ simplifies to $\alpha s - x + (1-\alpha)s - y$, which gives $s - x - y$. Using the fact that $x + y = 2s/3$, the expression simplifies to $s/3$. Therefore,

$$
\begin{aligned}
\alpha u(A_2^T) + (1-\alpha)u(A_3^T) &\geq \frac{3}{s}\left(\frac{s}{3}\right)u(C_1) + u(B_2) + u(B_3) \\
&= u(C_1) + u(B_2) + u(B_3) \\
&= u(A_1^T).
\end{aligned}
$$

Let $j \in \arg\max_{k\in\{1,2,3\}} u(A_k^T)$. Since $\alpha u(A_2^T) + (1-\alpha)u(A_3^T) \geq u(A_1^T)$ for some $\alpha \in (0,1)$, we may assume that $j \in J^T$. Suppose without loss of generality that $j = 2$. Note that agent 2 is weak-EF1 towards every other agent in $\mathcal{A}^T$. Agent 1 is weak-EF1 towards every other agent in $\mathcal{A}^T$ by Lemma B.1.3. Let $t < T$ be the round that agent 2 exchanges a good with agent 1 for the final time, i.e., agent 2 exchanges a good with agent 1 going from $\mathcal{A}^t$ to $\mathcal{A}^{t+1}$. Then, by Lemma B.1.3, agent 3 is weak-EF1 towards agent 2 in $\mathcal{A}^{t+1}$. Since the utility of agent 3's bundle does not decrease thereafter and agent 2's bundle remains the same thereafter, agent 3 is weak-EF1 towards agent 2 in $\mathcal{A}^T$. Then, agent 3 is weak-EF1 towards every other agent in $\mathcal{A}^T$. This shows that $\mathcal{A}^T$ is weak-EF1, contradicting the original assumption.

Case 2: $|I^T| = 2$. We consider the allocation $\mathcal{A}^T$ relative to $\mathcal{A}^0$. Without loss of generality, let $1 \in J^T$. Let $B_{1,2} \subseteq A_1^0$ and $B_2 \subseteq A_2^0$ be the sets of the goods in the respective bundles that are exchanged between agents 1 and 2, $B_{1,3} \subseteq A_1^0$ and $B_3 \subseteq A_3^0$ be the sets of the goods in the respective bundles that are exchanged between agents 1 and 3, and let $C_1 = A_1^0 \setminus (B_{1,2} \cup B_{1,3})$, $C_2 = A_2^0 \setminus B_2$, and $C_3 = A_3^0 \setminus B_3$. Note that all these sets are disjoint by Lemma B.1.6. Let $x = |B_{1,2}| = |B_2|$ and $y = |B_{1,3}| = |B_3|$. We have $x + y = 2s/3$, $|C_2| = s - x$, $|C_3| = s - y$, and $|C_1| = s - x - y = s/3$. Without loss of generality, let $x \geq y$. Note that $x \leq s/2$, since otherwise agent 1 will be weak-EF1 towards agent 2 by Lemma B.1.8 and does not need to exchange more goods with agent 2.

In $\mathcal{A}^0$, we have $A_1^0 = B_{1,2} \cup B_{1,3} \cup C_1$, $A_2^0 = B_2 \cup C_2$, and $A_3^0 = B_3 \cup C_3$. In $\mathcal{A}^T$, we have $A_1^T = B_2 \cup B_3 \cup C_1$, $A_2^T = B_{1,2} \cup C_2$, and $A_3^T = B_{1,3} \cup C_3$. Since the algorithm always exchanges the least valuable goods from agent 1's bundle and the most valuable goods from agent 2's and agent 3's bundles, we have $u(B_{1,2})/x \leq u(C_1)/(s/3)$, $u(B_{1,3})/y \leq u(C_1)/(s/3)$, $u(C_2)/(s-x) \leq u(B_2)/x$, and $u(C_3)/(s-y) \leq u(B_3)/y$. By Lemma B.1.2, we have $u(B_{1,2}) \leq u(B_2)$ and $u(B_{1,3}) \leq u(B_3)$.

Since $x \geq y$, we have $s/3 \leq x \leq s/2$. Let $\alpha = (6x-s)/(3x+s) = 2 - 3s/(3x+s) = 2 - s/(s-y)$, $\alpha_2 = \alpha(s-x)/x$, and $\alpha_3 = (1-\alpha)(s-y)/y$. By the same reasoning as in Case

1, we have $1/2 \leq \alpha \leq 4/5$, $1-\alpha \leq \alpha_2 \leq 1$, and $\alpha_3 = 1$.

We shall show that $\alpha u(A_2^T) + (1-\alpha)u(A_3^T) \leq u(A_1^T)$. We have

$$\begin{aligned}
&\alpha u(A_2^T) + (1-\alpha)u(A_3^T) \\
&= \alpha(u(B_{1,2}) + u(C_2)) + (1-\alpha)(u(B_{1,3}) + u(C_3)) \\
&\leq \alpha u(B_{1,2}) + \frac{\alpha(s-x)}{x}u(B_2) + (1-\alpha)u(B_{1,3}) + \frac{(1-\alpha)(s-y)}{y}u(B_3) \\
&= \alpha u(B_{1,2}) + \alpha_2 u(B_2) + (1-\alpha)u(B_{1,3}) + \alpha_3 u(B_3) \\
&= (\alpha+\alpha_2-1)u(B_{1,2}) + (1-\alpha_2)u(B_{1,2}) + \alpha_2 u(B_2) + (1-\alpha)u(B_{1,3}) + u(B_3) \\
&\leq (\alpha+\alpha_2-1)u(B_{1,2}) + (1-\alpha_2)u(B_2) + \alpha_2 u(B_2) + (1-\alpha)u(B_{1,3}) + u(B_3) \\
&= (\alpha+\alpha_2-1)u(B_{1,2}) + u(B_2) + (1-\alpha)u(B_{1,3}) + u(B_3) \\
&\leq (\alpha+\alpha_2-1)\frac{3x}{s}u(C_1) + u(B_2) + (1-\alpha)\frac{3y}{s}u(C_1) + u(B_3) \\
&= \frac{3}{s}\left((\alpha+\alpha_2)x - x + (1-\alpha)y\right)u(C_1) + u(B_2) + u(B_3).
\end{aligned}$$

By the same reasoning as in Case 1, we have $(\alpha+\alpha_2)x - x + (1-\alpha)y = s/3$, and therefore, $\alpha u(A_2^T) + (1-\alpha)u(A_3^T) \leq u(A_1^T)$.

Let $i \in \arg\min_{k\in\{1,2,3\}} u(A_k^T)$. Since $\alpha u(A_2^T)+(1-\alpha)u(A_3^T) \leq u(A_1^T)$ for some $\alpha \in (0,1)$, we may assume that $i \in I^T$. Suppose without loss of generality that $i = 2$. Note that every agent is weak-EF1 towards agent 2 in $\mathcal{A}^T$. Every agent is weak-EF1 towards agent 1 in $\mathcal{A}^T$ by Lemma B.1.3. Let $t < T$ be the round that agent 2 exchanges a good with agent 1 for the final time, i.e., agent 2 exchanges a good with agent 1 going from $\mathcal{A}^t$ to $\mathcal{A}^{t+1}$. Then, by Lemma B.1.3, agent 2 is weak-EF1 towards agent 3 in $\mathcal{A}^{t+1}$. Since the utility of agent 3's bundle does not increase thereafter and agent 2's bundle remains the same thereafter, agent 2 is weak-EF1 towards agent 3 in $\mathcal{A}^T$. Then, every agent is weak-EF1 towards agent 3 in $\mathcal{A}^T$. This shows that $\mathcal{A}^T$ is weak-EF1, contradicting the original assumption. □

We are now ready to show the result on $\tilde{f}_{\mathrm{id}}(n,s)$ for three agents.

Theorem B.1.10. *Let s be a positive integer divisible by 3. Then, $\tilde{f}_{\mathrm{id}}(3,s) = 2s/3$.*

Proof. The lower bound of $\tilde{f}_{\mathrm{id}}(3,s)$ follows from Theorem 5.5.1—note that weak-EF1 and EF1 are equivalent for binary utilities. The upper bound follows from Lemma B.1.9. □

B.2 Beneficial Exchanges

Let us say that an exchange is *beneficial* if the two agents involved in the exchange strictly benefit from the exchange, i.e., if the goods $g \in A_i$ and $g' \in A_{i'}$ are exchanged, then $u_i(g') > u_i(g)$ and $u_j(g) > u_j(g')$. In this section, we investigate the decision problem of whether a given initial allocation can be reformed into an EF1 allocation using *only* beneficial exchanges. For convenience, we refer to this problem as Beneficial Exchanges.

We show that Beneficial Exchanges is NP-complete, even for binary utilities, using a reduction from Minimum k-Coverage. In Minimum k-Coverage, we are given positive integers k, ℓ, p, q such that $k \leq q$ and $\ell \leq p$, a set $X = \{x_1, \ldots, x_q\}$, and a collection

$C = \{Y_1, \ldots, Y_p\}$ of subsets of X. The problem is to decide whether there exists a set $I \subseteq \{1, \ldots, p\}$ of indices such that $|I| = \ell$ and $|\bigcup_{i \in I} Y_i| \leq k$. This decision problem is known to be NP-hard (Vinterbo, 2002).

Theorem B.2.1. *Beneficial Exchanges is NP-complete for binary utilities.*

Proof. For membership in NP, observe that in a sequence of beneficial exchanges for binary utilities, each good $g \in G$ can only be part of at most one exchange. Indeed, if good g is part of at least two beneficial exchanges, then it must be received by some agent i (and hence worth 1 to i) and be given away by agent i (and hence worth 0 to i), which is impossible. Therefore, such a sequence consists of at most $m/2$ exchanges, and can be used as a certificate for polynomial-time verification.

It remains to show that the problem is NP-hard. Let an instance of Minimum k-Coverage be given. Define an instance of Beneficial Exchanges as follows. There are $n = 2p + q + k - \ell$ agents and $m = 2n$ goods. We shall label the agents $a_{1,1}, \ldots, a_{1,q}$, $a_{2,1}, \ldots, a_{2,k}$, $a_{3,1}, \ldots, a_{3,p}$, $a_{4,1}, \ldots, a_{4,p-\ell}$; we use $u_{i,j}$ for the utility of agent $a_{i,j}$. For each agent $a_{i,j}$, there are two goods $g^0_{i,j}$ and $g^1_{i,j}$ that are both in agent $a_{i,j}$'s bundle in the initial allocation. The valuable goods for the agents are as follows:

- For $i \in \{1, \ldots, q\}$, $u_{1,i}(g^1_{2,j}) = 1$ for all $j \in \{1, \ldots, k\}$. Additionally, if $x_i \in Y_j$ for some $j \in \{1, \ldots, p\}$, then $u_{1,i}(g^0_{3,j}) = u_{1,i}(g^1_{3,j}) = 1$.
- For $i \in \{1, \ldots, k\}$, $u_{2,i}(g^1_{1,j}) = 1$ for all $j \in \{1, \ldots, q\}$.
- For $i \in \{1, \ldots, p\}$, $u_{3,i}(g^1_{4,j}) = 1$ for all $j \in \{1, \ldots, p - \ell\}$.
- For $i \in \{1, \ldots, p - \ell\}$, $u_{4,i}(g^1_{3,j}) = 1$ for all $j \in \{1, \ldots, p\}$.

All other goods not mentioned above are worth 0 to the respective agents. This reduction can be done in polynomial time.

In the initial allocation, every agent has zero utility for her own bundle, and the only agents who are possibly not EF1 are agents $a_{1,i}$, who envy $a_{3,j}$ if $x_i \in Y_j$. By construction, the only possible beneficial exchanges are between $g^1_{1,i}$ in agent $a_{1,i}$'s bundle and $g^1_{2,j}$ in agent $a_{2,j}$'s bundle, or between $g^1_{3,i}$ in agent $a_{3,i}$'s bundle and $g^1_{4,j}$ in agent $a_{4,j}$'s bundle.

We claim that the initial allocation can be reformed into an EF1 allocation via only beneficial exchanges if and only if there exists a set $I \subseteq \{1, \ldots, p\}$ of indices such that $|I| = \ell$ and $|\bigcup_{i \in I} Y_i| \leq k$.

($\Leftarrow$) Suppose that there exists a set $I \subseteq \{1, \ldots, p\}$ of indices such that $|I| = \ell$ and $|\bigcup_{i \in I} Y_i| \leq k$.

- Let $I' = \{1, \ldots, p\} \setminus I$. Since $|I'| = p - \ell$, there exists a bijection $\sigma : I' \to \{1, \ldots, p - \ell\}$. For each $i' \in I'$, exchange $g^1_{3,i'}$ in agent $a_{3,i'}$'s bundle with $g^1_{4,\sigma(i')}$ in agent $a_{4,\sigma(i')}$'s bundle.
- Let $J = \{j \mid x_j \in \bigcup_{i \in I} Y_i\}$. Since $|J| \leq k$, there exists an injection $\phi : J \to \{1, \ldots, k\}$. For each $j \in J$, exchange $g^1_{1,j}$ in agent $a_{1,j}$'s bundle with $g^1_{2,\phi(j)}$ in agent $a_{2,\phi(j)}$'s bundle.

We now show that the new allocation is EF1. It is easy to see that the allocation is EF1 for agents $a_{2,i}$, $a_{3,i}$, and $a_{4,i}$, since every other agent has at most one of their valuable goods. Therefore, it suffices to show that the allocation is EF1 for agents $a_{1,j}$.

- If $j \in J$, then agent $a_{1,j}$ has the valuable good $g^1_{2,\phi(j)}$ in her bundle, so her utility of her own bundle is at least 1. Since her utility of every other agent's bundle is at most 2, agent $a_{1,j}$ is EF1 towards every other agent.
- If $j \notin J$, then $x_j \notin \bigcup_{i \in I} Y_i$, and so $x_j \notin Y_i$ for all $i \in I$. Note that the valuable goods for $a_{1,j}$ are possibly in the form $g^1_{2,i}$, $g^0_{3,i}$, and $g^1_{3,i}$. Suppose on the contrary that $a_{1,j}$ is not EF1 towards some agent. This agent must have two such goods in the final allocation. The only way for this to happen is when there exists i^* such that agent a_{3,i^*} has both g^0_{3,i^*} and g^1_{3,i^*}. This means agent a_{3,i^*} had not exchanged any goods, and so $i^* \notin I'$. This implies that $i^* \in I$. Since $x_j \notin Y_i$ for all $i \in I$, we must have $u_{1,j}(g^0_{3,i^*}) = u_{1,j}(g^1_{3,i^*}) = 0$. This contradicts the assumption that $a_{1,j}$ is not EF1 towards agent a_{3,i^*}. Therefore, $a_{1,j}$ is EF1 towards every agent.

($\Rightarrow$) Suppose that the initial allocation can be reformed into an EF1 allocation via only beneficial exchanges. Consider one such sequence of beneficial exchanges. Let $I' \subseteq \{1, \dots, p\}$ be the set of all indices i' such that agent $a_{3,i'}$ exchanged a good with another agent in this sequence. Since $a_{3,i'}$ can only exchange a good with some $a_{4,i''}$ once, and there are only $p - \ell$ agents of the form $a_{4,i''}$, we have $|I'| \leq p - \ell$. Therefore, $I_0 := \{1, \dots, p\} \setminus I'$ has cardinality at least ℓ, and I_0 contains indices i such that agent $a_{3,i}$ retains her original bundle from the initial allocation.

We claim that $|\bigcup_{i \in I_0} Y_i| \leq k$. Let $J \subseteq \{1, \dots, q\}$ be the set of all indices j such that agent $a_{1,j}$ exchanged a good with another agent in this sequence. Since $a_{1,j}$ can only exchange a good with some $a_{2,j''}$ once, and there are only k agents of the form $a_{2,j''}$, we have $|J| \leq k$. Therefore, $J' := \{1, \dots, q\} \setminus J$ has cardinality at least $q - k$, and J' contains all indices j' such that agent $a_{1,j'}$ retains her original bundle from the initial allocation; these agents have utility 0. Since the final allocation is EF1, these agents do not envy agents $a_{3,i}$ by more than one good for each $i \in I_0$. Therefore, we must have $u_{1,j'}(g^0_{3,i}) = u_{1,j'}(g^1_{3,i}) = 0$, which implies that $x_{j'} \notin Y_i$ for all $j' \in J'$ and $i \in I_0$. This means that $x_{j'} \notin \bigcup_{i \in I_0} Y_i$ for every $j' \in J'$. Therefore, at least $q - k$ of the x_j's are not in $\bigcup_{i \in I_0} Y_i$, which shows that $\bigcup_{i \in I_0} Y_i$ has cardinality at most $q - (q - k) = k$, as claimed.

Finally, take any subset $I \subseteq I_0$ with cardinality ℓ. The proof is completed by noting that $\bigcup_{i \in I} Y_i \subseteq \bigcup_{i \in I_0} Y_i$. □

While we have shown that a sequence of beneficial exchanges must be of polynomial length for binary utilities, the same statement in fact holds for general utilities. Indeed, for any sequence of beneficial exchanges, for each good g_{t_1} in some agent i's initial bundle, it is exchanged with another good g_{t_2}, which is subsequently exchanged with another good g_{t_3}, and so on, until some g_{t_k} in agent i's final bundle. Since we must have $u_i(g_{t_1}) < \cdots < u_i(g_{t_k})$ due to the exchanges being beneficial, it must hold that $k \leq m$, and so there are at most $m-1$ exchanges starting from g_{t_1}. Since there are m goods in total and each exchange involves two goods, the maximum number of exchanges in the sequence is $m(m-1)/2$. Hence, by Theorem B.2.1, we have NP-completeness for general utilities as well.

Appendix C

Appendix for Chapter 6

C.1 Left-Marks and Right-Marks

Our results assume that algorithms have access to eval and right-mark queries in the Robertson-Webb model. We show that our results also hold if algorithms have access to eval and *left-mark* queries.

For each agent $i \in N$, value $r \in [0, 1]$, and point $x \in C$, define *left-mark* such that $\textit{Left-Mark}_i(x, r)$ is the *leftmost* (smallest) point $z \in C$ such that $u_i([x, z]) = r$ (such a point exists due to the continuity of the valuations); if $u_i([x, 1]) < r$, then $\textsc{Left-Mark}_i(x, r)$ returns ∞. We define $\textit{Right-Mark}_i(x, r)$ to be the same as $\textsc{Mark}_i(x, r)$ in Section 6.2, but use "right-mark" in this section to avoid confusion with left-mark. Note that for a hungry agent i, $\textsc{Left-Mark}_i(x, r) = \textsc{Right-Mark}_i(x, r)$ for all $x \in C, r \in [0, 1]$.

Let $\mathcal{I}$ be an instance with n agents with valuation functions $(u_i)_{i \in N}$ and entitlements $\mathbf{w}$. A *mirrored instance* $\overleftarrow{\mathcal{I}}$ of $\mathcal{I}$ is the instance with n agents with valuation functions $(\overleftarrow{u}_i)_{i \in N}$ and entitlements $\mathbf{w}$ such that $\overleftarrow{u}_i([x, y]) = u_i([1 - y, 1 - x])$ for all $x, y \in C$ with $x \leq y$. In other words, the cake in $\overleftarrow{\mathcal{I}}$ is "mirrored" from the cake in $\mathcal{I}$. A class of instances $\mathcal{C}$ is *closed under mirror* if $\mathcal{I} \in \mathcal{C}$ implies that $\overleftarrow{\mathcal{I}} \in \mathcal{C}$.

Note that the classes of instances we consider in our results are closed under mirror. We now show that if there is an algorithm, having access to right-mark queries, that can determine the existence of a connected strongly-proportional allocation in a class of instances closed under mirror, then there exists another algorithm, having access to left-mark queries, that can also do the same in the same class of instances. Furthermore, the number of queries made by the new algorithm is within a factor of 2 from the number of queries made by the old algorithm. This shows that our results hold in whichever model of Robertson-Webb, and the use of right-mark is only for convenience.

Proposition C.1.1. *Let $\mathcal{C}$ be a class of instances closed under mirror. Suppose there exists an algorithm $\mathfrak{A}$ such that for each instance in $\mathcal{C}$, algorithm $\mathfrak{A}$ can determine the existence of a connected strongly-proportional allocation in the instance using at most k eval and right-mark queries and using no left-mark queries. Then, there exists an algorithm $\mathfrak{B}$ such that for each instance in $\mathcal{C}$, algorithm $\mathfrak{B}$ can determine the existence of a connected strongly-proportional allocation in the instance using at most $2k$ eval and left-mark queries and using no right-mark* *queries.*

Proof. Let $\mathcal{I}$ be an instance in $\mathcal{C}$ in which we wish to determine the existence of a connected strongly-proportional allocation using eval and left-mark queries and using no right-mark queries. By assumption, there exists an algorithm $\mathfrak{A}$ that can determine the existence of a connected strongly-proportional allocation in $\overleftarrow{\mathcal{I}}$ using at most k eval and right-mark queries and using no left-mark queries.

Our algorithm $\mathfrak{B}$ simulates algorithm $\mathfrak{A}$ on $\mathcal{I}$ as follows:

- **Case 1: $\mathfrak{A}$ makes an $\textsc{Eval}_i(x, y)$ query on $\overleftarrow{\mathcal{I}}$.**
  Then, algorithm $\mathfrak{B}$ makes an $\textsc{Eval}_i(1 - y, 1 - x)$ query on $\mathcal{I}$.

- **Case 2: $\mathfrak{A}$ makes a $\textsc{Right-Mark}_i(x, r)$ query on $\overleftarrow{\mathcal{I}}$.**
  Then, algorithm $\mathfrak{B}$ makes a $\textsc{Left-Mark}_i(0, \textsc{Eval}_i(0, 1 - x) - r)$ query on $\mathcal{I}$.

Let us verify that the two implementations are identical.

- Suppose $r = \textsc{Eval}_i(x, y)$ on $\overleftarrow{\mathcal{I}}$. Then, $\textsc{Eval}_i(1-y, 1-x)$ on $\mathcal{I}$ is equal to $u_i([1-y, 1-x])$, which is equal to $\overleftarrow{u}_i([x, y]) = r$.

- Suppose $y = \textsc{Right-Mark}_i(x, r)$ on $\overleftarrow{\mathcal{I}}$. Then, since the value of $\textsc{Eval}_i(0, 1-x)$ on $\mathcal{I}$ is equal to $u_i([0, 1-x])$ and $\textsc{Left-Mark}_i(0, k)$ on $\mathcal{I}$ is equal to $1-\textsc{Right-Mark}_i(0, 1-k)$ on $\overleftarrow{\mathcal{I}}$ for any $k \in [0, 1]$, $\textsc{Left-Mark}_i(0, \textsc{Eval}_i(0, 1 - x) - r)$ on $\mathcal{I}$ is equal to $1 - \textsc{Right-Mark}_i(0, 1 - (u_i([0, 1 - x]) - r))$ on $\overleftarrow{\mathcal{I}}$. But $1 - u_i([0, 1 - x]) = 1 - \overleftarrow{u}_i([x, 1]) = \overleftarrow{u}_i([0, x])$, which means that the result is equal to $1 - \textsc{Right-Mark}_i(0, \overleftarrow{u}_i([0, x]) + r)$. This is equal to $1 - \textsc{Right-Mark}_i(x, r) = 1 - y$, which is the mirrored point of y.

This shows that if algorithm $\mathfrak{A}$ determines the existence of a connected strongly-proportional allocation on $\overleftarrow{\mathcal{I}}$, then $\mathfrak{B}$ determines the existence of a connected strongly-proportional allocation on $\mathcal{I}$. Note that algorithm $\mathfrak{B}$ makes at most two times the number of queries that $\mathfrak{A}$ makes. $\square$

C.2 Proportionality

With unequal entitlements, even a connected *proportional* allocation may not exist. Algorithm 6.3 can be modified to determine the existence of a connected proportional allocation for agents with unequal entitlements (and to output one if it exists) by using *left-marks* instead of right-marks.

Proposition C.2.1. *Algorithm C.1 decides whether a connected proportional allocation exists for n agents using at most $n \cdot 2^{n-1}$ queries.*

We remark that our algorithm is similar to that in Aumann et al. (2012, Theorem 4) where they find an approximate optimum egalitarian welfare on a piece of cake—here, we extend it to agents with possibly unequal entitlements.

Algorithm C.1 Determining the existence of a connected proportional allocation for n agents.

$b_\varnothing \leftarrow 0$
for $k = 1, \ldots, n$ **do**
    for each subset $N' \subseteq N$ with $|N'| = k$ **do**
        $b_{N'} \leftarrow \infty$
        for each agent $i \in N'$ **do**
            $y \leftarrow$ LEFT-MARK$_i(b_{N' \setminus \{i\}}, w_i)$
            if $y < b_{N'}$ **then** $b_{N'} \leftarrow y$ ▷ this finds the "best" $b_{N'}$
        end for
    end for
end for
if $b_N \leq 1$ **then return** true **else return** false

Appendix D

The Divide Algorithm in Chapter 7

In this section, we describe the algorithm DIVIDE (Algorithm D.1), which takes as input a connected subgraph H worth β_0 to some agent in $N' \subseteq N$, a positive threshold $\beta \leq \beta_0$, and a root vertex r of H. The output of DIVIDE satisfies the conditions stated in Lemma 7.2.1.

We will work with rooted trees. Let $T = (V, E)$ be a rooted tree. For a vertex $v \in V$, let S_v be the subtree at v, that is, S_v is the subgraph induced by v and all of its descendants in T. For a child vertex w of v, let $S_{v,w}$ be the subgraph induced by v, w, and all descendants of w in T.

We begin by converting the graph H into a tree (Lines 1 to 8). As long as H contains a cycle, select an edge $[v_1, v_2]$ belonging to the cycle, add a new vertex v_2', and replace the edge $[v_1, v_2]$ with the edge $[v_1, v_2']$ while keeping the remaining edges of the graph. Note that the new edge created should have the same value to each agent as the one it replaces. This procedure decreases the number of cycles in the graph by at least one. Therefore, by repeating this procedure, H eventually becomes a tree. Note that any connected share of this tree corresponds to a connected share in the original graph with the same value for every agent.

We now split the tree H into two subtrees H_1 and H_2 (Lines 9 to 25). To do so, traverse the tree from the root vertex r until some vertex v is reached such that the subtree S_v is worth at least β to some agent in N', while the subtree S_w at each child vertex w of v is worth less than β to all agents in N'. Since S_r is worth at least β to some agent while S_z is worth 0 to all agents for every leaf vertex z, there must be some vertex v where the condition holds. We consider two cases.

- **Case 1 (Lines 14 to 17): There exists an agent $i \in N'$ and a child vertex w of v such that $u_i(S_{v,w}) \geq \beta$.**
  Let w^* be one such child vertex. By assumption, the subtree S_{w^*} is worth less than β to all agents. Find the point $x \in [w^*, v]$ closest to w^* such that $S_{x,w}$ is worth exactly β to some agent i^* and at most β to all other agents. (Here we abuse notation slightly and treat x as a vertex.) Let $S_{x,w}$ be the first share, and the remaining portion of H be the second share. Note that a new vertex $x \in [w^*, v]$ is created and belongs to both shares. The first share is worth β to some agent and at most β to every agent.

- **Case 2 (Lines 19 to 24): For all agents $i \in N'$ and all child vertices w of v,**

Algorithm D.1 DIVIDE(H, N', β, r).

Input: Connected subgraph H, set of agents N', threshold $\beta \in (0, \beta_0]$ where $\beta_0 = u_i(H)$ for some $i \in N'$, vertex r of H.
Output: Graphs H_1 and H_2.

1: $L \leftarrow \varnothing$ (a list of vertices)
2: **while** there exists a cycle in H **do**
3:     $[v_1, v_2] \leftarrow$ any edge on the cycle
4:     add a new vertex v_2' to the graph
5:     append v_2' to L
6:     $[v_1, v_2'] \leftarrow [v_1, v_2]$
7:     delete edge $[v_1, v_2]$ from the graph
8: **end while** (this converts H into a tree)
9: $v \leftarrow r$
10: **while** there exists $i \in N'$ and a child vertex w of v such that $u_i(S_w) \geq \beta$ **do**
11:     $v \leftarrow$ any child vertex w of v with $u_i(S_w) \geq \beta$
12: **end while** (S_v is worth at least β to some agent)
13: **if** there exists $i \in N'$ and a child vertex w of v such that $u_i(S_{v,w}) \geq \beta$ **then**
14:     $w^* \leftarrow$ any child vertex w of v with $u_i(S_{v,w}) \geq \beta$
15:     $x \leftarrow$ the point in $[w^*, v]$ closest to w^* such that there exists $i \in N'$ with $u_i(S_{x,w^*}) = \beta$
16:     $H_1 \leftarrow S_{x,w^*}$
17:     $H_2 \leftarrow H \setminus H_1$ (add the point x to H_2)
18: **else**
19:     $C \leftarrow \varnothing$ (a list of child vertices of v)
20:     **while** $\sum_{w \in C} u_i(S_{v,w}) < \beta$ for all $i \in N'$ **do**
21:         add to C any child vertex of v that is not yet in C
22:     **end while**
23:     $H_1 \leftarrow \bigcup_{w \in C} S_{v,w}$
24:     $H_2 \leftarrow H \setminus H_1$ (add the point v to H_2)
25: **end if**
26: **while** $L \neq \varnothing$ **do**
27:     reverse lines 3 to 7 based on the last element of L
28:     remove from L its last element
29:     adjust H_1 and H_2 accordingly
30: **end while** (this converts the graph back to the original H)
31: **return** (H_1, H_2)

we have $u_i(S_{v,w}) < \beta$.

Initialize H_1 to be a graph with only vertex v. For each child vertex w of v, iteratively add $S_{v,w}$ to H_1 until H_1 is worth at least β to some agent i^*. Let H_1 be the first share, and the remaining portion of H be the second share. Note that the vertex v belongs to both shares. The first share is worth at least β to some agent and less than 2β to all agents, because H_1 is worth less than β *before* the last $S_{v,w}$ is added, and the last $S_{v,w}$ is also worth less than β, so their combined value is less than 2β to every agent.

Lastly, we convert the tree back to the original graph (Lines 26 to 30). This is done simply by reversing the steps taken to convert the original graph into a tree. Along the way, we also adjust the two shares accordingly. At each step, each of the two shares is connected since the vertices that are removed are always leaf vertices.