FAIR RESOURCE ALLOCATION IN RICH DOMAINS

LU XINHANG

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

August 2021

FAIR RESOURCE ALLOCATION IN RICH DOMAINS

LU XINHANG

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the
degree of Doctor of Philosophy

August 2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me except where otherwise stated in this thesis. The thesis work has not been submitted for a degree or professional qualification to any other university or institution. I declare that this thesis is written by myself and is free of plagiarism and of sufficient grammatical clarity to be examined. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

20 August 2021	陆馨杭
. . . .	. . . .
Date	Lu Xinhang

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it of sufficient grammatical clarity to be examined. To the best of my knowledge, the thesis is free of plagiarism and the research and writing are those of the candidate's except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

20 August 2021	
. . . .	. . . .
Date	Bei Xiaohui

Authorship Attribution Statement

Please select one of the following; *delete as appropriate:

*(A) ~~This thesis **does not** contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.~~

*(B) This thesis contains material from four papers published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 4 is published as Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. Fair division of mixed divisible and indivisible goods. *Artificial Intelligence (AIJ)*, 293: 103436, 2021. doi:10.1016/j.artint.2020.103436. Preliminary version in AAAI'20.

The contributions of the co-authors are as follows:

- By convention, publications in theoretical computer science list authors in alphabetical order and authors are considered as equal contribution. This is a theory work and all results were obtained from discussion. After the publication of the initial version of this work, Mr. Zihao Li pointed out a bug in a proof in the conference version of this paper.
- All of us were involved in preparing the conference submission. Specifically, Prof. Jinyan Liu prepared the very first draft of the conference version of this paper; next, I took over the preparation; the draft was later revised by Profs. Xiaohui Bei and Shengxin Liu.
- All of us were involved in preparing the journal version. I took the lead on the journal revision.

Chapter 5 is published as Xiaohui Bei, Shengxin Liu, Xinhang Lu, and Hongao Wang. Maximin fairness with mixed divisible and indivisible goods. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 35(2):34, 2021. doi:10.1007/s10458-021-09517-7. Preliminary version in AAAI'21.

The contributions of the co-authors are as follows:

- By convention, publications in theoretical computer science list authors in alphabetical order and authors are considered as equal contribution. This is a theory work and all results were obtained from discussion.
- Mr. Hongao Wang and I prepared the draft of the conference version of this paper, which was later revised by Profs. Xiaohui Bei and Shengxin Liu.
- All of us were involved in preparing the journal version. I took the lead on the journal revision.

Chapter 6 is published as Xiaohui Bei, Xinhang Lu, Pasin Manurangsi, and Warut Suksompong. The price of fairness for indivisible goods. *Theory of Computing Systems (TOCS)*, Forthcoming. doi:10.1007/s00224-021-10039-8. Preliminary version in IJCAI'19.

The contributions of the co-authors are as follows:

- By convention, publications in theoretical computer science list authors in alphabetical order and authors are considered as equal contribution. This is a theory work and all results were obtained from discussion.

- All of us were involved in preparing the conference submission by writing different parts of the paper. The manuscript was overall taken care of by Prof. Warut Suksompong.

Chapter 7 is published as Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. The price of connectivity in fair division. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5151–5158, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16651`.

The contributions of the co-authors are as follows:

- By convention, publications in theoretical computer science list authors in alphabetical order and authors are considered as equal contribution. This is a theory work and all results were obtained from discussion.

- All of us were involved in preparing the submission. I wrote the initial version of the proofs; Prof. Warut Suksompong took over the writing.

20 August 2021	陆馨杭
Date	Lu Xinhang

Abstract

How should one allocate scarce resources among a group of people in a satisfactory manner when the participants have different preferences over the resources, like how to divide an inheritance or ration healthcare? *Fairness* is one of the desirable properties in resource allocation applications. This thesis aims at allocating scarce resources among interested agents in such a way that every agent involved feels that she gets a fair share. To this end, we present our results in three parts: (i) axiomatic study of fairness for mixed goods, (ii) quantitative measure of trade-offs between competing objectives, and (iii) mechanism design for sharing public goods.

In Part I, we study the problem of fair division when the resources to be divided consist of both divisible and indivisible goods, or mixed goods for short. In this setting, we propose a novel notion of fairness, *envy-freeness for mixed goods (EFM)*, and show that EFM can always be satisfied for any number of agents with additive valuations. In addition, we extend the analysis of a well-known fairness notion of maximin share (MMS) guarantee to the mixed goods setting by providing insights on how the MMS approximation guarantee would be affected when divisible resources are introduced as well as designing algorithms that could find allocations with good MMS approximations.

In Part II, we focus on indivisible goods settings, and study trade-offs between fairness and other desiderata. An issue orthogonal to fairness is *efficiency*, or social welfare. The concept of *(strong) price of fairness* quantitatively measures the worst-case loss of social welfare due to fairness constraints. We study the (strong) price of fairness for indivisible goods by focusing on several well-established fairness notions with guaranteed existence. Next, in the same vein, we introduce the *price of connectivity* to quantify the price in terms of fairness that we have to pay if we desire connectivity in a fair division model where indivisible goods form an undirected graph and each agent must receive a connected subgraph, and derive bounds on this quantity.

In Part III, we depart from the "division" framework and study a public goods variant of the classic cake cutting problem where instead of competing with one another for the cake, the agents all share the same subset of the cake which must be chosen subject to a length constraint. We refer to this setting as *cake sharing*. Our focus in this setting is on the design of *truthful* and *fair* mechanisms in the presence of strategic agents.

Acknowledgements

First and foremost, I would like to thank my supervisor, Xiaohui Bei, for the great support, the wise guidance, and the constant encouragement. Back then, Xiaohui gave a talk on *Efficient Algorithms for Unknown Markets* at the Shanghai Theory Day; I was there and totally captivated by his clear, interesting talk and his research at the time. I feel extremely happy and fortunate that later I have the opportunity to do research under his supervision. For the past four years, Xiaohui has guided me with patience, provided opportunities in all aspects to develop me into a better researcher, and furthermore, inspired me with his quick and clear thoughts, deep research insights, and incredible intuitions.

Next, I wish to thank Bo An and Yi Li for serving as my Thesis Advisory Committee members as well as Yi Li, Huaxiong Wang, and Zhenzhen Yan for serving as the Oral Qualifying Examination examiners. I would like to extend my gratitude to external and internal examiners for their helpful comments/suggestions to my thesis as well as the Oral Examination Panel members.

I have been fortune to collaborate with many wonderful people on various interesting research projects, including Haris Aziz, Xiaohui Bei, Hau Chan, Jiarui Gan, Ayumi Igarashi, Bo Li, Zihao Li, Jinyan Liu, Shengxin Liu, Pasin Manurangsi, Warut Suksompong, and Hongao Wang. It has been my pleasure to learn from all of the collaborators.

Special thanks go to Warut Suksompong. At the end of my first semester, Warut visited Xiaohui and gave a talk on *Computing a Small Agreeable Set of Indivisible Items*. I found the talk really fascinating and liked it very much. I was super excited about the rigorously mathematical way of defining fairness, and have been inspired to know more about fair division since then. In addition, during those collaborations and communication, Warut has inspired me with his excellent writing as well as infinite patience, enthusiasm, and humour.

I am grateful for everyone who has made my time at NTU so pleasant. I would like to thank Hao Wang, Guangda Huzhang, Jialin Wang, Shaoyi Wang, Bowen Yuan, Shengxin Liu, Hongao Wang, Chunxue Yang, Zihao Li, and Junjie Luo for having a collegial atmosphere at SPMS-MAS-05-34, and Jareena Tharnnukhroh, Hanh Hong Tran, Yongsheng Liu, Zhengyang Guo, Zhibin Zhang, Nankun Hong, Jiazu Zhou, Yongli Zhou, Xiaoyu Liu, Zhixiang Zhang, and Xin Sun for their help and/or those outings. Thanks to the staff at NTU, especially Liu Shuying Leneatte, Lee Yean Chin, Pearlyn Quek Si Yu, Wendy Tan Hui Min, and Tong Wan Ling, for helping me with all sorts of administrative matters.

Shout out to NTUGSA dance crews: Choreomantics and Jazz All Night. I really had good times with the wonderful dancers at Yunnan Corner. Huge thanks go to Chit Lin Su, Souravik Dutta, Sijie Ji, Yang Liu, Ziqing Peng, and Boyi Su for their friendship and for teaching me how to dance Hip Hop and Jazz. Very special thanks to Ziqing: your smile was so warm and gorgeous and truly healed me when I was depressed.

Thank you to all of my friends for your care and encouragement as well as for sharing my ups and downs over the years: In particular, I would like to thank Yujun Cai for our chats about friendship, (Ph.D.) life, work, vocation, etc., Yuejia Chen for sharing our hobbies, passions, ideas, and opinions with each other, Raven de la Cruz for making my life colourful and for everything that you have done to support me, Muyi He for being an inspiration to me, Shuqi Wang for knowing each other for two decades, Yue Zhang for sharing your knowledge and experiences with me, and Kemin Zhu for taking care of me. I am very fortunate and grateful to have all of you in my life.

Finally, and most importantly, words cannot express how grateful I am to Mom, Yanxue Yin, Dad, Jinglian Lu, and my sister, Ying Lu for their unwavering love, care, patience, support, and encouragement throughout the years. Thank you very much for always being there for me. I love you!

Pursuing a Ph.D. has been one of the most fulfilling experiences of my life, and I feel truly grateful to have had the opportunity to pursue my intellectual curiosity and interests. 那扇窗依然明亮为我守候。[1]

Xinhang
August 2021

[1] *A Foreigner*, Jian Li.

Table of Contents

Chapter 1

Introduction

In the first Politics class of high school, we learned and discussed about *justice*. After that class, we were assigned a short essay as an informal homework to get used to high school life; that is, we had the freedom to discuss whatever we want, as long as it centred around the topic. I started my essay by saying something like "everyone should get a fair share, for instance, each individual gets an equal size of the resources that are to be allocated among them." When I continued working on the essay, I felt confused and did not know what "fair share" really means. There was some vague idea in my mind: different individuals may have different preferences over the resources. In other words, even if it is possible to give each individual exactly the same size of the resources, they might still not be able to have the same level of happiness.[1] I ended my essay by suggesting that if we can make the happiness level of each individual be (roughly) the same, then such an allocation should be fair, in some sense; however, I did not present any concrete method to reach this situation.

How to *fairly* allocate resources among multiple interested agents? Similar question has occurred to many others in a wide variety of real-world applications, such as when assigning rooms to and dividing the rent between several housemates—commonly known as rent division in the literature [90], when dividing marital property in a divorce settlement [56] or inheritance among multiple heirs in an inheritance division, when allocating medical resource such as vaccines, ventilators and other essential medical supplies in healthcare rationing under emergency scenarios [14, 130], and so on. The allocation of scarce resources among interested agents is a problem that arises frequently and plays a major role in our society. We often want to ensure that the allocation that we select is *fair* to all of the agents.

A conceptual challenge is to specify compelling fairness notions—after all, fairness is an abstract idea and has its subjective nature. Notably, in the 1940s, Steinhaus [146] formulated and studied the cake cutting problem: how to fairly divide a cake between multiple participants with potentially different preferences, in which the cake serves as a metaphor for heterogeneous *divisible* resources such as land or time. Steinhaus proposed that in an allocation that involves n participants, every participant should receive a piece of cake which

[1] Now, I know that the solution concept, termed "equitability" by academics, has been studied for over half a century in the literature of fair division.

is worth at least $1/n$ of her value for the cake. This property is known as *proportionality*, one of the most classic and prominent fairness notions. In addition, Steinhaus also showed that a proportional allocation can always be found for any number of agents over any cake.

On the conceptual level, Steinhaus' insight suggested that it is indeed possible to rigorously define fairness in mathematical terms. The literature of *fair division*, which dates back to the design of cake cutting algorithms [79, 146], provides several ways of defining what fair means. For instance, in addition to proportionality, *envy-freeness* requires that each agent weakly prefers her own piece of cake to the piece given to any other agent and *equitability* requires that every agent receives the same utility.

Looking beyond cake cutting, the allocation of heterogeneous *indivisible* goods is deeply practical. This pertains to the allocation of houses, jewellery, electronics, artworks, and many other common items. The fair allocation of indivisible goods has received considerable attention from various research communities, especially in the last few years. We refer to surveys [12, 50, 119, 128] for an overview of recent developments in the area. While all of the aforementioned fairness notions can always be satisfied in the cake cutting setting, this is not the case for indivisible goods allocation.[2] As a result, relaxations have been studied; we defer the discussion of these relaxations to Chapter 3 where we present a literature review on fair division with divisible or indivisible goods.

This dissertation explores the same line of research, with a focus on the existence, computation, and/or approximation of a fairness notion as well as its interaction with other desideratum such as economic efficiency, social welfare, feasibility constraint of an allocation, truthfulness, etc.

1.1 Overview of Thesis Structure and Our Results

We begin in Chapter 2 (Preliminaries) by introducing canonical fair division models, i.e., cake cutting and indivisible goods allocation as well as the recently-introduced mixed goods model. Along the way, we also introduce concepts and definitions that are used across multiple chapters of the dissertation. Then, to present our results, we divide the dissertation into three parts: Part I axiomatically studies fair division of a mixture of divisible and indivisible goods, Part II quantitatively measures trade-offs between fairness and other compelling objectives in indivisible goods allocation, and Part III considers a variant resource allocation setting where public goods are to be allocated. Finally, we conclude the dissertation and provide a summary of interesting open questions for future work in Chapter 9.

Part I: Axiomatic Study of Fairness for Mixed Goods

In this part, we perform axiomatic study of fairness, with a special focus on fair division of *mixed* goods—the set of resources contains both divisible and indivisible goods. We begin in

[2]The canonical example is two agents trying to divide a single valuable good.

Chapter 3 by motivating this line of research. Next, we present our analyses of two intriguing solution concepts—envy-freeness for mixed goods and maximin share guarantee—in Chapters 4 and 5, respectively.

Chapter 4: Envy-Freeness for Mixed Goods Since classic fairness notions such as envy-freeness and envy-freeness up to one good (EF1) cannot be directly applied to the mixed goods setting, Chapter 4 proposes to study a new fairness notion—*envy-freeness for mixed goods (EFM)*, which is a direct generalization of both envy-freeness and EF1 to the mixed goods setting. We show that an EFM allocation always exists for any number of agents with additive valuations. We also devise efficient algorithms to compute an EFM allocation for two agents with general additive valuations as well as for any number of agents with piecewise linear valuations over the divisible goods. Then, we relax the envy-freeness requirement, instead asking for *ϵ-envy-freeness for mixed goods (ϵ-EFM)*, and present an efficient algorithm that finds an ϵ-EFM allocation. Last, we discuss the possibilities and difficulties of combining EFM together with economic efficiency considerations.

Chapter 5: Maximin Share Guarantee In this chapter, we focus on the well-studied fairness notion of *maximin share (MMS) guarantee*. With only indivisible goods, an MMS allocation may not exist, but a constant multiplicative approximate allocation always does. We analyze how the MMS approximation guarantee would be affected when the resources to be allocated also contain divisible goods. In particular, we show that the worst-case MMS approximation guarantee with mixed goods is no worse than that with only indivisible goods. However, there exist problem instances to which adding some divisible resources would strictly decrease the MMS approximation ratios of the instances. On the algorithmic front, we propose a constructive algorithm that will always produce an α-MMS allocation for any number of agents, where α takes values between $1/2$ and 1 and is a monotonically increasing function determined by how agents value the divisible goods relative to their maximin share.

Part II: Fairness Versus Other Desideratum for Indivisible Goods

In this part, we turn our attention to the indivisible goods setting and quantitatively measure trade-offs between fairness and other desiderata such as efficiency (Chapter 6) and connectivity (Chapter 7).

Chapter 6: Price of Fairness Fairness, certainly, is not the only objective in resource allocation. For instance, getting rid of the resources, which leaves all agents empty-handed, definitely does not occur any envy. With that being said, such an allocation is undesirable because it severely hurts economic efficiency. This chapter investigates the efficiency of fair allocations of indivisible goods using the well-studied *price of fairness* concept. Previous work has focused on classical fairness notions such as envy-freeness, proportionality, and equitability. However, these notions cannot always be satisfied for indivisible goods, leading to certain instances being ignored in the analysis. In this chapter, we focus instead on notions with guaranteed existence, including envy-freeness up to one good, balancedness, maximum Nash welfare, and leximin. We also introduce the concept of *strong price of fairness*, which captures the efficiency loss in the worst fair allocation as opposed to that in the best fair allocation as in the price of fairness. We mostly provide tight or asymptotically tight bounds on the worst-case efficiency loss for allocations satisfying these notions, for both the price of fairness and the strong price of fairness.

Chapter 7: Price of Connectivity This chapter deals with the trade-off between fairness and connectivity. That is, we study the allocation of indivisible goods that form an undirected graph and quantify the loss of fairness when we impose a constraint that each agent must receive a connected subgraph. Our focus is on well-studied fairness notions including envy-freeness and maximin share fairness. We introduce the *price of connectivity* to capture the largest gap between the graph-specific and the unconstrained maximin share, and derive bounds on this quantity which are tight for large classes of graphs in the case of two agents and for paths and stars in the general case. For instance, with two agents we show that for biconnected graphs it is possible to obtain at least $3/4$ of the maximin share with connected allocations, while for the remaining graphs the guarantee is at most $1/2$. In addition, we determine the optimal relaxation of envy-freeness that can be obtained with each graph for two agents, and characterize the set of trees and complete bipartite graphs that always admit an allocation satisfying envy-freeness up to one good for three agents. Our work demonstrates several applications of graph-theoretic tools and concepts to fair division problems.

Part III: Other Settings

In this part, we discuss a variant resource allocation setting in Chapter 8.

Chapter 8: Truthful Cake Sharing The classic cake cutting problem concerns the fair allocation of a heterogeneous resource among interested agents. In this chapter, we study a public goods variant of the problem where instead of competing with one another for the cake, the agents all share the same subset of the cake which must be chosen subject to a length constraint. We focus on the design of *truthful* and *fair* mechanisms in the presence of strategic agents who have piecewise uniform utilities over the cake. On the one hand, we show that the leximin solution is truthful and moreover maximizes an egalitarian welfare measure among all truthful and position oblivious mechanisms. On the other hand, we demonstrate that the maximum Nash welfare solution is truthful for two agents but not in general. Our results assume that mechanisms can block each agent from accessing parts that the agent does not claim to desire; we provide an impossibility result when blocking is not allowed.

Chapter 2

Preliminaries

2.1 Cake Cutting Model

Let $[k] \coloneqq \{1, 2, \ldots, k\}$. The model includes a set of agents denoted by $N = [n]$ and a *heterogeneous* divisible good C (or *cake*) represented by the normalized interval $[0, 1]$. A *piece of cake* is a union of finitely many disjoint intervals. Each agent $i \in N$ is endowed with an integrable *density function* $f_i \colon [0, 1] \to \mathbb{R}_{\geq 0}$, which captures how the agent values different parts of the cake. Given a piece of cake $S \subseteq [0, 1]$, agent i's value over S is then defined as $u_i(S) \coloneqq \int_{x \in S} f_i(x) \, \mathrm{d}x$. Denote by $\mathcal{C} = (C_1, C_2, \ldots, C_n)$ the allocation of cake C such that $C_i \cap C_j = \emptyset$, $C = \bigsqcup_{i \in N} C_i$, and agent i gets the piece of cake C_i.

Definition 2.1 (Foley [88], Steinhaus [146], Varian [154])**.** An allocation $\mathcal{C} = (C_1, \ldots, C_n)$ is said to satisfy

- *envy-freeness (EF)* if for any pair of agents $i, j \in N$, $u_i(C_i) \geq u_i(C_j)$;
- *proportionality (PROP)* if for each agent $i \in N$, $u_i(C_i) \geq u_i(C)/n$.

It is easy to see that envy-freeness implies proportionality when the whole cake is required to be allocated.

Query Model We adopt the Robertson-Webb (RW) query model [138], a standard model in cake cutting, to access agents' density functions for the cake. In this model, an algorithm is allowed to interact with the agents via the following two types of queries:

- $\textsc{Eval}_i(x, y)$ asks agent i to evaluate the interval $[x, y]$ and returns the value $u_i([x, y])$;
- $\textsc{Cut}_i(x, \alpha)$ asks agent i to return the leftmost point y such that $u_i([x, y]) = \alpha$, or state that no such point exists.

Structural Density Functions In some other cases, we do not use the RW model, but rather assume that the density functions are structural and provided to us in *full information*.

Definition 2.2. A density function f is said to be

- *piecewise linear* (resp., *piecewise constant*) if interval $[0, 1]$ can be partitioned into a finite number of intervals such that f is linear (resp., constant) on each interval;
- *piecewise uniform* if f is piecewise constant and moreover f takes on some constant $c \in \mathbb{R}_{>0}$ across all desired intervals and zero for all undesired parts.

As is clear from the definition, piecewise linear function generalizes both piecewise uniform and piecewise constant functions, each of which has been considered in several previous fair division works [31, 34, 75].

Homogeneous Cake In some examples or sections, we focus on a special case where the cake to be allocated is *homogeneous*.[1] It means that every agent values all pieces of equal size the same.

2.2 Indivisible Goods Setting

Problem Instance Denote by $N = [n]$ the set of n agents and $M = [m]$ the set of m indivisible goods. A *bundle* is a subset of M. Each agent $i \in N$ is endowed with a *utility function* $u_i \colon 2^M \to \mathbb{R}_{\geq 0}$ such that $u_i(\emptyset) = 0$. For simplicity, we will write $u_i(g)$ to denote $u_i(\{g\})$. A utility function u is said to be

- *monotonic* if $u(M'') \leq u(M')$ for any bundles $M'' \subseteq M' \subseteq M$;
- *additive* if $u(M') = \sum_{g \in M'} u(g)$ for any bundle $M' \subseteq M$.

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the *utility profile* of the agents. We refer to a setting with agents, indivisible goods, and utility functions as an *instance*, denoted by $\langle N, M, \mathcal{U} \rangle$. An allocation $\mathcal{M} = (M_1, M_2, \ldots, M_n)$ is a partition of M into n bundles such that agent i receives bundle M_i; note that M_i might be empty.

Fairness Notion We are now ready to define fairness properties that we consider in this dissertation. To begin, we consider several relaxations of the envy-freeness notion.

Definition 2.3 (Budish [65], Caragiannis et al. [69], Lipton et al. [113])**.** An allocation $\mathcal{M}$ is said to satisfy

- *envy-freeness up to k goods (EFk)*, for a given non-negative integer k, if for every pair of agents $i, j \in N$, there exists a (possibly empty) bundle $M' \subseteq M_j$ with $|M'| \leq k$ such that $u_i(M_i) \geq u_i(M_j \setminus M')$;
- *envy-freeness up to any good (EFX)* if for any pair of agents $i, j \in N$ and any good $g \in M_j$, $u_i(M_i) \geq u_i(M_j \setminus \{g\})$.

[1] An even more restricted case is when the cake is valued the same to all agents. The canonical example of the divisible goods of this special case is *money*.

Algorithm 1: Round-Robin Algorithm

1 Arrange the agents in some arbitrary order.
2 Let the next agent in the order choose her favourite good from the remaining goods.

An EF0 allocation is said to be envy-free. It follows immediately from the definition that envy-freeness implies EFX, which in turn imposes a stronger requirement than EF1. If we do not have to allocate all of the goods, achieving envy-freeness and all of its relaxations is trivial, e.g., by simply not allocating any good. Hence, we will assume that all goods must be allocated when we discuss envy-freeness and its relaxations. In the context of indivisible goods allocation, the gold standard of fairness—envy-freeness—cannot be guaranteed in the simple instance with (at least) two agents and a single valuable good to be divided. In contrast, an EF1 allocation exists for any number of agents with arbitrary monotonic utilities [113]. For EFX, the existence question is still unresolved even if agents have additive utilities [69, 135].

The round-robin algorithm, which is described in Algorithm 1, always computes an EF1 allocation (see for example [69]).

Definition 2.4. An allocation is said to satisfy *round-robin* if it is the result of applying the algorithm with some ordering of the agents.[2]

Our next fairness notion is the maximin share guarantee proposed by Budish [65].

Definition 2.5. Given a set of goods M and the number of agents n, the *maximin share (MMS)* of agent i is defined as

$$\text{MMS}_i(n, M) = \max_{(P_1, P_2, \ldots, P_n)} \min_{j \in [n]} u_i(P_j),$$

where the maximum is taken over all partitions $(P_1, P_2, \ldots, P_n)$ of M. A partition for which the maximum is attained is called an *MMS partition* for agent i.

For notational convenience, we will simply write MMS_i when parameters n and M are clear from the context.

Definition 2.6 (α-MMS)**.** An allocation $\mathcal{M}$ is said to satisfy the *α-approximate maximin share guarantee (α-MMS)*, for some $\alpha \in [0, 1]$, if for every agent $i \in N$,

$$u_i(M_i) \geq \alpha \cdot \text{MMS}_i(n, M).$$

We say a 1-MMS allocation satisfies the maximin share guarantee and write MMS as a shorthand for 1-MMS.

[2]In case there are ties between goods, we may assume worst-case tie breaking, since it is possible to obtain an instance with infinitesimal difference in welfare and any desired tie-breaking between goods by slightly perturbing the utilities.

We now define balancedness, which means that the goods are spread out among the agents as much as possible. Balancedness and similar cardinality constraints have been considered in recent work [46]. In addition to satisfying EF1, an allocation produced by the round-robin algorithm is also balanced.

Definition 2.7. An allocation is said to be *balanced* if $||M_i| - |M_j|| \leq 1$ for any $i, j \in N$.

Welfare Maximizer and Economic Efficiency Now, we define a number of welfare maximizers considered in this dissertation.

Definition 2.8. The *(utilitarian) social welfare* of an allocation $\mathcal{M}$ is defined as $\mathsf{SW}(\mathcal{M}) := \sum_{i=1}^{n} u_i(M_i)$. The optimal social welfare for an instance $I = \langle N, M, \mathcal{U} \rangle$, denoted by $\mathsf{OPT}(I)$, is the maximum social welfare over all allocations for this instance.

Definition 2.9 (MNW)**.** The *Nash welfare* of an allocation $\mathcal{M}$ is defined as $\prod_{i \in N} u_i(M_i)$. An allocation is said to be a *maximum Nash welfare (MNW)* allocation if it has the maximum Nash welfare among all allocations.[3]

Definition 2.10 (MEW)**.** The *egalitarian welfare* of an allocation $\mathcal{M}$ is $\min_{i \in N} u_i(M_i)$. An allocation is said to be a *maximum egalitarian welfare (MEW)* allocation if it has the maximum egalitarian welfare among all allocations.

Definition 2.11. An allocation is said to be *leximin* if it maximizes the lowest utility (i.e., the egalitarian welfare), and, among all such allocations, maximizes the second lowest utility, and so on.

Finally, we define Pareto optimality, a fundamental property in the context of resource allocation.

Definition 2.12 (PO)**.** Given an allocation $\mathcal{M}$, another allocation $(M'_1, M'_2, \ldots, M'_n)$ is said to be a *Pareto improvement* if $u_i(M'_i) \geq u_i(M_i)$ for all agents $i \in [n]$, with at least one strict inequality. An allocation is said to be *Pareto optimal (PO)* if it does not admit a Pareto improvement.

Caragiannis et al. [69] showed that an MNW allocation always satisfies EF1 and Pareto optimality. It is clear from the definition that any leximin allocation is Pareto optimal and maximizes egalitarian welfare. The problem of computing an MEW allocation has been studied by Bezáková and Dani [42] and Bansal and Sviridenko [26]. Leximin allocations were studied by Bogomolnaia and Moulin [47] and shown to be applicable in practice by Kurokawa et al. [107].

[3] In the case where the maximum Nash welfare is 0, an allocation is an MNW allocation if it gives positive utility to a set of agents of maximal size and moreover maximizes the product of utilities of the agents in that set.

2.3 Mixed Goods Model

In this section, we consider a resource allocation setting with both divisible and indivisible goods (mixed goods for short) to be allocated. Denote by $N = [n]$ the set of n agents, $M = [m]$ the set of m indivisible goods, and $\{D_1, D_2, \ldots, D_\ell\}$ the set of ℓ cakes. Since the fairness notions we investigate in this dissertation do not distinguish pieces from different cakes, without loss of generality, we assume that each cake D_i is represented by the interval $\left[\frac{i-1}{\ell}, \frac{i}{\ell}\right]$ and use a single cake $C = [0, 1]$ to represent the union of all cakes.[4] An *allocation* of the mixed goods is denoted by $\mathcal{A} = (A_1, A_2, \ldots, A_n)$ where $A_i = M_i \cup C_i$ is the *bundle* allocated to agent i. Agent i's utility for the allocation is then defined as $u_i(A_i) \coloneqq u_i(M_i) + u_i(C_i)$.

Now, we are ready to define the fairness notions considered for mixed goods. Neither envy-freeness nor EF1 alone is a suitable definition; thus, we introduce the following fairness notion.

Definition 2.13 (EFM)**.** An allocation $\mathcal{A}$ is said to satisfy *envy-freeness for mixed goods (EFM)* if for any pair of agents $i, j \in N$,

- if agent j's bundle consists of only indivisible goods, there exists $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{g\})$;
- otherwise, $u_i(A_i) \geq u_i(A_j)$.

It is easy to see that when the goods are all divisible, EFM reduces to envy-freeness and when the goods are all indivisible, EFM reduces to EF1. Therefore, EFM naturally generalizes both envy-freeness and EF1 to the mixed goods setting.

Next, we define a relaxation of the EFM notion. Our definition only relaxes the envy-free condition when the bundle contains some divisible goods; the EF1 condition is not relaxed.

Definition 2.14 (ϵ-EFM)**.** An allocation $\mathcal{A}$ is said to satisfy *ϵ-envy-freeness for mixed goods (ϵ-EFM)* if for any pair of agents $i, j \in N$,

- if agent j's bundle consists of only indivisible goods, there exists $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{g\})$;
- otherwise, $u_i(A_i) \geq u_i(A_j) - \epsilon$.

We now define the maximin share guarantee (Definition 2.5) and its relaxations in the mixed goods setting.

Definition 2.15. Let $\Pi_k(M \cup C)$ be the set of k-partitions of $M \cup C$. Define the k-maximin share of agent i as

$$\text{MMS}_i(k, M \cup C) = \max_{(P_1, P_2, \ldots, P_k) \in \Pi_k(M \cup C)} \min_{j \in [k]} u_i(P_j).$$

[4] We assume that agents' valuation functions over the cakes are non-atomic, and thus we can view two consecutive cakes as disjoint even if they intersect at one boundary point.

The *maximin share (MMS)* of agent i is $\text{MMS}_i(n, M \cup C)$. Every partition in

$$\arg \max_{(P_1, P_2, \dots, P_n) \in \Pi_n(M \cup C)} \min_{j \in [n]} u_i(P_j)$$

is called an *MMS partition* for agent i.

Again, when parameters n, M, and C are clear from the context, we will simply write MMS_i as agent i's maximin share for convenience.

Definition 2.16 (α-MMS)**.** An allocation $\mathcal{A}$ is said to satisfy the *α-approximate maximin share guarantee (α-MMS)*, for some $\alpha \in [0, 1]$, if for every agent $i \in N$,

$$u_i(A_i) \geq \alpha \cdot \text{MMS}_i(n, M \cup C).$$

Part I

Axiomatic Study of Fairness for Mixed Goods

Chapter 3

Introduction

Fair division concerns the problem of allocating scarce resources among interested agents, with the objective of finding an allocation that is fair to all agents involved. Initiated by Steinhaus [146], the study of fair division has since been attracting interest from various disciplines for decades, including among others, mathematics, politics, economics, and computer science [55, 60, 82, 127, 128, 134, 138, 139, 153, 160]. Moreover, due to its subjective nature, a plethora of fairness notions have been proposed and investigated in different resource allocation scenarios. Two of the most prominent fairness notions in the literature are *proportionality* and *envy-freeness*, introduced by Steinhaus [146] and Foley [88], respectively. An allocation is said to be proportional if every agent receives a bundle which is worth at least $1/n$ of her value for the entire set of goods, and envy-free if each agent weakly prefers her own bundle to any other bundle in the allocation. It follows from the definition that when goods are all allocated, envy-freeness implies proportionality.

The literature of fair division can be divided into two classes, categorized by the type of the resources to be allocated. The first class assumes the resources to be heterogeneous and infinitely divisible. The corresponding problem is commonly known as *cake cutting*, with the cake serving as a metaphor for the heterogeneous divisible resource. An envy-free allocation with divisible resources always exists (see, e.g., Alon [4]) and can be found via a discrete and bounded protocol [15]; a proportional allocation can always be found efficiently for any number of agents over any divisible good [146].

The second class considers *indivisible resource allocation*, i.e., fair division of heterogeneous and *indivisible* resources. When goods are indivisible, neither envy-freeness nor proportionality can always be fulfilled, e.g., when two agents try to divide a single valuable good. In order to circumvent this issue, relaxations of both notions have been studied. Envy-freeness is often relaxed to *envy-freeness up to one good (EF1)* [65, 113], which requires that it is possible to eliminate any envy one agent has towards another agent by removing some good from the latter's bundle. An EF1 allocation exists for any number of agents with arbitrary monotonic utilities and can be found efficiently [113]. Likewise, Budish [65] introduced a natural alternative to the proportionality that also works for indivisible goods, known as the *maximin share guarantee*. In that definition, the *maximin share (MMS)* of an agent is defined as the largest value that the agent can guarantee for herself if she is allowed to partition goods into n bundles and always receives the least desirable bundle. An allocation that gives every agent her maximin share—said to satisfy the MMS guarantee—does not always exist when there are at least three agents and utilities are additive, but a constant multiplicative approximation can be obtained [108].[1]

The vast majority of the fair division literature assumes that the resources either are completely divisible, or consist of only indivisible goods.[2] This is, however, not always the case in many real-world scenarios. In a divorce settlement or an inheritance division, for instance, the goods to be divided may contain divisible goods such as land and money, as well as indivisible goods like jewellery, houses, artworks, and many other common items. What fairness notion should one adopt when allocating mixed types of resources?

Although both envy-freeness and EF1 work well in their respective settings, neither can be directly applied to the mixed goods model. On the one hand, an envy-free allocation may not exist, when, for example, all goods are indivisible. On the other hand, the EF1 notion, when interpreted as that each agent does not envy another agent after removing one indivisible good from the latter agent's bundle, may also produce unfair allocations. Consider the example where an indivisible good and a cake are both equally valued by two agents; dividing the cake in half and then giving the indivisible good to one of the agents is EF1 but arguably unfair. Thus, dividing mixed types of resources calls for new fairness notions that could unify envy-freeness and EF1 together in a natural and non-trivial way. In this dissertation, we provide such a treatment. Specifically, in Chapter 4, we introduce and study a notion called *envy-freeness for mixed goods (EFM)*, which is a generalization of envy-freeness and EF1 in the mixed goods setting.

EFM is an envy-based fairness notion; we then turn our attention to a fair-share-based notion and focus on investigating the maximin share guarantee. While previous works has been mainly studying the maximin share guarantee in the context of indivisible resource allocation (see Section 3.1), it is actually a very well-defined notion when dividing mixed types of resources. Comparing with EFM studied in Chapter 4, the maximin share guarantee can be directly applied to the mixed goods setting without any modification.[3] This allows us to compare the results of MMS for mixed goods directly to those for indivisible goods. To be more specific, in Chapter 5, we study the existence, approximation, and computation of maximin share guarantee with mixed goods.

[1] An MMS allocation always exists when there are two agents [49].

[2] A notable exception to this is the work of Rubchinsky [140], who considered the fair division problem between two agents with both indivisible and homogeneous divisible items, and introduced three fairness notions with computationally efficient algorithms for finding them.

[3] With only divisible goods, envy-freeness implies proportionality but not vice versa. With only indivisible goods, the notions of EF1 and MMS do not imply each other [69]. Since EFM generalizes both envy-freeness and EF1 to the setting of mixed goods and EFM reduces to EF1 with only indivisible goods, it is obvious that neither EFM nor MMS implies the other.

3.1 Related Work

As mentioned earlier, most previous works in fair division are from two categories based on whether the resources to be allocated are divisible or indivisible.

Cake Cutting When the resources are divisible, the existence of an envy-free allocation is guaranteed [79, 111], even with only $n-1$ cuts [147, 148]. Brams and Taylor [54] gave the first finite (but unbounded) envy-free protocol for any number of agents. Recently, Aziz and Mackenzie [16] gave the first bounded protocol for computing an envy-free allocation with four agents and their follow-up work extended the result to any number of agents [15]. Besides envy-freeness, other classic fairness notions include proportionality and *equitability*, both of which have been studied extensively [70, 79–81, 83, 136].

Indivisible Resource Allocation When the resources are indivisible, none of the aforementioned fairness notions is guaranteed to exist, and thus relaxations have been considered. Among other notions, these include envy-freeness up to one good (EF1), envy-freeness up to any good (EFX), and maximin share (MMS) guarantee [65, 69, 113]; see Chapter 2 for their definitions. The existence of EFX allocations is still open [135], except for several special cases [8, 39, 68, 72–74, 115, 132]. That said, we can further strengthen our EFM notion in the sense that agents use the EFX condition to compare their bundles to an agent who only receives indivisible goods; see our discussion in Section 4.3.

The MMS guarantee nicely captures the local measure of fairness even when the goods to be allocated are indivisible. Since the seminal work of Kurokawa et al. [108], many subsequent works have been carried out on the improvements of MMS approximation guarantee, design of simpler algorithms and so on when allocating indivisible goods [7, 27, 87, 91–93].[4] The MMS guarantee has also been adopted as the fairness solution concept in several practical applications [65, 95].

Slightly deviating from this line of work, MMS allocations of indivisible resources have also been extensively studied in a variety of fair division settings, including (but not limited to) for agents with unequal entitlements [86] or in different groups [149], for goods forming an undirected graph [51, 114], for allocations under matroid constraints [96] or in conjunction with economic efficiency [102], as well as in the context of *chore division*, where chores refer to negatively valued items [19, 22, 23, 101].

In addition, several works studied fair division with the assumption that (indivisible) resources can be shared among agents. For instance, the adjusted-winner (AW) procedure ensures that at most one good must be split in a fair and (economically) efficient division between two agents [55]. Focus has also been given to obtain a fair and efficient division with minimum number of objects shared between two or more agents [126, 141]. All of the works discussed in this paragraph implicitly assumed that the indivisible items are *homogeneous*.

[4]The approximation guarantee for MMS was improved to $3/4$ by Ghodsi et al. [93] and the currently best-known ratio is $\frac{3}{4} + \frac{1}{12n}$ due to Garg and Taki [91].

Money Involved A related line of research incorporates money into the fair division of indivisible goods, with the goal of finding envy-free allocations [3, 13, 64, 66, 94, 97, 105, 120, 122]. In a recent work, Halpern and Shah [97] bounded the amount of money needed to achieve envy-freeness for agents with additive valuations, assuming that the value of each agent for each good is at most 1. Their result were further improved by Brustle et al. [64]. Aziz [13] then studied the problem of using monetary transfers to achieve envy-freeness and equitability simultaneously. On the other hand, Caragiannis and Ioannidis [66] studied the optimization problem of computing the minimum amount of money needed to obtain envy-freeness given an indivisible instance and showed both approximation and hardness results. Then, following the work of Halpern and Shah [97], but in a more restricted setting where agents have submodular dichotomous valuations,[5] Goko et al. [94] designed a truthful mechanism that achieves envy-freeness by subsidizing each agent with at most 1. This part of the dissertation (Chapters 4 and 5) and these works indeed share some similarities in that we consider a mixed goods model which contains a cake and money can be viewed as a special type of cake which is homogeneous and valued the same across all agents. However, a major difference between Part I and theirs is that we have different objectives, which also makes our results and theirs are not comparable. Briefly speaking, those works focused on finding envy-free allocations with the help of a sufficient amount of money, which is different from our goal in the sense that our method could also be used even in cases where the money is insufficient. Put differently, those works aim to *determine* the amount of money needed to be added to the set of indivisible goods such that an envy-free allocation can be obtained; their results, however, do not specify what to do when there is not enough money as they required. In this part of the dissertation, we focus on instances in which the amount of indivisible goods and cake are both *fixed*, and regardless of whether the cake is enough to guarantee an envy-free (resp., MMS) allocation, we aim to find an EFM allocation in Chapter 4 (resp., a reasonably fair allocation with MMS approximation guarantee in Chapter 5).

Also related is *rent division*; see, for example, [3, 9, 10, 90, 137, 148, 152, 155] and references therein for an overview. Its cardinal utility version can be viewed as a special case of the mixed setting where one wants to allocate (indivisible) rooms and (divisible) rent among agents.

Mixed Resources Model After the publication of our work [36], Bhaskar et al. [43] showed that EFM allocations always exist in a setting where we divide doubly monotone indivisible items[6] and a bad cake (i.e., non-positively valued heterogeneous divisible item) as well as in some special cases of the setting where indivisible chores and a cake are to be divided.

[5] A utility function u is said to be *dichotomous* if the marginal value of any good is either 0 or 1; *submodular* if $u(S \cup \{g\}) - u(S) \geq u(T \cup \{g\}) - u(T)$ for every pair of bundles $S \subseteq T \subseteq m$ and every good $g \in M$; *submodular dichotomous* if it is both submodular and dichotomous.

[6] Each agent i can partition the set of items into goods and chores, that is, adding a good does not decrease the agent's utility for her bundle and adding a chore does not increase the agent's utility for her bundle.

Chapter 4

Envy-Freeness for Mixed Goods †

4.1 Introduction

While neither envy-freeness nor EF1 is a suitable fairness notion in the mixed goods setting, a possibly tempting solution is to divide the divisible and indivisible resources using envy-free and EF1 protocols separately and independently, and then combine the two allocations together. This approach, however, also has problems. Consider the example where two agents need to divide a cake and an indivisible good. EF1 requires to allocate the indivisible good to one of the agents, say agent 1. However, if we then divide the cake using an arbitrary envy-free allocation, the overall allocation is arguably unfair to agent 2. As a matter of fact, if the whole cake is valued less than the indivisible good, it would make more sense to allocate the entire cake to agent 2. When the cake is valued more than the indivisible good, it is still a fairer solution to allocate more cake to agent 2 in order to compensate her disadvantage in the indivisible goods allocation. Our discussion demonstrates that it is not straightforward to generalize envy-freeness and EF1 to the mixed goods setting.

In this chapter, we introduce and study the notion of *envy-freeness for mixed goods (EFM)*, which naturally combines envy-freeness and EF1 together in and works well for the mixed goods model defined in Section 2.3. Intuitively, EFM requires that for each agent, if her bundle consists of only indivisible goods, other agents will compare their bundles to hers using the EF1 criterion; however, if her bundle contains *any* positive amount of divisible resources, other agents will compare their bundles to hers using the stricter envy-free condition. Our definition of EFM generalizes both envy-freeness and EF1 to the mixed goods setting and strikes a natural balance between them.

In Section 4.2, we show that with mixed goods, an EFM allocation always exists for any number of agents with additive valuations. Our proof is constructive and gives an algorithm for computing an EFM allocation. The algorithm requires an oracle to compute a *perfect allocation* in cake cutting and can find an EFM allocation in a polynomial number of steps. In addition, in Section 4.3, we present two algorithms that could compute an EFM allocation for two special cases without using the oracle: (1) two agents with general additive valuations

† This chapter has been published in a paper by Bei, Li, Liu, Liu, and Lu [36].

in the RW model, and (2) any number of agents with piecewise linear valuations for the cake.

While it is still unclear to us whether in general an EFM allocation can be computed in a finite number of steps in the RW model, in Section 4.4, we turn our attention to *ϵ-envy-freeness for mixed goods (ϵ-EFM)*, a relaxation of the EFM notion. We devise an algorithm to compute an ϵ-EFM allocation in the RW model with running time *polynomial* in the number of agents n, the number of indivisible goods m, and $1/\epsilon$ as well as with query complexity *polynomial* in n and $1/\epsilon$. We note that this algorithm does not require the perfect allocation oracle. This result is appealing in particular due to its polynomial running time and query complexity. A bounded exact EFM protocol, even if exists, is likely to require a large number of queries and cuts as in the special case of cake cutting, EFM reduces to envy-freeness, for which the best-known protocol of Aziz and Mackenzie [15] has a very high query complexity, i.e., a tower of exponents of n. Thus, if one is willing to allow a small margin of errors, such an allocation could be found much more efficiently.

Last but not least, we discuss EFM in conjunction with economic efficiency (Section 4.5). In particular, we show that EFM and PO are incompatible. Thus, we propose a weaker version of EFM and discuss the possibilities and difficulties in combining it with PO.

4.2 Existence

Although EFM naturally generalizes both EF and EF1, it is not straightforward if an EFM allocation would always exist with mixed goods. In this section, we show via a constructive algorithm that with mixed goods and any number of agents, an EFM allocation always exists.

We begin with the following concepts which will be helpful for our algorithm and proofs.

Perfect Allocation Intuitively, a perfect allocation in cake cutting divides the cake into k pieces such that every agent in N values these k pieces equally.

Definition 4.1 (Perfect allocation)**.** A partition $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ of cake C is said to be *perfect* if for all $i \in N, j \in [k]$, we have $u_i(C_j) = u_i(C)/k$.

Alon [4] showed that a perfect allocation always exists for any number of agents and any k. We will assume that our algorithm is equipped with the $\textsc{PerfectAlloc}(C, k, N)$ oracle which returns us a perfect allocation for any k and cake C among all of the agents in N.

Envy Graph We use an *envy graph* to capture envy relations between the agents.

Definition 4.2 (Envy graph)**.** Given an allocation $\mathcal{A}$, its corresponding *envy graph* $G = (N, E_{\text{envy}} \cup E_{\text{eq}})$ is a directed graph, where each vertex represents an agent, and E_{envy} and E_{eq} consist of the following two types of edges, respectively:

- Envy edge: $i \xrightarrow{\text{ENVY}} j$ if $u_i(A_i) < u_i(A_j)$;
- Equality edge: $i \xrightarrow{\text{EQ}} j$ if $u_i(A_i) = u_i(A_j)$.

A cycle in an envy graph is called an *envy cycle* if it contains at least one envy edge. The concepts of envy edge and equality edge were also used in [105, 113].

We next define an *addable set*, which corresponds to a specific set of agents.

Definition 4.3 (Addable set)**.** Given an envy graph, a non-empty set of agents $S \subseteq N$ forms an *addable set* if

- there is no envy edge between any pair of agents in S;
- there exists neither an envy edge nor an equality edge from $N \setminus S$ to S.

Moreover, an addable set $S \subseteq N$ is called a *maximal addable set* if there does not exist any other addable set $S' \subseteq N$ such that $S \subsetneq S'$. The following lemma shows the uniqueness of the maximal addable set in an envy graph.

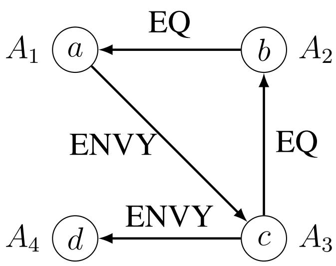
Lemma 4.4. *Given an envy graph, the maximal addable set, if exists, is unique. Moreover, we can find it or decide that none exists in $O(n^3)$ time.*

Proof. Suppose for contradiction that there exist two distinct maximal addable sets S_1 and S_2 in the given envy graph. First, there exists neither an envy edge nor an equality edge from $N \setminus (S_1 \cup S_2)$ to $S_1 \cup S_2$ since, otherwise, either S_1 or S_2 is not an addable set. We next argue that there is no envy edge between agents in $S_1 \cup S_2$. Clearly, according to Definition 4.3, there is no envy edge within each of S_1 and S_2. The envy edges between S_1 and S_2 also cannot exist because there are no envy edges coming from outside of S_1 or S_2 into any of them. Thus, $S_1 \cup S_2$ is also an addable set, contradicting the maximality of S_1 and S_2.

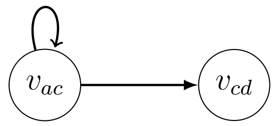
Now, we show how to find the unique maximal addable set or decide its non-existence in $O(n^3)$ time: for each j which has an incoming envy edge, let R_j be the collection of vertices (including j) that are reachable by j via the union of envy edges and equality edges, and let $S = N \setminus \bigcup_j R_j$. We will show that an addable set does not exist in the envy graph if $S = \emptyset$. Otherwise, S is the unique maximal addable set. First, S is an addable set because any agent in S does not have any incoming envy edge and is not reachable via the union of envy edges and equality edges from any other agent with an incoming envy edge. In addition, S is maximal because any agent in $\bigcup_j R_j$ cannot be in any addable set. Such S can be found in $O(n^3)$ time because it takes $O(n)$ time to check if an agent has an incoming edge, and for any agent j who has an incoming envy edge, it then takes $O(n^2)$ time to construct R_j via, for example, breadth-first search (BFS). $\square$

Intuitively, agents in addable set S can be allocated some cake without creating new envy because each agent in $N \setminus S$ values her own bundle strictly more than the bundles of agents in S. Our next result characterizes the relation between the addable set and the envy cycle.

Lemma 4.5. *Any envy graph $G = (N, E_{envy} \cup E_{eq})$ that does not have any envy cycle must have an addable set.*

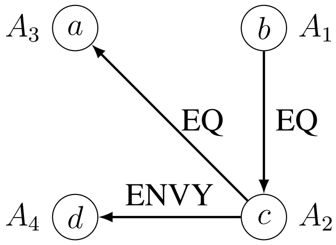


(a) Envy graph G.

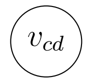


(b) The corresponding G'.

Figure 4.1: Figure 4.1a shows an envy graph G with four vertices (i.e., agents) a, b, c, d. The bundle each agent gets is labelled with A_1, A_2, A_3, A_4. We show in Figure 4.1b its corresponding G'. Here, G has an envy cycle (involving vertices a, b, c) but no addable set.



(a) Envy graph G.



(b) The corresponding G'.

Figure 4.2: After rotating the bundles along the envy cycle in Figure 4.1a, we obtain the envy graph in Figure 4.2a. The corresponding G' is then shown in Figure 4.2b. In this example, G has addable sets $\{b\}$, $\{b, c\}$ and $\{a, b, c\}$ but no envy cycle.

Proof. We assume without loss of generality that $E_{\text{envy}} \neq \emptyset$, otherwise N itself is an addable set. Now, we construct graph $G' = (N', E')$ from G as follows. Each envy edge $i \xrightarrow{\text{ENVY}} j$ in G corresponds to a vertex v_{ij} in G'. For two envy edges $i \xrightarrow{\text{ENVY}} j$ and $i' \xrightarrow{\text{ENVY}} j'$ in G, if there exists a path from j to i', we construct an edge $v_{ij} \to v_{i'j'}$ in G'. Note that, if there is an envy edge $i \xrightarrow{\text{ENVY}} j$ and a path from j to i in G, there will be a self-loop $v_{ij} \to v_{ij}$ in G'. We illustrate this transformation using two examples in Figures 4.1 and 4.2.

A cycle in G' implies an envy cycle in G, and thus by the assumption that there is no envy cycle in G, G' must be acyclic. Then there must exist a vertex $v_{ij} \in N'$ which is not reachable by any other vertices in G'. Because v_{ij} corresponds to the envy edge $i \xrightarrow{\text{ENVY}} j$ in G and v_{ij} cannot be reached by any vertices in G', the vertex i is also not reachable by any j' which is pointed by an envy edge. We note that, however, vertex i may be reachable by other vertices via only equality edges. Thus, we need to not only include agent i in the addable set but also those agents who are able to reach i via equality edges.

Let S be the set containing agent i and all other agents who can reach i in G via equality edges. We will show that S is an addable set. First, S is non-empty as it contains at least agent i. Second, by our construction, there is no envy edge between agents in S. Third, recall that agent i is not reachable by any j' which is pointed by an envy edge; thus, S is also not pointed by any envy edge. Last, S is also not pointed by any equality edge by our construction of S. Therefore, according to Definition 4.3, S must be an addable set. $\square$

Algorithm 2: EFM Algorithm

Input: Agents N, mixed goods $M \cup C$, as well as utility and density functions.

1 Find an arbitrary EF1 allocation $(A_1, A_2, \ldots, A_n)$ of M to n agents.
2 Construct an envy graph $G = (N, E_{\text{envy}} \cup E_{\text{eq}})$ accordingly.
3 **while** $C \neq \emptyset$ **do** // Allocate divisible goods
4   **if** *there exists an addable set in* G **then** // Cake-adding phase
5   Let S be the maximal addable set.
6   **if** $S = N$ **then**
7   Find an envy-free allocation $(C_1, C_2, \ldots, C_n)$ of C.
8             $C \leftarrow \emptyset$
9   Add C_i to bundle A_i for all $i \in N$.
10   **else**
11             $\delta_i \leftarrow \min_{j \in S}(u_i(A_i) - u_i(A_j))$ for each $i \in N \setminus S$.
12   **if** $u_i(C) \leq |S| \cdot \delta_i$ *holds for each* $i \in N \setminus S$ **then**
13                 $C' \leftarrow C, C \leftarrow \emptyset$
14   **else**
15   Suppose w.l.o.g. that $C = [a, b]$. For each agent $i \in N \setminus S$, if $u_i([a, b]) \geq |S| \cdot \delta_i$, let x_i be a point such that $u_i([a, x_i]) = |S| \cdot \delta_i$; otherwise, let $x_i = b$.
16                 $i^* \leftarrow \arg\min_{i \in N \setminus S} x_i$
17                 $C' \leftarrow [a, x_{i^*}], C \leftarrow C \setminus C'$
18   Let $(C_1, C_2, \ldots, C_k) = \textsc{PerfectAlloc}(C', k, N)$ where $k = |S|$.
19   Add C_i to the bundle of the i-th agent in S.
20   Update envy graph G accordingly.
21   **else** // Envy-cycle-elimination phase
22   Let T be an envy cycle in envy graph G.
23   For each agent $j \in T$, give her bundle to agent i who points to agent j in T.
24   Update envy graph G accordingly.
25 **return** $(A_1, A_2, \ldots, A_n)$

4.2.1 The Algorithm

The pseudocode to compute an EFM allocation is presented as Algorithm 2. In general, our algorithm always maintains a partial allocation that is EFM. Then, we repeatedly and carefully add resources to the partial allocation, until all resources are allocated. We start with an EF1 allocation of only indivisible goods to all agents in line 1, and construct the corresponding envy graph in line 2. Then, our algorithm executes in rounds (lines 3 to 24). In each round, we try to distribute some cake to the partial allocation while ensuring the partial allocation to be EFM. Such a distribution needs to be done carefully because once an agent is allocated with a positive amount of cake, the fairness condition with regard to her bundle changes from EF1 to envy-freeness, which is more demanding. We repeat the process until the whole cake is allocated.

In each round of Algorithm 2, depending on whether there is an addable set that can be given some cake in line 4, we execute either an *cake-adding phase* (lines 4 to 20) or an *envy-cycle-elimination phase* (lines 21 to 24).

- In the cake-adding phase, we have a maximal addable set S. By its definition, each agent in $N \setminus S$ values her own bundle strictly more than the bundles of agents in S. Thus there is room to allocate some cake C' to agents in S. We carefully select C' to be allocated to S such that it does not create any new envy among the agents. To achieve this, we choose a piece of cake $C' \subseteq C$ to be *perfectly* allocated to S in lines 11 to 17 so that no agent in N will envy agents in S after distributing C' in lines 18 and 19. More specifically, for each agent $i \in N \setminus S$, we determine in line 11 the largest value δ_i to be added to any agent in S such that i would still not envy any agent in S. Then, the way we decide x_{i^*} in lines 16 and 17 ensures that for all agents $i \in N \setminus S$, $v_i([a, x_{i^*}]) \leq |S| \cdot \delta_i$. Next, in line 18, cake $C' = [a, x_{i^*}]$ is divided into $|S|$ pieces that are valued equally by all agents in N. This is to ensure that no agent $i \in N \setminus S$ values any piece more than δ_i.

- In the envy-cycle-elimination phase, i.e., when there does not exist any addable set, we show that in this case there must exist an envy cycle T in the current envy graph. We can then apply the envy-cycle-elimination technique to reduce some existing envy from the allocation by rearranging the bundles along T. More specifically, for each agent $j \in T$, we give agent j's bundle to agent i who points to her in T (line 23).

When all goods are indivisible, our algorithm performs lines 1 and 2 and terminates with an EF1 allocation (which is also EFM). When the whole good is a cake, the algorithm goes directly to line 7 and ends with an envy-free allocation of the cake, which is again EFM.

4.2.2 Analysis

Our main result for the EFM allocation is as follows:

Theorem 4.6. *An EFM allocation always exists for any number of agents with additive valuations and can be found by Algorithm 2 in polynomial time with $O(n^4)$ Robertson-Webb queries and $O(n^3)$ calls to the* PERFECTALLOC *oracle.*

Invariants

To prove Theorem 4.6, we first show that the following invariants are maintained by Algorithm 2 during its run.

A1. In each round there is either an addable set for the cake-adding phase or an envy cycle for the envy-cycle-elimination phase.

A2. The partial allocation is always EFM.

Lemma 4.7. *Invariant A1 holds during the algorithm's run.*

Proof. This invariant is implied directly by Lemma 4.5. □

Lemma 4.8. *Invariant A2 holds during the algorithm's run.*

Proof. The partial allocation is clearly EFM after line 1. Then the allocation is updated in three places in the algorithm: lines 9 and 19 in the cake-adding phase and line 23 in the envy-cycle-elimination phase. Given a partial allocation that is EFM, we will show that each of these updates maintains the EFM condition.

First, when we have $S = N$ in line 6, i.e., the addable set S consists of all n agents, the current envy graph does not contain any envy edge due to the definition of addable set (Definition 4.3). This implies that current partial allocation actually is envy-free. Because all valuation functions are additive, adding another envy-free allocation on top of it in line 9 results in an envy-free and, hence, EFM allocation.

We next consider line 19 in the cake-adding phase where a piece of cake is added to the addable set S. In order to maintain an EFM partial allocation, we need to ensure that this process does not introduce any new envy towards agents in S. Since we add a perfect allocation in lines 18 and 19, envy will not emerge among agents in S. We also carefully choose the amount of cake to be allocated in lines 12 to 17 such that each agent in $N \setminus S$ weakly prefers her bundle to any bundles that belong to agents in S. To achieve this, we choose a piece of cake $C' \subseteq C$ to be *perfectly* allocated to S in lines 11 to 17 so that no agent in N will envy agents in S after distributing C' in lines 18 and 19. More specifically, for each agent $i \in N \setminus S$, we determine in line 11 the largest value δ_i to be added to any agent in S such that i would still not envy any agent in S. Then, the way we decide x_{i^*} in lines 16 and 17 ensures that for all agents $i \in N \setminus S$, $v_i([a, x_{i^*}]) \leq |S| \cdot \delta_i$. Next, in line 18, cake $C' = [a, x_{i^*}]$ is divided into $|S|$ pieces that are valued equally by all agents in N. This is to ensure that for any piece of cake C_j allocated to $j \in S$, we have $u_i(C_j) \leq \delta_i$ for all $i \in N \setminus S$. Thus, agents in $N \setminus S$ continues to not envy agents in S in line 19.

Finally, in the envy-cycle-elimination phase, line 23 eliminates envy edges by rearranging the partial allocation within the envy cycle T. Since each agent in T is weakly better off, the partial allocation remains EFM. For agents in $N \setminus T$, rearranging the partial allocation that is EFM will not make EFM infeasible. The conclusion follows. □

Correctness

Lemma 4.9. *Algorithm 2 always returns an EFM allocation upon termination.*

Proof. By Invariant A2, it suffices to prove that all goods are allocated when Algorithm 2 terminates. All indivisible goods are allocated in line 1. Then the `while` loop (lines 3 to 24) terminates only when the cake is also fully allocated, as desired. □

Termination and Time Complexity

We use the number of envy edges in the envy graph and the size of the maximal addable set as a potential function to bound the running time of this algorithm.

Lemma 4.10. *After the algorithm completes a cake-adding phase, the number of envy edges never increases. In addition, if the piece of cake to be allocated is not the whole remaining cake, either (a) the number of envy edges strictly decreases, or (b) the size of the maximal addable set strictly decreases or an addable set no longer exists.*

Proof. By Lemma 4.8, the partial allocation is always EFM after a cake-adding phase. In a cake-adding phase, some positive amount of cake is added to every agent in S. This means that after this phase, there would never be any envy edge between agents in S or from $N \setminus S$ to S. The bundles of agents in $N \setminus S$ remains the same, hence the set of edges among agents in $N \setminus S$ remains unchanged. Finally, since only agents in S are allocated new resources in the cake-adding phase, no new envy edge will be introduced from S to $N \setminus S$. This proves the first part of Lemma 4.10.

For the second part, we only study the situation when the piece of cake to be allocated to agents in S is not the whole remaining cake (lines 14 to 17). Note that the number of envy edges will never increase after a cake-adding phase as proved above. It suffices to show that if the number of envy edges remains unchanged and an addable set still exists, then the size of the maximal addable set must strictly decrease.

Note that based on how we choose i^* in line 16, after the cake-adding phase, at least one equality edge will be generated in the envy graph from agent i^* to some agent $j \in S$. Let G and G' be the envy graphs before and after the cake-adding phase. Let S and S' be the maximal addable set of G and G', respectively. In the following we will show that $S' \subsetneq S$.

We first show $S' \subseteq S$. Suppose otherwise, we will show that $S \cup S'$ is also an addable set in G, which contradicts to the maximality of S. The reasons that $S \cup S'$ is an addable set in G are as follows.

(i) We have already proved that compared to G, there is no new envy edge in G'. If G and G' has the same number of envy edges, they must share exactly the same set of envy edges. Hence, there will be no envy edge pointing to either S or S' in G.

(ii) If there is an equality edge from $N \setminus (S \cup S')$ to $S \cup S'$ in G, this equality edge cannot be from $N \setminus (S \cup S')$ to S because S is an addable set in G. Hence, it must be from $N \setminus (S \cup S')$ to $S' \setminus S$. This equality edge remains in G' because neither the agents in $N \setminus (S \cup S')$ nor the agents in $S' \setminus S$ receive any good. However, this is impossible because S' is an addable set in G'. In summary, there cannot be any equality edges from $N \setminus (S \cup S')$ to $S \cup S'$ in G.

To further show that $S' \subsetneq S$, recall that according to our algorithm, at least one equality edge, from agent i^* in $N \setminus S$ to some agent $j \in S$, will be included in G'. It is then clear that j cannot be in S'. This concludes the proof. □

Lemma 4.11. *After the algorithm completes an envy-cycle-elimination phase, the number of envy edges strictly decreases.*

Proof. The basic idea of this proof follows from Lipton et al. [113], albeit only strict envy edges were considered in their context. In the envy-cycle-elimination phase, an envy cycle T is eliminated by giving agent j's bundle to agent i for each edge $i \xrightarrow{\text{ENVY}} j$ or $i \xrightarrow{\text{EQ}} j$ in the cycle. First, this process does not affect the bundles of agents in $N \setminus T$, hence the set of envy edges among them remains the same. Next, since we only swap bundles in this phase, the number of envy edges from $N \setminus T$ to T remains the same. In addition, every agent $i \in T$ receives a weakly better bundle, meaning that the number of envy edges from T to $N \setminus T$ does not increase. Finally, because T contains at least one envy edge, some agent in T will receive a strictly better bundle. As a result, although some envy edges between agents in T may still exist, the total number of envy edges will decrease by at least one. □

Lemma 4.12. *Algorithm 2 terminates in polynomial time with $O(n^3)$ calls to the* PERFECTALLOC *oracle and $O(n^4)$ Robertson-Webb queries.*

Proof. We start by analyzing the number of calls to the PERFECTALLOC oracle and next focus on the running time and RW queries.

Calls to the PERFECTALLOC Oracle By Invariant A1, each round in Algorithm 2 executes either a cake-adding phase or an envy-cycle-elimination phase. According to Lemmas 4.10 and 4.11, the number of envy edges never increases. Thus the number of rounds in which the number of envy edges strictly decreases is bounded by $O(n^2)$.

We now upper bound the number of cake-adding phase rounds between any two consecutive rounds that decrease the number of envy edges. If the whole remaining cake is allocated (line 7), PERFECTALLOC(C, n, N) is called once and then Algorithm 2 terminates. In the case that a piece of remaining cake is allocated, by Lemma 4.10, the size of the maximal addable set strictly decreases or an addable set no longer exists; in the latter case, the algorithm proceeds to an envy-cycle-elimination phase. Because the size of any addable set is $O(n)$, it means that the number of cake-adding phase rounds between any two consecutive rounds that decrease the number of envy edges is $O(n)$.

Finally, it follows that Algorithm 2 executes at most $O(n^2) \cdot O(n) = O(n^3)$ cake-adding phase rounds. Each such round calls the PERFECTALLOC oracle once. Algorithm 2 makes $O(n^3)$ calls to the PERFECTALLOC oracle.

Polynomial Running Time and RW Queries Note that during the algorithm's run, we add resources to a bundle and rotate bundles among agents, but never split a bundle. For example, the partition of indivisible goods is computed in line 1 and remains the same since then. To avoid redundant computations, we maintain an $n \times n$ array to keep track of $u_i(A_j)$ for all $i, j \in N$ and update them as necessary.

In line 1, finding an EF1 allocation of indivisible goods can be done in $O(mn \log m)$ via the round-robin algorithm [69]. The implementation details are as follows. We first compute the sorted order of goods according to each agent's valuation, which takes $O(nm \log m)$ time overall. Next, in each agent's turn, we keep looking for the next unallocated good in that agent's sorted list. This step takes $O(mn)$ time in total. Therefore, the overall running time of the round-robin algorithm is dominated by $O(mn \log m)$. Next in line 2, the overall time to construct the corresponding envy graph is $O(n^2)$.

We now consider the `while` loop. According to Lemma 4.4, we can find the maximal addable set or decide its non-existence in time $O(n^3)$. In the case that we need to perform an envy-cycle-elimination, an envy cycle T can be found in the following way. Fix an agent i, we can first spend $O(n)$ time scanning all outgoing edges and ignore those equality edges. Then, we apply depth-first search (DFS) starting from vertex i. If there is a back edge pointing to vertex i, then there must be an envy cycle with at least one envy edge, say, e.g., $i \xrightarrow{\text{ENVY}} j$, for some $j \in N$. This takes $O(n^2)$ time since DFS dominates the time complexity. Since there are $O(n)$ agents, overall, this step can be implemented in $O(n^3)$ time.

In the following, we discuss the steps in each phase at length.

Cake-adding phase When we have $S = N$ satisfied in line 6, we implement an envy-free allocation by calling PerfectAlloc(C, n, N). It takes $O(n)$ time to update the allocation. Algorithm 2 then terminates.

It takes $O(n^2)$ time in line 11 to compute δ_i for all $i \in N \setminus S$. Lines 12 and 15 need $O(n)$ evaluation and cut queries, respectively. Once C' is determined in line 17, we can make $O(n)$ evaluation queries from all n agents over C'. Because we use a perfect allocation of C', we can directly compute $u_i(C')/|S|$ for all $i \in N$ to obtain the value increment of each agent in the addable set. It then takes $O(n^2)$ time to update all agents' valuations of all bundles after line 19. After this, updating an envy graph also takes $O(n^2)$ time.

Since only this phase requires RW queries, we summarize here that Algorithm 2 makes $O(n^4)$ Robertson-Webb queries in that there are $O(n^3)$ cake-adding phases (stated earlier in this proof) and each such phase needs $O(n)$ RW queries.

Envy-cycle-elimination phase Since we maintain an array as the reference for agents' valuations over the current bundles, we can rotate the bundles as well as update the array and the envy graph in $O(n^2)$ time.

The remaining steps can be implemented in $O(n)$ time. Overall, Algorithm 2 runs in time $O(mn \log m + n^6)$, where the n^6 term comes from the $O(n^3)$ total number of `while` loops and $O(n^3)$ time to run each loop. □

To conclude, the correctness of Theorem 4.6 is directly implied by Lemmas 4.9 and 4.12.

Remark. Algorithm 2 and our analysis actually work for arbitrary monotonic utilities for indivisible goods as long as utilities for divisible goods are additive.

Bounded Protocol in the RW Model

Even though we showed that Algorithm 2 can produce an EFM allocation, it is not a bounded protocol in the RW model. This is because our algorithm utilizes an oracle that can compute a perfect allocation of any piece of cake. However, while a perfect allocation always exists, it is known that such an allocation cannot be implemented with a finite number of queries in the RW model, even if there are only two agents [138]. Whether there exists a *bounded* protocol in the RW model to compute an EFM allocation remains a very interesting open question. Note that the perfect allocation oracle cannot be implemented even with a finite number of queries, therefore it is even an open question to find a *finite* EFM protocol.

A natural and tempting approach to get a bounded EFM protocol might be replacing the perfect allocation (line 18) with an envy-free allocation, for which a bounded protocol in the RW model is known [15]. Note that doing so would not create any envy within set S. Then, in order to not create any envy from $N \setminus S$ to S, we need the total value of the piece of cake allocated to S to not exceed δ_i (line 11) for every agent $i \in N \setminus S$. However, when doing so, we will not be able to quantify the progress of the algorithm like in Lemma 4.10. Specifically, we can no longer guarantee that either the number of envy edges strictly decreases or the size of the maximal addable set strictly decreases. This is because we are not guaranteed the equality edge from agent i^* to some agent $j \in S$ as we rely on in the proof of Lemma 4.10. In other words, we cannot show the algorithm will always terminate in bounded steps. Interestingly, in sharp contrast, we will show in Section 4.4 that an approximate envy-free protocol, instead of an ϵ-perfect protocol, is enough to give an efficient ϵ-EFM algorithm. We will discuss this phenomenon in further detail later in Section 4.4.

4.3 EFM Allocation in Special Case

In this section, we show two special cases where an EFM allocation can be computed in polynomial time without using the perfect allocation oracle.

4.3.1 Two Agents

We first show that with only two agents, an EFM allocation can be found using a simple cut-and-choose type of algorithm. We start with a partition (M_1, M_2) of all indivisible goods such that agent 1 is EF1 with respect to either bundle. Without loss of generality, we assume that $u_1(M_1) \geq u_1(M_2)$. Next agent 1 adds the cake into M_1 and M_2 so that the two bundles are as close to each other as possible. Note that if $u_1(M_1) > u_1(M_2 \cup C)$, agent 1 would add all cake to M_2. If $u_1(M_1) \leq u_1(M_2 \cup C)$, agent 1 has a way to make the two bundles equal. We then give agent 2 her preferred bundle and leave to agent 1 the remaining bundle.

Theorem 4.13. *Algorithm 3 returns an EFM allocation in the case of two agents in polynomial time.*

Proof. It is obvious that all goods are allocated. We next show that the allocation returned is EFM. Agent 2 is guaranteed to be envy-free (thus EFM) since she gets her preferred bundle between A_1 and A_2. In the following, we focus on agent 1. If $u_1(M_1) \leq u_1(M_2 \cup C)$ holds, agent 1 is indifferent between bundles A_1 and A_2, so either A_1 or A_2 makes her envy-free

Algorithm 3: EFM Allocation for Two Agents

Input: Agents 1 and 2, mixed goods $M \cup C$, as well as utility and density functions.

1 Divide M into M_1, M_2 such that agent 1 is EF1 with respect to either bundle. Assume w.l.o.g. that $u_1(M_1) \geq u_1(M_2)$ (otherwise we can swap M_1 and M_2).
2 **if** $u_1(M_1) \leq u_1(M_2 \cup C)$ **then**
3   Let agent 1 partition C into C_1, C_2 such that $u_1(M_1 \cup C_1) = u_1(M_2 \cup C_2)$.
4   Let $(A_1, A_2) = (M_1 \cup C_1, M_2 \cup C_2)$.
5 **else**
6   Let $(A_1, A_2) = (M_1, M_2 \cup C)$.
7 Give agent 2 her preferred bundle among A_1, A_2 and agent 1 the remaining bundle.

(thus EFM). In the case that $u_1(M_1) > u_1(M_2 \cup C)$ holds, agent 1 is envy-free if she receives A_1, and is EFM if she gets A_2 because A_1 consists of only indivisible goods and there exists some good g in A_1 such that $u_1(A_2) \geq u_1(M_2) \geq u_1(A_1 \setminus \{g\})$.

We can run the polynomial-time round-robin algorithm for two copies of agent 1 on the indivisible goods to obtain the initial partition (M_1, M_2). The remaining steps only take constant running time and a constant number of RW queries. The conclusion follows. □

A Stronger EFM Notion With two agents, an EFX allocation, in which no agent prefers the bundle of another agent following the removal of *any* single good, always exists [132]. This result can be carried over to show the existence of a stronger EFM notion in the mixed goods setting, in which an agent is EFX towards any agent with only indivisible goods, and envy-free towards the rest. Such an allocation can be obtained by using an EFX partition (with respect to agent 1) instead of an EF1 partition in line 1 of Algorithm 3. Moreover, with any number of agents, whenever an EFX allocation exists,[1] we can start with this EFX allocation in line 1 of Algorithm 2. The cake-adding phase maintains the EFM condition. Thus Algorithm 2 will produce an allocation with this stronger notion of EFM.

4.3.2 Any Number of Agents with Piecewise Linear Functions

In the second case, we consider an arbitrary number of agents when agents' valuation functions over the cake are *piecewise linear* (see Definition 2.2).

The only obstacle in converting Algorithm 2 into a bounded protocol is the implementation of a perfect allocation oracle for cake cutting. When agents have piecewise linear functions, Chen et al. [75] showed that a perfect allocation can be computed in polynomial time. This fact, combined with Theorem 4.6, directly implies the following result.

Corollary 4.14. *For any number of agents with piecewise linear density functions over the cake, an EFM allocation can be computed in polynomial time.*

[1] EFX exists for three agents [72] or n agents with identical valuations [132], but the existence of EFX remains open for four or more agents with additive valuations.

4.4 Relaxation: ϵ-EFM

In this section, we focus on ϵ-EFM, a relaxation of the EFM condition. We will also assume without loss of generality that agents' utilities are normalized to 1, i.e., $u_i(M \cup C) = 1$ for all $i \in N$. Despite the computational issues with finding bounded exact EFM protocols, we will show that there is an efficient algorithm in the RW model that computes an ϵ-EFM allocation for general density functions with running time polynomial in n, m and $1/\epsilon$.

Since the difficulty in finding a bounded EFM protocol in the RW model lies in computing perfect allocations of a cake, it might be tempted to simply replace the exact procedure with a bounded protocol which returns an ϵ-perfect allocation. Here, a partition $(C_1, C_2, \ldots, C_k)$ of cake C is *ϵ-perfect* if for all $i \in N$, $j \in [k]$, $\left| u_i(C_j) - \frac{u_i(C)}{k} \right| \leq \epsilon$. Although a bounded ϵ-perfect protocol exists in the RW model [62, 138], all known protocols have running time exponential in $1/\epsilon$.[2] It is still an open question to find an ϵ-perfect allocation with both query and time complexity polynomial in $1/\epsilon$. Therefore, to design an efficient ϵ-EFM protocol, extra work needs to be done to circumvent this issue.

We next define a relaxed version of envy-freeness and envy graph.

Definition 4.15 (ϵ-EF)**.** An allocation $\mathcal{A}$ is said to satisfy *ϵ-envy-freeness (ϵ-EF)* if for all agents $i, j \in N$, $u_i(A_i) \geq u_i(A_j) - \epsilon$.

Definition 4.16 (ϵ-envy graph)**.** Given an allocation $\mathcal{A}$ and a parameter ϵ, the ϵ-envy graph is defined as $G(\epsilon) = (N, E_{\epsilon\text{-envy}} \cup E_{\epsilon\text{-eq}})$, where every vertex represents an agent, and $E_{\epsilon\text{-envy}}$ and $E_{\epsilon\text{-eq}}$ consist of the following two types of edges, respectively:

- ϵ-envy edge: $i \xrightarrow{\epsilon\text{-ENVY}} j$ if $u_i(A_i) < u_i(A_j) - \epsilon$;
- ϵ-equality edge: $i \xrightarrow{\epsilon\text{-EQ}} j$ if $u_i(A_j) - \epsilon \leq u_i(A_i) \leq u_i(A_j)$.

A cycle in an ϵ-envy graph is said to be an *ϵ-envy cycle* if it contains at least one ϵ-envy edge. When $\epsilon = 0$, the ϵ-envy graph degenerates into the envy graph in Definition 4.2.

4.4.1 The Algorithm

The complete algorithm to compute an ϵ-EFM allocation is shown in Algorithm 4. Similarly to Algorithm 2, our algorithm adds resources to the partial allocation iteratively. We always maintain the partial allocation to be $\hat{\epsilon}$-EFM where $\hat{\epsilon}$ is updated increasingly and would never exceed ϵ. This will ensure that the final allocation is ϵ-EFM. Like Algorithm 2, starting with an EF1 allocation of indivisible goods to all agents in line 2, Algorithm 4 then executes in rounds (lines 4 to 25). Even though each round still executes either a cake-adding phase or an envy-cycle-elimination phase, the execution details are different from Algorithm 2.

[2] Brânzei and Nisan [62] showed that an ϵ-perfect allocation can be computed in $O(n^3/\epsilon)$ RW queries. However, although the query complexity is polynomial, the protocol still requires an exponential *running time* because it finds the correct partition of the small pieces into bundles via an exhaustive enumeration, of which no polynomial time algorithm is known.

Algorithm 4: ϵ-EFM Algorithm

Input: Agents N, mixed goods $M \cup C$, utility and density functions, and parameter ϵ.

$\hat{\epsilon} \leftarrow \frac{\epsilon}{4}, \gamma \leftarrow \frac{\epsilon^2}{8n}$
Find an arbitrary EF1 allocation $(A_1, A_2, \ldots, A_n)$ of M to n agents.
Construct an $\hat{\epsilon}$-envy graph $G(\hat{\epsilon}) = (N, E_{\hat{\epsilon}\text{-envy}} \cup E_{\hat{\epsilon}\text{-eq}})$ accordingly.
while $C \neq \emptyset$ **do** // Allocate divisible goods
    if *there exists an addable set* S **then** // Cake-adding phase
        if $S = N$ **then**
            Let $(C_1, C_2, \ldots, C_n) = \epsilon/4$-EFAlloc$(C, N)$.
            $C \leftarrow \emptyset$
            $\hat{\epsilon} \leftarrow \hat{\epsilon} + \epsilon/4$
            Add C_i to bundle A_i for all $i \in N$.
        else
            if $\max_{i \in N \setminus S} u_i(C) \leq \hat{\epsilon}$ **then**
                $C' \leftarrow C, C \leftarrow \emptyset$
            else
                Suppose w.l.o.g. that $C = [a, b]$. For all $i \in N \setminus S$, if $u_i([a, b]) \geq \hat{\epsilon}$, let x_i be a point such that $u_i([a, x_i]) = \hat{\epsilon}$; otherwise, let $x_i = b$.
                $i^* \leftarrow \arg\min_{i \in N \setminus S} x_i$
                $C' \leftarrow [a, x_{i^*}], C \leftarrow C \setminus C'$
            Let $(C_1, C_2, \ldots, C_k) = \gamma$-EFAlloc$(C', S)$ where $k = |S|$.
            $\hat{\epsilon} \leftarrow \hat{\epsilon} + \gamma$
            Add C_i to the bundle of the i-th agent in S.
            Update $\hat{\epsilon}$-envy graph $G(\hat{\epsilon})$ accordingly.
    else // Envy-cycle-elimination phase
        Let T be an $\hat{\epsilon}$-envy cycle in the $\hat{\epsilon}$-envy graph.
        For each agent $j \in T$, give her bundle to agent i who points to agent j in T.
        Update $\hat{\epsilon}$-envy graph $G(\hat{\epsilon})$ accordingly.
return $(A_1, A_2, \ldots, A_n)$

- In the cake-adding phase, instead of allocating some cake to an addable set S in a way that is perfect, we resort to a γ-EF allocation, where γ will be fixed later in Algorithm 4. In the following, we will utilize an algorithm γ-EFAlloc(C, S) that could return us a γ-EF allocation for any set of agents S and cake C. Note that, for any $\bar{\epsilon} > 0$, the algorithm $\bar{\epsilon}$-EFAlloc can be implemented with both running time and query complexity polynomial in the number of agents involved and $1/\bar{\epsilon}$ due to Procaccia [134]. We also update $\hat{\epsilon}$ to a larger number, say $\hat{\epsilon} + \gamma$, in order to avoid generating $\hat{\epsilon}$-envy edges due to cake-adding.

- In the envy-cycle-elimination phase, we eliminate an $\hat{\epsilon}$-envy cycle, instead of an envy cycle, by rearranging the current partial allocation.

4.4.2 Analysis

Our main result for the ϵ-EFM allocation is as follows:

Theorem 4.17. *An ϵ-EFM allocation can be found by Algorithm 4 with running time $O(n^4/\epsilon + mn \log m)$, $O(n^3/\epsilon)$ Robertson-Webb queries, and $O(n/\epsilon)$ calls to the approximate EFAlloc oracle.*

Invariants

We start by showing the following invariants are maintained by Algorithm 4 during its run.

B1. In each round there is either an addable set for the cake-adding phase or an $\hat{\epsilon}$-envy cycle for the envy-cycle-elimination phase.

B2. The partial allocation is always $\hat{\epsilon}$-EFM with the current $\hat{\epsilon}$.

We next prove these invariants in the following.

Lemma 4.18. *Invariant B1 holds during the algorithm's run.*

Proof. The proof is similar to the proof of Lemma 4.7, except that we consider the $\hat{\epsilon}$-envy edge instead of the envy edge. □

Lemma 4.19. *Invariant B2 holds during the algorithm's run.*

Proof. First, it is worth noting that at the beginning, when indivisible goods are allocated, the allocation is EF1 and therefore EFM. We then note that the partial allocation is only updated in lines 10 and 20 in the cake-adding phase as well as in line 24 in the envy-cycle-elimination phase. Given a partial allocation that is $\hat{\epsilon}$-EFM, we will show that each of these updates maintains $\hat{\epsilon}$-EFM with the updated $\hat{\epsilon}$, which completes the proof of Lemma 4.19. We note that $\hat{\epsilon}$ is only updated in the cake-adding phase and is non-decreasing during the algorithm's run.

For analysis in the envy-cycle-elimination phase, the proof is identical to that of Lemma 4.8 in the case of envy-cycle-elimination phase.

We then discuss the cases in the cake-adding phase. For the updated partial allocation in line 10, we allocate the remaining cake C in a way that is $\frac{\epsilon}{4}$-EF which implies that $u_i(C_i) \geq u_i(C_j) - \epsilon/4$ holds for any pair of agents $i, j \in N$. Given the partial allocation that is $\hat{\epsilon}$-EFM, we have

$$u_i(A_i \cup C_i) \geq u_i(A_j \cup C_j) - \hat{\epsilon} - \epsilon/4.$$

Thus it is clear that no $\hat{\epsilon}$-envy edge will be generated among agents in S if we update $\hat{\epsilon}$ to $\hat{\epsilon} + \epsilon/4$ in line 9.

For the updated partial allocation in line 20, we allocate some cake C' to agents in S in a way that is γ-EF. By a similar argument to the case above, we know that no $\hat{\epsilon}$-envy edge will be generated among agents in S if we update $\hat{\epsilon}$ to $\hat{\epsilon}+\gamma$. Then, for any agent $i \in N \setminus S$, we have $u_i(A_i) > u_i(A_j)$ where $j \in S$. As $u_i(C') \leq \hat{\epsilon}$, we have

$$u_i(A_i) > u_i(A_j \cup C'_j) - \hat{\epsilon},$$

where C'_j is the piece of cake allocated to agent $j \in S$. It means that again no $\hat{\epsilon}$-envy edge will be generated from $N \setminus S$ to S when we update $\hat{\epsilon}$ to $\hat{\epsilon}+\gamma$. Last, it is obvious that for any pair of agents in $N \setminus S$, we do not generate any $(\hat{\epsilon}+\gamma)$-envy edge because the bundles of these agents remain the same. We note that $\hat{\epsilon}$ is updated to $\hat{\epsilon}+\gamma$ in line 19 in order to make sure that there is no introduced $\hat{\epsilon}$-envy edge in the updated $\hat{\epsilon}$-envy graph. □

Correctness

Lemma 4.20. *In the cake-adding phase, if the piece of cake to be allocated is not the whole remaining cake, the sum of all agents' valuations on the remaining cake decreases by at least $\hat{\epsilon}$.*

Proof. We consider the cake-adding phase in lines 5 to 21. If the piece of cake C' to be allocated is not the whole remaining cake, there exists an agent i^* in $N \setminus S$ such that $u_{i^*}(C') = \hat{\epsilon}$ according to lines 15 to 17. Thus the lemma follows. □

We are now ready to show the correctness of Algorithm 4.

Lemma 4.21. *Algorithm 4 always returns an ϵ-EFM allocation upon termination.*

Proof. By Invariant B2, it suffices to prove that all goods are allocated and $\hat{\epsilon}$ is at most ϵ when Algorithm 4 terminates. All indivisible goods are allocated in line 2. Then the `while` loop (lines 4 to 25) terminates only when the cake is also fully allocated, as desired.

We now turn our attention to $\hat{\epsilon}$. First, $\hat{\epsilon}$ is initialized to be $\epsilon/4$ and never decreases during the algorithm's run. If the whole remaining cake is allocated, there is at most one execution of the cake-adding phase (lines 6 to 10). Moreover, $\hat{\epsilon}$ is increased by $\epsilon/4$ in line 9. We will show later in this proof that this increment would not let $\hat{\epsilon}$ exceed ϵ. We then focus on the case where the remaining cake is not fully allocated. There are at most $n/\hat{\epsilon} \leq 4n/\epsilon$ executions of the cake-adding phase (lines 11 to 21) according to Lemma 4.20 and the fact that agents' utilities are normalized to 1. In addition, $\hat{\epsilon}$ is increased by γ in each cake-adding phase in line 19. Thus, $\hat{\epsilon}$ is upper bounded by $\epsilon/4 + 4n/\epsilon \cdot \gamma + \epsilon/4 = \epsilon$ due to $\gamma = \frac{\epsilon^2}{8n}$. It follows that the final allocation is ϵ-EFM. □

Termination and Time Complexity

Lemma 4.22. *In the envy-cycle-elimination phase, the social welfare $\sum_{i \in N} u_i(A_i)$, increases by at least $\hat{\epsilon}$.*

Proof. We eliminate an $\hat{\epsilon}$-envy cycle T which contains at least one $\hat{\epsilon}$-envy edge in the envy-cycle-elimination phase (lines 22 to 25). Line 24 eliminates the cycle by giving agent j's bundle to agent i for each edge $i \xrightarrow{\hat{\epsilon}\text{-ENVY}} j$ or $i \xrightarrow{\hat{\epsilon}\text{-EQ}} j$ in cycle T. None of the agents involved in T is worse off and at least one agent in T is better off by at least $\hat{\epsilon}$ by the definition of $\hat{\epsilon}$-envy edge in Definition 4.16. Since agents outside cycle T do not change their bundles, we complete the proof. $\square$

Lemma 4.23. *Algorithm 4 has running time $O(n^4/\epsilon + mn\log m)$ and invokes $O(n/\epsilon)$ calls to the approximate EFAlloc oracle and $O(n^3/\epsilon)$ Robertson-Webb queries.*

Proof. Since most parts of Algorithm 4 are similar to those in Algorithm 2, and we have discussed their time complexities in the proof of Lemma 4.12, we will focus on the steps that affect the time complexity for Algorithm 4 in this proof.

Similar to Algorithm 2, Algorithm 4 takes $O(mn\log m + n^2)$ time to perform lines 2 and 3. Afterwards, by Invariant B1, Algorithm 4 executes either a cake-adding phase or an envy-cycle-elimination phase in each round, and it takes $O(n^3)$ time to check which phase to go into each time. Next, recall that agents' utilities are normalized to 1 and $\hat{\epsilon}$ is always no less than $\epsilon/4$. This means there are at most $O(n/\epsilon)$ cake-adding rounds by Lemma 4.20 and at most $O(n/\epsilon)$ envy-cycle-elimination rounds by Lemma 4.22.

In the following, we discuss the steps in each phase in details.

Cake-adding phases When we have $S = N$ in line 6, we invoke $\frac{\epsilon}{4}$-EFAlloc once, use $O(n)$ time to update the allocation, and terminate the algorithm.

To determine the piece of cake C' to be allocated later, we need $O(n)$ evaluation queries in line 12, and $O(n)$ cut queries in line 15 if the condition check in line 12 fails. In each cake-adding phase, we invoke the γ-EFAlloc oracle with $\gamma = \frac{\epsilon^2}{8n}$ once (line 18). In order to update agents' valuations for each bundle, we invoke $O(n^2)$ evaluation queries to obtain all agents' valuations of all pieces in $(C_1, C_2, \ldots, C_k)$, and then use $O(n^2)$ time to update the envy graph.

Summarizing everything, we conclude that Algorithm 4 makes $O(n^3/\epsilon)$ Robertson-Webb queries, and $O(n/\epsilon)$ calls to the $\frac{\epsilon^2}{8n}$-EFAlloc oracle in total in all cake-adding phases.

Envy-cycle-elimination phases Since we are keeping track of all agents' valuations for all bundles, it takes no Robertson-Webb queries and $O(n^2)$ time to rearrange the bundles as well as update the $\hat{\epsilon}$-envy graph. Overall all envy-cycle-elimination phases take $O(n^3/\epsilon)$ running time.

The remaining steps can be implemented in time $O(n)$. The overall time complexity of our algorithm is $O(n/\epsilon \cdot n^3 + n/\epsilon \cdot n^2 + mn\log m + n^2) = O(n^4/\epsilon + mn\log m)$. The overall Robertson-Webb query complexity is $O(n^3/\epsilon)$, and the number of calls to the γ-EFAlloc oracle is $O(n/\epsilon)$. $\square$

Finally the correctness of Theorem 4.17 is directly implied by Lemmas 4.21 and 4.23.

Note that at the end of Section 4.2, we explained why an exact envy-free oracle may not be helpful to obtain an EFM allocation. However, as we showed in this section, the approximate envy-free oracle does help to obtain an ϵ-EFM allocation. Lemma 4.20 provides the key difference. In particular, the error allowed in the ϵ-EFM condition ensures that agents' welfare for the remaining cake is reduced by at least an amount of $\hat{\epsilon}$. This claim, however, makes no sense when discussing exact envy-freeness. Furthermore, Algorithm 4 introduces additional error into the EFM condition on top of the error that comes from the approximate envy-free oracle. As a result, even if Algorithm 4 was paired with an exact envy-free oracle, it would still not produce an exact EFM allocation.

4.5 EFM and Economic Efficiency

In this section, we discuss about combining EFM with economic efficiency. In particular, we focus on the well-studied efficiency notion of *Pareto optimality*.

Definition 4.24 (PO)**.** An allocation $\mathcal{A}$ is said to satisfy *Pareto optimality (PO)* if there is no allocation $\mathcal{A}'$ that Pareto-dominates $\mathcal{A}$, i.e., satisfies $u_i(A'_i) \geq u_i(A_i)$ for all $i \in N$ and at least one inequality is strict.

Definition 4.25 (fPO [28])**.** An allocation $\mathcal{A}$ is said to satisfy *fractional Pareto optimality (fPO)* if it is not Pareto dominated by any fractional allocation.[3]

An fPO allocation is also PO but not vice versa [28]. With divisible resources, an allocation that is both envy-free and PO always exists [156]. With indivisible goods, an allocation satisfying both EF1 and fPO (and hence PO) also exists [28]. Next, we show via a counter-example that with mixed types of goods, EFM and PO are no longer compatible.

Example 4.26 (EFM is not compatible with PO)**.** Consider an instance with two agents, one indivisible good, and one cake. Agents' valuation functions are listed below.

	Indivisible good	Cake $C = [0, 1]$
Agent 1	0.6	$u_1(C) = 0.4$ with uniform density over $[0, 0.5]$
Agent 2	0.6	$u_2(C) = 0.4$ with uniform density over $[0.5, 1]$

It is obvious that in any EFM allocation, one agent will get the indivisible good and the entire cake has to be allocated to the other agent. However, such an allocation cannot be PO since the agent with the cake has no value for half of it, and giving that half to the other agent would make that agent better off without making the first agent worse off. △

This counter-example relies on the fact that in the definition of EFM, if some agent i's bundle contains any positive amount of cake, another agent j will compare her bundle to agent i's bundle using the stricter envy-free condition, even if agent j has value zero over

[3] In a fractional allocation, an agent may get a fractional share of an *indivisible* good. We refer to [28] for its formal definition.

agent i's piece of cake. This may seem counter-intuitive, because when agent j has no value over agent i's piece of cake, removing that piece from agent i's bundle will not help eliminate agent j's envy. To this end, one may consider the following weaker version of EFM.

Definition 4.27 (Weak EFM)**.** An allocation $\mathcal{A}$ is said to satisfy *weak envy-freeness for mixed goods (weak EFM)*, if for any agents $i, j \in N$,

- if agent j's bundle consists of indivisible goods with either no divisible good or divisible goods that yield value zero to agent i (i.e., $u_i(C_j) = 0$), there exists an indivisible good $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{g\})$;
- otherwise, $u_i(A_i) \geq u_i(A_j)$.

From the definition, it is easy to see that EFM implies weak EFM, which means that all existence results for EFM established in Section 4.2 can be carried over to weak EFM.

This weaker version of EFM precludes the incompatibility result in Example 4.26. Nevertheless, we show in the following example that weak EFM is incompatible with fPO.

Example 4.28 ((Weak) EFM is incompatible with fPO)**.** Consider an instance with two agents, one indivisible good and two homogeneous divisible goods. Agents' valuation functions are listed below.

	Indivisible good	Divisible good 1	Divisible good 2
Agent 1	2	1	2
Agent 2	2	2	1

Because the valuations are symmetric, we can assume without loss of generality that in an EFM allocation, the indivisible good is given to agent 1. We also observe that in any EFM allocation, we cannot allocate all divisible goods to a single agent. This means that both agents' bundles must contain some divisible good, which then implies that both agents need to be envy-free towards the other agent's bundle. Next, via two simple linear programs, one can compute the maximum utility of each agent in EFM allocations: giving the indivisible good and one half of divisible good 2 to agent 1 gives her a maximum utility of 3; giving divisible good 1 and three quarters of divisible good 2 to agent 2 gives her a maximum utility of 2.75. We note that the maximum utilities for the two agents are achieved under different allocations.

However, even putting these two maximum utilities together, it is dominated by the utilities guaranteed by the fractional allocation in which agent 1 gets divisible good 2 and half of the indivisible good while agent 2 gets divisible good 1 and the other half of the indivisible good, which will give both agents a utility of 3. This means that any EFM allocation is not fPO in this problem instance. △

Could there always exist an allocation that satisfies both weak EFM and PO? We do not know the answer and believe that it is a very interesting open question. One tempting approach to answer this open question is to consider the maximum Nash welfare (MNW) allocation. This is the allocation that maximizes the Nash welfare $\prod_{i \in N} u_i(A_i)$ among all allocations.[4] It has been shown that an MNW allocation enjoys many desirable properties in various settings. In particular, an MNW allocation is always envy-free and PO in cake cutting [144], as well as EF1 and PO for indivisible resource allocation [69]. It is therefore natural to conjecture that it also satisfies EFM and PO for mixed goods. Unfortunately, this is not the case. Here we give such a counter-example.

Example 4.29 (MNW does not imply (weak) EFM)**.** Consider the following instance with two agents, two indivisible goods and some money (a homogeneous divisible good for which every agent has the same value). Agents' valuation functions are listed below.

	Indivisible good 1	Indivisible good 2	Money
Agent 1	0.4	0.4	0.2
Agent 2	49.9	49.9	0.2

We discuss the following cases to find the MNW allocation.

- When both indivisible goods are given to agent 1, giving the whole cake to agent 2 maximizes the Nash welfare, which is $(0.4 + 0.4) \times 0.2 = 0.16$.
- When both indivisible goods are given to agent 2, giving the whole cake to agent 1 maximizes the Nash welfare, which is $0.2 \times (49.9 + 49.9) = 19.96$.
- When each agent gets exactly one indivisible good, in the Nash welfare maximizing allocation, denoted by $\mathcal{A}$, agent 1 receives an indivisible good and the entire cake, and agent 2 receives the other indivisible good. The Nash welfare of $\mathcal{A}$ is $(0.4 + 0.2) \times 49.9 = 29.94$. This is also the overall MNW allocation for this instance.

However, allocation $\mathcal{A}$ is not weak EFM as agent 1's bundle contains some cake that yields positive value to agent 2, and agent 2 is envious of agent 1. It is also worth noting that there is a simple envy-free and PO allocation for this instance: each agent gets one indivisible good and one half of the cake, with Nash welfare $(0.4 + 0.1) \times (49.9 + 0.1) = 25$. $\triangle$

Note that the compatibility of weak EFM and PO remains an open question even for the special case with indivisible goods and a single homogeneous divisible good (e.g., money), even though this case is well-studied when there is enough money.

[4] In the case where the maximum Nash welfare is 0, an allocation is an MNW allocation if it gives positive utility to a set of agents of maximal size and moreover maximizes the product of utilities of the agents in that set.

Chapter 5

Maximin Share Guarantee †

5.1 Introduction

In this chapter, we extend the analysis of MMS allocations to the mixed goods model, study its existence, approximation and computation, and in particular, focus on questions below:

1. Is the worst-case MMS approximation guarantee across all mixed goods instances the same as that across all indivisible goods instances?

2. Given any instance, would adding some divisible resources to it always (weakly) increase the best possible MMS approximation ratio of this instance?

3. How to design algorithms that could find allocations with good MMS approximation guarantee in mixed goods instances?

In Section 5.2, we start by showing that any instance of mixed goods can be converted into another instance with only indivisible goods, such that the two instances have the same maximin share for every agent, and any allocation of the indivisible instance can be converted into an allocation in the mixed instance. This reduction directly implies that the worst-case MMS approximation guarantee across all mixed goods instances is the same as that across all indivisible goods instances. This is not a surprising result, after all, the non-existence of MMS allocations only arises when the resources to be allocated become indivisible. It is therefore reasonable to think that adding divisible goods to the set of indivisible goods can only help with the MMS approximation guarantee. However, we show that this intuition no longer holds at the per-instance level. In particular, we provide an instance with only indivisible goods and when a small amount of cake is added to the instance, the MMS approximation guarantee of the instance *strictly decreases*, i.e., while an α-MMS allocation exists in the original instance, no α-MMS allocation exists after adding the cake.

Next in Section 5.3, we focus on finding allocations of mixed goods with *good* MMS approximations. To this end, we show via a constructive algorithm that given any mixed goods instance, there exists an α-MMS allocation, where the parameter α, ranged between

†This chapter has been published in a paper by Bei, Liu, Lu, and Wang [37].

$1/2$ and 1, is a monotonically increasing function of how agents value the divisible goods relative to their maximin share. The main idea of our algorithm is to repeatedly assign some agent a set of indivisible goods along with a piece of cake to reach α fraction of this agent's maximin share, and then reduce the problem to a smaller size. When the cake to be allocated is heterogeneous, the algorithm also utilizes a generalized fairness notion, *weighted proportionality*, to help allocate the cake. On the computational front, we present polynomial-time approximation schemes to approximate an agent's maximin share and to compute a $(1-\epsilon)\alpha$-MMS allocation. These algorithms run in time polynomial in the number of agents, the number of indivisible goods, and the input bit length.

5.2 MMS Approximation Guarantee

We examine how mixed goods affect the existence and approximation of MMS allocations.

5.2.1 Worst-Case MMS Approximation Guarantee

Maximin share guarantee, while being an appealing solution concept, may not always be satisfied with indivisible goods [108]. Therefore, one has to resort to approximate MMS allocations. Allocating mixed goods generalizes the indivisible goods case, and hence, suffers from the same issue. We start by analyzing the worst-case MMS approximation guarantee for mixed goods instances.

Definition 5.1. Given a mixed goods instance I, let $\gamma(I)$ denote the maximum value of α such that this instance admits an α-MMS allocation.[1] We also call $\gamma(I)$ the *MMS approximation guarantee* of instance I.

We further define two constants: $\gamma_{\text{mix}} := \inf_{I=\langle N, M\cup C\rangle} \gamma(I)$ and $\gamma_{\text{ind}} := \inf_{I=\langle N,M\rangle} \gamma(I)$. Put differently, γ_{mix} (resp., γ_{ind}) is the worst MMS approximation guarantee across all mixed (resp., indivisible) goods instances. We have $\gamma_{\text{ind}} < 1$ according to [108] and $\gamma_{\text{ind}} \geq \frac{3}{4} + \frac{1}{12n}$ according to [91]. It is also straightforward from the definition that $\gamma_{\text{mix}} \leq \gamma_{\text{ind}}$. In the following, we show via the following reduction theorem that γ_{mix} is also at least γ_{ind}.

Theorem 5.2. *Given any mixed goods instance $I = \langle N, M \cup C\rangle$, there is another instance $I' = \langle N, M'\rangle$ with only indivisible goods M' and the same set of agents N such that*

- *any allocation $\mathcal{A}'$ of M' can be converted into another allocation $\mathcal{A}$ of $M \cup C$ with $u_i(A_i) = u_i(A'_i)$ for each agent $i \in N$;*
- *$\text{MMS}_i(n, M \cup C) = \text{MMS}_i(n, M')$ for each agent $i \in N$.*

Proof. We first transform the mixed goods instance $I = \langle N, M \cup C\rangle$ into an instance $I' = \langle N, M'\rangle$ with only indivisible goods. Consider an agent i and an MMS partition $\mathcal{P}_i$ for this

[1] The $\gamma(I)$ is defined to be the maximum value of α instead of the supremum in that the density functions are non-atomic and the maximum α can always be achieved.

agent in I. Clearly, we can assume that $\mathcal{P}_i$ divides cake C into at most n intervals with at most $n-1$ cuts. This assumption is without loss of generality because in an MMS partition it only matters how much value worth of cake is assigned to each bundle, but not their positions. Then, by collecting all cuts from all n MMS partitions $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$ on cake C, the cuts break up the cake into at most $n(n-1)+1$ intervals. Let these intervals of C be the set M'' of "indivisible frozen intervals". Together with M, we now have $M' = M'' \cup M$.

Given any allocation $\mathcal{A}'$ of M', we can easily convert it into an allocation $\mathcal{A}$ of $M \cup C$ by transforming those "indivisible frozen intervals" back to normal pieces of cake. This also gives $u_i(A_i) = u_i(A'_i)$ for each agent $i \in N$, which proves the first part of Theorem 5.2.

It is also clear that every agent can have the same MMS partition in I' as that in I, because the cuts do not affect their MMS partitions. This implies that $\text{MMS}_i(n, M') \geq \text{MMS}_i(n, M \cup C)$ for each agent $i \in N$. On the other hand, the first part of the theorem also implies $\text{MMS}_i(n, M \cup C) \geq \text{MMS}_i(n, M')$. Hence, we have $\text{MMS}_i(n, M \cup C) = \text{MMS}_i(n, M')$ for each agent $i \in N$. □

It is worth noting that this reduction is not computationally efficient as it needs to compute agents' MMS partitions. Moreover, Theorem 5.2 directly implies the following result.

Corollary 5.3. $\gamma_{\text{ind}} = \gamma_{\text{mix}}$.

In other words, having mixed types of goods does not affect the worst-case MMS approximation guarantee across all instances. As another corollary, this also means that if there exists a universal β-MMS algorithm for indivisible goods for some β, it immediately implies that every instance with mixed goods also admits a β-MMS allocation. We will discuss more about the algorithmic implication of this result in Section 5.3.4.

5.2.2 Cake Does Not Always Help

It is noteworthy that Corollary 5.3 is about the worst-case MMS approximation guarantee across all instances. We next show that such an equivalence may not hold on a per-instance level. In particular, we will demonstrate via an example that sometimes, adding some divisible goods to some instance I may hurt its MMS approximation guarantee $\gamma(I)$.

Theorem 5.4. *For any $n \geq 6$, there exist some set of agents N, indivisible goods M, and divisible goods C, such that $\gamma(\langle N, M \rangle) > \gamma(\langle N, M \cup C \rangle)$. That is, adding some divisible goods to a set of goods may decrease the MMS approximation guarantee of the instance.*

We start by explaining the intuition behind the proof. We want to find an instance $I = \langle N, M \rangle$ such that $\gamma(\langle N, M \rangle) < 1$, and the instance should have the following properties.

Fix an agent i. In her MMS partition, the least valued bundle is unique, i.e., the value of the least valued bundle is *strictly* less than that of the second worst bundle. If this is the case, then given a cake C with small enough value ϵ, the new maximin share of this agent $\text{MMS}_i(n, M \cup C)$ should be exactly $\text{MMS}_i(n, M) + \epsilon$. Now, suppose that in instance

I, all of the agents have this property. This means that every agent's maximin share will increase by ϵ when we add cake C to instance I. The second required property of instance I is that in any $\gamma(\langle N, M\rangle)$-MMS allocation, there are at least two agents receiving a value of exactly $\gamma(\langle N, M\rangle)$ times their maximin share. With these two properties, the actual cake C will not be enough for distributing to all of the agents while clinging to a large enough MMS approximation ratio $\gamma(\langle N, M \cup C\rangle)$. In other words, with cake C being added, the new MMS approximation ratio $\gamma(\langle N, M \cup C\rangle)$ will decrease, comparing with $\gamma(\langle N, M\rangle)$. Finally, the counter-example used to show the non-existence of MMS allocations by Kurokawa et al. [108] can be utilized to construct instance I that satisfies the two aforementioned properties. By utilizing their construction, our argument requires at least six agents.

Proposition 5.5 (Kurokawa et al. [108]). *For any $n \geq 6$, there exists a matrix $M \in \mathbb{R}^{n\times n}$ with the following properties:*

1. *All entries are non-negative (i.e., $\forall i, j\colon M_{i,j} \geq 0$).*
2. *All entries of the last row and column as well as the first entry in the first row are positive (i.e., $M_{1,1} > 0$ and $\forall i\colon M_{i,n}, M_{n,i} > 0$).*
3. *All rows and columns sum to 1 (i.e., $M\vec{1} = M^\top\vec{1} = \vec{1}$).*
4. *Define M^+ as the set of all positive entries in M. Then if we wish to partition M^+ into n subsets that sum to exactly 1, then our partition must correspond to either the rows of M or the columns of M.*

Proof of Theorem 5.4. Construct two $n \times n$ matrices P^+ and P^-. Let $P^+_{1,1} = P^-_{1,1} = -\epsilon$, $P^+_{n,1} = P^-_{1,n} = -\epsilon$, $P^+_{n,n} = P^-_{n,n} = (2n-3)\epsilon$, and $P^+_{n,i} = P^-_{i,n} = -2\epsilon$ for $2 \leq i \leq n-1$. Take $n = 6$ as an example, we show below the construction of matrices P^+ and P^-:

$$P^+ = \begin{bmatrix} -\epsilon & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\epsilon & -2\epsilon & -2\epsilon & -2\epsilon & -2\epsilon & 9\epsilon \end{bmatrix} \quad \text{and} \quad P^- = \begin{bmatrix} -\epsilon & 0 & 0 & 0 & 0 & -\epsilon \\ 0 & 0 & 0 & 0 & 0 & -2\epsilon \\ 0 & 0 & 0 & 0 & 0 & -2\epsilon \\ 0 & 0 & 0 & 0 & 0 & -2\epsilon \\ 0 & 0 & 0 & 0 & 0 & -2\epsilon \\ 0 & 0 & 0 & 0 & 0 & 9\epsilon \end{bmatrix}$$

Now, consider a matrix M satisfying all properties listed in Proposition 5.5. By setting ϵ to a sufficiently small value, we can always make sure that every entry of $M + P^+$ and $M + P^-$ is non-negative. In the following, we will regard each entry of matrix M as an indivisible good. We next divide N into two disjoint subsets. One subset contains $\lfloor \frac{n}{2} \rfloor$ agents, denoted by N^+; the other contains the remaining agents, denoted by N^-. We let each agent $i \in N^+$ take the values of n^2 goods as in matrix $M + P^+$, and each agent $i \in N^-$ take the values of n^2 goods as in matrix $M + P^-$. We call this instance I. One can check that in instance I, the maximin share of every agent is $1 - \epsilon$.

According to the fourth property in Proposition 5.5, there are only two ways to distribute these goods into n bundles such that each bundle has value close to 1: either the rows of M or the columns of M. In each of these two partitions, we can always have at least two agents who value their bundles exactly $1 - 2\epsilon$. For example, there are two such agents from N^+ if the partition is n columns, or two from N^- if the partition is n rows. In particular, this means that the MMS approximation guarantee $\gamma(I)$ of instance I is $\frac{1-2\epsilon}{1-\epsilon}$.

Suppose now that we add a homogeneous cake to instance I. This cake has value ϵ to each agent. Therefore, every agent's maximin share will now increase exactly from $1 - \epsilon$ to 1. However, in any allocation, there will still be at least two agents whose values for the indivisible goods are at most $1 - 2\epsilon$. Then the best possible way to distribute the cake is to allocate it only to those agents, which means that at least one such agent will receive a bundle of value at most $1-2\epsilon+\epsilon/2 = 1-3\epsilon/2$. Thus, in this case, the MMS approximation ratio of this agent will be at most $1-3\epsilon/2$, which is strictly less than $\frac{1-2\epsilon}{1-\epsilon}$ when $\epsilon < 1/3$. $\square$

5.3 Existence and Computation of Approximate MMS Allocations

Section 5.2 investigates MMS approximation guarantee, which is the *best possible* MMS approximation of an instance. In this section, our goal is to design algorithms that could compute allocations with *good* MMS approximation ratios in mixed goods instances. We hope such an algorithm can be flexible in the sense that when the instance contains only indivisible goods, the MMS approximation of the output allocation should match or be close to the currently best approximation ratio for indivisible goods; on the other hand, when the resources contain enough valuable divisible goods, the indivisible goods would become negligible, and our algorithm should be able to produce an MMS allocation.

As the main result of this section, in the following, we present such an algorithm. We will show that the algorithm will always produce an α-MMS allocation in the mixed goods setting, where α is a monotonically increasing function of how agents value the divisible goods relative to their maximin share and ranges between $1/2$ and 1.

Precision and Input Representation When discussing the computational aspects, it is necessary to specify the precision and representation of the input instance. In this section, we assume that $u_i(g)$'s for all $i \in N, g \in M$ and $u_i(C)$'s for all $i \in N$ are rational numbers as well as the whole input can be represented in at most L bits.

Theorem 5.6. *Given any mixed goods instance $\langle N, M \cup C \rangle$, an α-MMS allocation always exists, where*

$$\alpha = \min\left\{1, \frac{1}{2} + \min_{i \in N}\left\{\frac{u_i(C)}{2 \cdot (n-1) \cdot \textit{MMS}_i}\right\}\right\}.$$

Furthermore, for any constant $\epsilon > 0$, we can compute a ratio α' and an allocation $\mathcal{A}$ in time polynomial in n, m, L such that

1. $\alpha' \geq \alpha$, and

2. $\mathcal{A}$ is $(1-\epsilon)\alpha'$-MMS.

Here, n is the number of agents, m is the number of indivisible goods, and L is the total bit length of all input parameters.

When every agent i has $u_i(C) \geq \frac{n}{2} \cdot \mathrm{MMS}_i$, Theorem 5.6 implies the existence of an α-MMS allocation with α being better than the currently best approximation ratio of $\frac{3}{4} + \frac{1}{12n}$ for indivisible goods [91]. In addition, we show in the following corollary the amount of divisible goods needed to ensure that an instance admits an MMS allocation.

Corollary 5.7. *Given a mixed goods instance $\langle N, M \cup C \rangle$, if $u_i(C) \geq (n-1) \cdot MMS_i$ holds for every agent $i \in N$, then an MMS allocation is guaranteed to exist.*

It means that even with the presence of indivisible goods, as long as there are enough valuable cake, an MMS allocation can always be found. This corollary, however, should not be interpreted as indicating the *least* amount of cake required. In a related line of work, Halpern and Shah [97] and Brustle et al. [64] studied the allocation of indivisible goods together with a very special type of divisible goods, i.e., money and bounded the amount of money needed for an indivisible goods instance to have an envy-free allocation, assuming that the value of each agent for each good is at most 1. Although an envy-free allocation satisfies MMS guarantee, their results and this corollary are not comparable because we have different objectives, and it is not our goal to find the minimum amount of cake needed to ensure an MMS allocation.

The remainder of this section is dedicated to the proof of Theorem 5.6, consisting of three steps as follows, and discuss how to improve the approximation ratio α in Section 5.3.4.

Section 5.3.1 We start by focusing on a restricted case in which the cake to be allocated is *homogeneous* to every agent. We show via a constructive, but not necessarily polynomial-time, algorithm that an α-MMS allocation always exists in this setting.

Section 5.3.2 Next, we generalize the above algorithm to the case with heterogeneous cake, using a concept called *weighted proportionality* in cake cutting.

Section 5.3.3 We then discuss how to convert the algorithm into a polynomial-time algorithm at the cost of a small loss in the MMS approximation ratio.

5.3.1 Homogeneous Cake

We begin with a special case where the cake to be allocated, denoted as $\widehat{C}$, is *homogeneous*.

The pseudocode to compute an α-MMS allocation is presented as Algorithm 5. Our algorithm is in spirit similar to the one devised by Ghodsi et al. [93]. The main idea of the algorithm is to repeatedly assign some agent a bundle of goods which is worth at least α times this agent's maximin share and then reduce the problem to a smaller size. Specifically, after initialization, our algorithm can be decomposed into two phases as follows:

Algorithm 5: MIXED-MMS-HOMOGENEOUS$(N, M \cup \widehat{C})$

Input: Agents N, indivisible goods M and a homogeneous cake $\widehat{C}$, as well as utility and density functions.

1 Compute MMS_i for each $i \in N$.
2 $\alpha \leftarrow \min\left\{1, \frac{1}{2} + \min_{i \in N}\left\{\frac{u_i(\widehat{C})}{2 \cdot (n-1) \cdot \text{MMS}_i}\right\}\right\}$
3 $A_1, A_2, \dots, A_n \leftarrow \emptyset$
4 **while** $\exists i \in N, g \in M, u_i(g) \geq \alpha \cdot \mathit{MMS}_i$ **do** // Phase 1
5     $A_i \leftarrow \{g\}$ // arbitrary tie-breaking
6     $N \leftarrow N \setminus \{i\}$, $M \leftarrow M \setminus \{g\}$
7 **while** $|N| \geq 2$ **do** // Phase 2
8     $B \leftarrow \emptyset$
9   Add one indivisible good at a time to B until $u_j(B) \geq (1-\alpha) \cdot \text{MMS}_j$ for some agent j, or $B = M$.
10   Suppose w.l.o.g. that $\widehat{C} = [a, b]$. For each agent $i \in N$, let x_i be the leftmost point such that $u_i(B \cup [a, x_i]) \geq \alpha \cdot \text{MMS}_i$.
11     $i^* \leftarrow \arg\min_{i \in N} x_i$ // arbitrary tie-breaking
12     $A_{i^*} \leftarrow B \cup [a, x_{i^*}]$
13     $N \leftarrow N \setminus \{i^*\}$, $M \leftarrow M \setminus B$, $\widehat{C} \leftarrow \widehat{C} \setminus [a, x_{i^*}]$
14 Give all remaining goods to the last agent.
15 **return** $(A_1, A_2, \dots, A_n)$

Phase 1: allocate big indivisible goods (lines 4 to 6) This phase repeatedly allocates some agent an indivisible good which has a value at least α times this agent's maximin share. Then, both the agent and the allocated indivisible good are removed from all further considerations.

Phase 2: allocate small indivisible goods and cake (lines 7 to 13) In each round of this phase, Algorithm 5 chooses an agent i^* and allocates some indivisible goods B (formed in line 9) along with a piece of cake $[a, x_{i^*}]$ to this agent (line 12). Then again, both the agent and her goods are removed from the instance.

Algorithm 5 consists of two mutually exclusive phases, and thus we will go through their analyses separately as follows.

Phase 1: Allocate Big Indivisible Goods When goods are all indivisible, it follows from Bouveret and Lemaître [49, Lemma 1] that allocating a single good to an agent does not decrease the MMS values of other agents; we adopt the name "monotonicity property" from Amanatidis et al. [7]. Here, we show this result holds in the mixed goods setting as well.

Lemma 5.8 (Monotonicity property)**.** *Given an instance $\langle N, G = M \cup C\rangle$, for any agent $i \in N$ and any indivisible good $g \in M$, it holds that $\mathit{MMS}_i(n-1, G \setminus \{g\}) \geq \mathit{MMS}_i(n, G)$.*

Proof. Removing a single indivisible good in an MMS partition of agent i affects exactly one bundle and each of the remaining $n-1$ bundles has value at least $\text{MMS}_i(n, G)$. Therefore, we have $\text{MMS}_i(n-1, G \setminus \{g\}) \geq \text{MMS}_i(n, G)$. □

Denote by N_1 the set of n_1 remaining agents and G_1 the set of unallocated goods just before Phase 2 executes. Applying the monotonicity property (Lemma 5.8) $n - n_1$ times, we have that for each agent $i \in N_1$, $\text{MMS}_i(n_1, G_1) \geq \text{MMS}_i(n, G)$. In addition, each agent i who leaves the system in Phase 1 receives an indivisible good of value at least $\alpha \cdot \text{MMS}_i(n, G)$. This implies that Phase 1 will not affect the correctness and termination of Algorithm 5. It also adds the property that in Phase 2, each remaining agent i will value each of the remaining indivisible goods less than $\alpha \cdot \text{MMS}_i$.

Phase 2: Allocate Small Indivisible Goods and Cake At each round of this phase, for agent i^* selected in line 11, we show it satisfies the following two properties:

(1) $u_{i^*}(A_{i^*}) \geq \alpha \cdot \text{MMS}_{i^*}$;

(2) for each agent j remaining in N, $u_j(A_{i^*}) \leq \text{MMS}_j$.

Property (1) is straightforward by the way each x_i is computed in line 10. To prove Property (2), we remark that no single good is valued more than $\alpha \cdot \text{MMS}_i$ for any agent i. Therefore, the set B formed in line 9 must satisfy $u_j(B) \leq \text{MMS}_j$ for all $j \in N$. In line 10, each agent cuts a piece of cake such that the sum of her value for B and this piece of cake is at least α fraction of her maximin share. Because $\alpha \leq 1$ and the cake is divisible, after line 10, it continues to satisfy that $u_j(B \cup [a, x_j]) \leq \text{MMS}_j$ for each $j \in N$. Then, because i^* is selected such that x_{i^*} is the smallest value, we have $u_j(A_{i^*} = B \cup [a, x_{i^*}]) \leq u_j(B \cup [a, x_j]) \leq \text{MMS}_j$ for each agent $j \in N$. In particular, Property (2) ensures that the last agent in line 14 is still left with enough goods to reach her maximin share. Therefore, every agent i will receive a value of at least $\alpha \cdot \text{MMS}_i$ after the two phases.

It only remains to show that cake $\widehat{C}$ is enough to be allocated throughout the process.

Lemma 5.9. *Cake $\widehat{C}$ is enough to be allocated in Algorithm 5. In other words, x_i for each agent $i \in N$ in line 10 is always well defined in each round.*

Proof. Line 2 of Algorithm 5 indicates that for each agent $i \in N$, $u_i(\widehat{C}) \geq (n-1) \cdot (2\alpha - 1) \cdot \text{MMS}_i$. As a result, each agent i has value at least $(2\alpha - 1) \cdot \text{MMS}_i$ for a $1/(n-1)$ fraction of the entire cake $\widehat{C}$. It is also clear that Phase 2 has been executed at most $n-1$ times during the algorithm's run. That is to say the action of cutting a piece of cake $\widehat{C}$ and allocating this piece to an agent is performed at most $n-1$ times.

Based on whether there exists some agent who values B in line 9 at least $(1-\alpha)$ times her maximin share, we distinguish two cases as follows.

Line 9: there exists some agent j with $u_j(B) \geq (1-\alpha) \cdot \text{MMS}_j$ As mentioned earlier, a $1/(n-1)$ fraction of $\widehat{C}$ is worth at least $(2\alpha-1) \cdot \text{MMS}_j$. Thus along with B, it is enough to give agent j a value of at least $\alpha \cdot \text{MMS}_j$. This means that in line 10, the length of $[a, x_j]$ is at most $1/(n-1)$. Moreover, Algorithm 5 chooses the agent who claims the smallest piece of cake as agent i^* (line 11), meaning that the length of $[a, x_{i^*}]$ is again at most $1/(n-1)$.

Combining the fact that Phase 2 executes at most $n-1$ times, if this case holds every time, the cake will be enough.

Line 9: $u_j(B) < (1-\alpha) \cdot$ MMS$_j$ for each agent j In this case, B is set to be M in line 9. We also note that after the first time of this case, M will become empty, and the agents left will divide only the cake for the remaining rounds. Let k be the number of the remaining agents when M becomes empty. By Property (2) that we showed above, we know the remaining cake is worth at least $k \cdot \text{MMS}_i$ for each remaining agent i. Thus, it is enough for each agent i to receive a piece with value at least $\alpha \cdot \text{MMS}_i$. □

Combining everything together, we conclude that Algorithm 5 is correct and always outputs an α-MMS allocation.

5.3.2 Heterogeneous Cake

We now show how to extend Algorithm 5 to the general setting with the heterogeneous cake C. Our new algorithm follows a very simple idea. First, we replace cake C with a homogeneous cake $\widehat{C}$ such that $u_i(\widehat{C}) = u_i(C)$ for every agent $i \in N$, and next allocate goods M and $\widehat{C}$ to all of the agents using Algorithm 5. Let $\widehat{C}_i$ be the piece allocated to agent i in this step. Note that since cake $\widehat{C}$ is homogeneous, only the length of $\widehat{C}_i$ matters, which we denote as w_i. Because the length of cake $\widehat{C}$ is 1, w_i also represents the fraction of cake $\widehat{C}$ allocated to agent i. Next, we view w_i as the *entitlement* (or *weight*) of agent i to cake C, and obtain the actual allocation of cake C via a procedure known as the *weighted proportional allocation*.

Weighted Proportional Cake Cutting This concept generalizes proportionality to the weighted case in cake cutting. Formally, assume that every agent $i \in N$ is assigned a non-negative *weight* w_i such that $\sum_{i \in N} w_i = 1$. We call the vector of weights $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ a *weight profile*.

Definition 5.10. Given a weight profile $\mathbf{w}$, an allocation $\mathcal{C} = (C_1, C_2, \ldots, C_n)$ of cake C is said to satisfy *weighted proportionality (WPR)* if for each agent $i \in N$, $u_i(C_i) \geq w_i \cdot u_i(C)$.

A weighted proportional allocation of cake gives each agent at least her entitled fraction of the entire cake from her own perspective. The notion of proportionality in Definition 2.1 is a special case of WPR with $\mathbf{w} = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$. With any set of agents and any weight profile, a WPR allocation always exists [77]. In the following, we will assume that our algorithm is equipped with the protocol $\text{WPRA}\textsc{lloc}(N, C, \mathbf{w})$ that could return us a weighted proportional allocation of cake C, among the set of agent N with weight profile $\mathbf{w}$.

The pseudocode to compute an α-MMS allocation of mixed goods for any number of agents is presented as Algorithm 6. To show that this algorithm can find an α-MMS allocation with mixed goods that contain a heterogeneous cake, it suffices to prove the following two simple facts:

Algorithm 6: Mixed MMS Algorithm

Input: Agents N, mixed goods $M \cup C$, as well as utility and density functions.

Let $\widehat{C} = [0, 1]$ be a homogeneous cake with $u_i(\widehat{C}) = u_i(C)$ for each agent $i \in N$.

$(M_1 \cup \widehat{C}_1, \ldots, M_n \cup \widehat{C}_n) \leftarrow$ Mixed-MMS-Homogeneous$(N, M \cup \widehat{C})$

foreach $i \in N$ **do** `// Assign weights.`

 if $u_i(C) > 0$ **then** $w_i \leftarrow u_i(\widehat{C}_i)/u_i(C)$ **else** $w_i \leftarrow 0$

$(C_1, C_2, \ldots, C_n) \leftarrow$ WPRAlloc$(N, C, \mathbf{w} = (w_1, \ldots, w_n))$

return $(M_1 \cup C_1, M_2 \cup C_2, \ldots, M_n \cup C_n)$

1. $\text{MMS}_i(n, M \cup C) = \text{MMS}_i(n, M \cup \widehat{C})$. This is obvious because both C and $\widehat{C}$ are divisible and we have $u_i(C) = u_i(\widehat{C})$. Only changing the density of a cake will not affect the maximin share of any agent.

2. $u_i(C_i) \geq u_i(\widehat{C}_i)$. Due to WPR, we have $u_i(C_i) \geq w_i \cdot u_i(C) = w_i \cdot u_i(\widehat{C}) = u_i(\widehat{C}_i)$.

5.3.3 Computation

We now investigate the computational issues in finding an α-MMS allocation in this section. Note that Algorithm 6 is not a polynomial-time algorithm unless P=NP in that it requires the knowledge of every agent's maximin share, which is NP-hard to compute even with only indivisible goods [108]. To obtain a polynomial-time approximation algorithm, we start by showing how to approximate the maximin share of an agent with mixed goods, and then focus on obtaining an approximate α-MMS allocation.

Approximate Maximin Share with Mixed Goods

When goods are indivisible, Woeginger [159] showed a polynomial-time approximation scheme (PTAS) to approximately compute the maximin share of an agent. More specifically, given any constant $\delta > 0$ and any agent, we can partition the indivisible goods into n bundles in polynomial time, such that each bundle is worth at least $1 - \delta$ of that agent's maximin share. Utilizing this PTAS, we present here a new PTAS to approximate an agent's maximin share with mixed goods.

Lemma 5.11. *Given any mixed goods instance $I = \langle N, M \cup C \rangle$ and constant $\epsilon > 0$, for any agent $i \in N$, one can compute a partition $(P_1, P_2, \ldots, P_n)$ of $M \cup C$ in polynomial time, such that $\min_{j \in N} u_i(P_j) \geq (1 - \epsilon) \cdot \text{MMS}_i(n, M \cup C)$.*

Proof. Let agent i cut the cake C into $\lceil \frac{2n}{\epsilon} \rceil$ disjoint intervals worth at most $\frac{\epsilon \cdot u_i(C)}{2n}$ each to this agent. Denote by $\widetilde{C}$ the collection of these discretized, indivisible intervals. The new discretized instance is then denoted by $I' = \langle N, M \cup \widetilde{C} \rangle$. It is worth noting that I' is an instance with only indivisible goods.

We first claim that

$$\text{MMS}_i(n, M \cup C) \geq \text{MMS}_i(n, M \cup \widetilde{C}) \geq \left(1 - \frac{\epsilon}{2}\right) \cdot \text{MMS}_i(n, M \cup C).$$

The first inequality holds trivially by definition. We now proceed to show the second inequality. Consider an MMS partition $\mathcal{T}$ of instance I for agent i. We construct a partition $\mathcal{T}'$ of instance I' as follows. First, let the partition of its original indivisible goods M be exactly the same as that in $\mathcal{T}$. We then distribute the intervals $\widetilde{C}$ to these n bundles. For any bundle whose value is strictly less than $\left(1 - \frac{\epsilon}{2}\right) \cdot \text{MMS}_i(n, M \cup C)$ to agent i, add one interval at a time to this bundle until agent i's value for this bundle falls in

$$\left[\left(1 - \frac{\epsilon}{2}\right) \cdot \text{MMS}_i(n, M \cup C), \text{MMS}_i(n, M \cup C)\right].$$

This is possible because we have $\text{MMS}_i(n, M \cup C) \geq \frac{u_i(C)}{n}$ and each interval is worth at most $\frac{\epsilon \cdot u_i(C)}{2n} \leq \frac{\epsilon}{2} \cdot \text{MMS}_i(n, M \cup C)$. Furthermore, $\widetilde{C}$ has enough pieces for such a distribution, because in $\mathcal{T}$, each bundle is worth at least $\text{MMS}_i(n, M \cup C)$ to agent i. Repeat this procedure for all n bundles. Finally, we arbitrarily distribute any remaining intervals to these bundles. Let the resulting partition be $\mathcal{T}'$.

At the end of these procedures, each bundle in $\mathcal{T}'$ is worth at least $\left(1 - \frac{\epsilon}{2}\right) \cdot \text{MMS}_i(n, M \cup C)$. Thus, by the definition of maximin share, the second inequality also holds. We remark that the aforementioned procedures are not actually implemented in our algorithm. They are only used to demonstrate the difference of maximin share of the two instances.

Now, since I' is an instance with only indivisible goods, we are able to compute a partition $(P_1, P_2, \ldots, P_n)$ of instance I' such that $\min_{j \in N} u_i(P_j) \geq \left(1 - \frac{\epsilon}{2}\right) \cdot \text{MMS}_i(n, M \cup \widetilde{C})$ using the PTAS of Woeginger [159] with $\delta = \epsilon/2$. It then holds that

$$\min_{j \in N} u_i(P_j) \geq \left(1 - \frac{\epsilon}{2}\right) \cdot \left(1 - \frac{\epsilon}{2}\right) \cdot \text{MMS}_i(n, M \cup C) \geq (1 - \epsilon) \cdot \text{MMS}_i(n, M \cup C). \quad \square$$

Lemma 5.11 also implies that in the mixed goods setting, we can compute in polynomial time a value MMS'_i such that $\text{MMS}_i(n, M \cup C) \geq \text{MMS}'_i \geq (1 - \epsilon) \cdot \text{MMS}_i(n, M \cup C)$.

Approximate α-MMS Allocation

We are now ready to present a polynomial-time algorithm to compute an approximate α-MMS allocation. It is almost similar to Algorithm 6, except for the following two differences:

1. line 1 of Algorithm 5: we compute an approximate maximin share, MMS'_i, which is at most $\text{MMS}_i(n, M \cup C)$ and at least $(1 - \epsilon) \cdot \text{MMS}_i(n, M \cup C)$ for each agent $i \in N$;

2. line 2 of Algorithm 5: we compute a ratio α' using approximate maximin share, i.e.,

$$\alpha' \leftarrow \min\left\{1, \frac{1}{2} + \min_{i \in N}\left\{\frac{u_i(C)}{2 \cdot (n-1) \cdot \text{MMS}'_i}\right\}\right\}.$$

A similar analysis to Lemma 5.9 shows that the modified algorithm with these approximate values still terminates. Next, according to Lemma 5.11, we have $\text{MMS}_i(n, M \cup C) \geq \text{MMS}'_i$ for each agent $i \in N$, which implies that $\alpha' \geq \alpha$. Then, for any agent i, by the design of the algorithm, this agent is guaranteed to receive a bundle with value at least $\alpha' \cdot \text{MMS}'_i \geq (1 - \epsilon)\alpha' \cdot \text{MMS}_i(n, M \cup C)$. Therefore, the resulting allocation is $(1 - \epsilon)\alpha'$-MMS.

Time Complexity Analysis

In the light of Lemma 5.11, computing approximate maximin share takes polynomial time. Thus, the only step that needs time complexity analysis is the WPRALLOC$(N, C, \mathbf{w})$ called in line 5 of Algorithm 6, which produces a weighted proportional allocation of cake C among agents N. When all weights are rational numbers, Cseh and Fleiner [77] implemented such a protocol using $O(n \log D)$ queries, where D is the common denominator of weights. They also showed that their implementation is asymptotically the fastest possible.

We have assumed that our input has size at most L bits. Each of the arithmetic operations in steps before line 5 of Algorithm 6 keeps the numbers rational with polynomial bit size. Thus, the WPRALLOC in line 5 of Algorithm 6 can be implemented in polynomial time [77]. Summarizing everything together, we obtain a polynomial-time algorithm.

Sections 5.3.1 to 5.3.3 together complete the proof of Theorem 5.6.

5.3.4 Improvement of the Approximation Ratio

The smallest value of α is $1/2$ in Theorem 5.6, achieved when the resources contain only indivisible goods. However, there is a gap between our result and the currently best approximation guarantee with only indivisible goods, i.e., $\gamma_{\text{ind}} \geq \frac{3}{4} + \frac{1}{12n}$ [91]. In the following, we show a simple procedure which improves the MMS approximation ratio computed by our algorithm to (almost) match the currently best ratio for indivisible goods.

Existence-wise, Corollary 5.3 ($\gamma_{\text{ind}} = \gamma_{\text{mix}}$) implies that we can directly improve the ratio to $\max\{\alpha, \gamma_{\text{ind}}\}$ in Theorem 5.6. We now discuss computation. Suppose that there exists a polynomial-time algorithm guaranteeing to return a β-MMS allocation of indivisible goods. Given a mixed goods instance, we can compute α' and next compare it with β: if $\alpha' \geq \beta$, we directly apply Theorem 5.6; otherwise, we cut cake C into small indivisible intervals with each being valued at most $\frac{\epsilon \cdot u_i(C)}{2n}$ for every agent i, and then apply the β-MMS algorithm to this new "indivisible" goods instance. To conclude, we strengthen our result as follows.

Theorem 5.12. *A $\max\{\alpha, \gamma_{\text{ind}}\}$-MMS allocation with mixed goods always exists for any number of agents. In addition, if there exists a polynomial-time algorithm that is guaranteed to output a β-MMS allocation with indivisible goods, then for any constant $\epsilon > 0$, there is another polynomial-time algorithm that computes a $(1 - \epsilon)\max\{\alpha', \beta\}$-MMS allocation.*

The proof of Theorem 5.12 utilizes the proof of Lemma 5.11 and is straightforward to prove. The currently best lower bound of γ_{ind} is $\frac{3}{4} + \frac{1}{12n}$ and the currently best guarantee for β is $3/4$; both are due to Garg and Taki [91]. Any better lower bound of γ_{ind} and any better guarantee for β found in the future would immediately imply a better MMS approximation guarantee in the mixed goods setting as well.

Part II

Fairness Versus Other Desideratum for Indivisible Goods

Chapter 6

Price of Fairness [†]

6.1 Introduction

While fairness is of great importance in resource allocation, it is obviously not the only objective of interest. An issue orthogonal to fairness is *efficiency*, or *social welfare*, which refers to the total happiness of the agents. A fundamental question is therefore how much efficiency we might lose if we want our allocation to be fair.

This question was first addressed independently by Bertsimas et al. [40] and Caragiannis et al. [67], who introduced the *price of fairness* concept to capture the efficiency loss due to fairness constraints. In particular, for any fairness notion and any given resource allocation instance with additive utilities, Caragiannis et al. defined the price of fairness of the instance to be the ratio between the maximum social welfare over all allocations and the maximum social welfare over allocations that are fair according to the notion. The overall price of fairness for this notion is then defined as the largest price of fairness across all instances. Caragiannis et al. considered three major fairness properties in the literature—envy-freeness, proportionality, and equitability, and presented a series of results on the price of fairness with respect to these notions; they assumed that the agents have additive utilities and each agent has utility 1 for the entire set of resources. As an example, they showed that for the allocation of indivisible goods among n agents, the price of proportionality is $n - 1 + 1/n$, meaning that the efficiency of the best proportional allocation can be a linear factor away from that of the best allocation overall.

Caragiannis et al.'s work sheds light on the trade-off between efficiency and fairness in the allocation of both divisible and indivisible resources. However, a significant limitation of their study is that while an allocation satisfying each of the three fairness notions always exists when goods are divisible, this is not the case for indivisible goods. Indeed, none of the notions can be satisfied in the simple instance with (at least) two agents and a single good to be allocated. Caragiannis et al. circumvented this issue by simply ignoring instances in which the fairness notion in question cannot be satisfied. As a result, their price of fairness analysis, which is meant to capture the worst-case efficiency loss, fails to cover certain sce-

[†]This chapter has been published in a paper by Bei, Lu, Manurangsi, and Suksompong [38].

narios that may arise in practice.[1] In addition, the fact that certain instances are not taken into account in the price of fairness have seemingly contradictory consequences. For example, since envy-free allocations are always proportional when utilities are additive, it may appear at first glance that the price of envy-freeness must be at least as high as the price of proportionality. This is not necessarily the case, however, because there are instances that admit proportional but no envy-free allocations.[2]

To address these limitations, in this chapter we study the price of fairness for indivisible goods with respect to fairness notions that can be satisfied in every instance. Among other notions, we consider envy-freeness up to one good (EF1), balancedness, maximum Nash welfare (MNW), maximum egalitarian welfare (MEW), and leximin; see Sections 2.2 and 6.2 for the formal definitions of these notions. In addition to deriving bounds on the price of fairness for these notions, we also introduce the concept of *strong price of fairness*, which captures the efficiency loss in the worst fair allocation as opposed to that in the best fair allocation. The strong price of fairness is relevant in settings where one is guaranteed an allocation satisfying some fairness notion but has no control over the particular allocation—for instance, we may be participating in an allocation exercise using an algorithm that guarantees EF1 or MNW, and wonder whether that fairness guarantee comes with any assurance on the social welfare. Indeed, certain fair division algorithms such as the *envy cycle elimination algorithm* [113] may output EF1 allocations with low efficiency; see the example in Theorem 6.6. The relationship between the price of fairness and the strong price of fairness is akin to that between the price of stability and the price of anarchy for equilibria. While the strong price of fairness is too demanding to yield any non-trivial guarantee for some fairness notions, as we will see, it does provide meaningful guarantees for other notions.

The majority of our results can be found in Table 6.1; we highlight a subset of these next. For the price of EF1, we provide a lower bound of $\Omega(\sqrt{n})$ and an upper bound of $O(n)$. We then show that two common methods for obtaining an EF1 allocation—the round-robin algorithm and MNW—have a price of fairness of linear order (for round-robin the price is exactly n), implying that these methods cannot be used to improve the upper bound for EF1. On the other hand, if we allow dependence on the number of goods m, the price of round-robin, and therefore the price of EF1, is $O(\sqrt{n}\log(mn))$—this means that the $\Omega(\sqrt{n})$ lower bound is almost tight unless the number of goods is huge compared to the number of agents. Our result illustrates a clear difference between EF1 and envy-freeness, as the price of the latter is $\Theta(n)$ [67]. For MNW, MEW, and leximin, we prove an asymptotically tight bound of $\Theta(n)$ on the price of fairness. Moreover, with the exception of EF1 and MNW, we establish exactly tight bounds in the case of two agents for all fairness notions—in particular, the price of EF1 is between 1.14 and 1.16, implying that there exists an EF1 allocation whose

[1] From the above example, one may think that such scenarios are rare exceptions. However, for envy-freeness, these scenarios are in fact common if the number of goods is not too large compared to the number of agents [78, 117].

[2] Indeed, the instance that Caragiannis et al. used to show that the price of proportionality is at least $n - 1 + 1/n$ admits no envy-free allocation. Thus, it is still possible that the price of envy-freeness is lower than the price of proportionality for indivisible goods.

	General n		$n = 2$	
EF1	LB: $\Omega(\sqrt{n})$ UB: $O(n)$	(Theorem 6.2) (Theorem 6.8)	LB: $8/7$ UB: $2/\sqrt{3}$	(Theorem 6.3) (Theorem 6.4)
EFX	—		$3/2$	(Theorem 6.5)
Round-robin	n	(Theorem 6.8)	2	(Theorem 6.8)
Balancedness	$\Theta(\sqrt{n})$	(Theorem 6.11)	$4/3$	(Theorem 6.12)
MNW	$\Theta(n)$	(Theorem 6.15)	LB: $27/23$ UB: $5/4$	(Theorem 6.17)
MEW	$\Theta(n)$	(Theorem 6.15)	$3/2$	(Theorem 6.18)
Leximin	$\Theta(n)$	(Theorem 6.15)	$3/2$	(Theorem 6.19)
PO	1 (Discussion at the beginning of Section 6.7)			

(a) Results for price of fairness.

	General n		$n = 2$	
EF1	∞ (Theorem 6.6)			
EFX	—		∞	(Theorem 6.6)
Round-robin	n^2	(Theorem 6.10)	4	(Theorem 6.10)
Balancedness	∞ (Theorem 6.13)			
MNW	$\Theta(n)$	(Theorem 6.15)	LB: $27/23$ UB: $5/4$	(Theorem 6.17)
MEW	∞ for $n \geq 3$	(Theorem 6.16)	$3/2$	(Theorem 6.18)
Leximin	$\Theta(n)$	(Theorem 6.15)	$3/2$	(Theorem 6.19)
PO	$\Theta(n^2)$	(Theorem 6.21)	3	(Theorem 6.22)

(b) Results for strong price of fairness.

Table 6.1: Summary of our results. LB denotes lower bound and UB denotes upper bound. We do not consider the (strong) price of EFX for general n because it is not known whether an EFX allocation always exists for $n > 3$. If we allow dependence on the number of goods m, we have an upper bound of $O(\sqrt{n}\log(mn))$ on the price of EF1; see Theorem 6.9.

welfare is close to the optimal welfare.

Our results point to round-robin as a promising allocation method: besides producing an EF1 allocation with high welfare, it is extremely simple and intuitive, and an allocation that it produces is always balanced.[3] Most of our upper bounds naturally give rise to polynomial-time algorithms for computing an allocation satisfying the fairness notion with the guaranteed welfare. However, there are two notable exceptions:[4] (i) the proof of Theorem 6.5 requires an agent to partition the goods into two bundles such that her utilities for the bundles are as equal as possible, an NP-hard problem; (ii) the upper bound in Theorem 6.10, which relies on Lemma 6.7, is based on a randomized approach and does not indicate how a desirable round-robin ordering can be efficiently chosen.

On the strong price of fairness front, we show via a simple instance that the strong price of EF1 and balancedness are infinite, meaning that there are arbitrarily bad EF1 and balanced allocations. Nevertheless, a round-robin allocation, which satisfies these two properties, al-

[3]Moreover, a round-robin allocation is likely to be envy-free and proportional as long as the number of goods is sufficiently larger than the number of agents [118].

[4]In addition to these exceptions, MNW, MEW, and leximin allocations are hard to compute regardless of price of fairness considerations; see, e.g., [132, Footnote 7].

ways has welfare within a factor n^2 of the optimal allocation, and this factor is exactly tight. For MNW and leximin, the strong price of fairness, like the price of fairness, is of linear order—hence, these two notions provide a better worst-case guarantee than the round-robin algorithm. However, while the price of MEW is also $\Theta(n)$, the strong price of MEW is infinite for $n \geq 3$ (and $3/2$ for $n = 2$), meaning that an MEW allocation does not provide any welfare guarantee when there are at least three agents. Finally, we consider Pareto optimality, for which the price of fairness is trivially 1, and show that the strong price of Pareto optimality is $\Theta(n^2)$. This demonstrates that an allocation that is optimal in the Pareto sense may be quite far from optimal with respect to social welfare.

6.1.1 Related Work

As we mentioned earlier, the price of fairness was introduced independently by Bertsimas et al. [40] and Caragiannis et al. [67]. Bertsimas et al. studied the concept for divisible goods with respect to fairness notions such as proportional fairness and max-min fairness; in particular, their results on proportional fairness imply that the price of EF and the price of MNW for divisible goods are both $\Theta(\sqrt{n})$.[5] Caragiannis et al. presented a number of bounds for both goods and chores, both when these items are divisible and indivisible. The price of fairness has subsequently been examined in several other settings, including for *contiguous* allocations of divisible goods [11], indivisible goods [150], and divisible chores [99], as well as in the context of machine scheduling [44] and budget division [125].

Typically, the price of fairness study focuses on quantifying the efficiency loss solely in terms of the number of agents. A notable exception to this is Kurz [109], who remarked that certain constructions used to establish worst-case bounds for indivisible goods require a large number of goods. Kurz investigated the dependence of the price of fairness on both the number of agents and the number of goods, and, as we do for the price of round-robin, found that the price indeed improves significantly if we limit the number of goods.

After the publication of the initial version of our work, Barman et al. [29] devised an algorithm that produces an allocation with social welfare within $O(\sqrt{n})$ of the optimum; together with our result, this implies that the price of EF1 is in fact $\Theta(\sqrt{n})$. Their algorithm works by starting with an optimal allocation, arranging the goods on a line so that each bundle in this allocation is connected, giving each agent her favourite good from her bundle, and updating the allocation by carefully assigning additional items so as to maintain EF1 and connectivity on the line. Moreover, their algorithm can be extended to the more general setting where agents have subadditive utilities.

6.2 Preliminaries

Consider an indivisible goods instance $\langle N, M, \mathcal{U} \rangle$, including the set of n agents N, the set of m indivisible goods M, and the agents' utility functions $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$, as described

[5] Interestingly, this stands in contrast to our result that the price of MNW for indivisible goods is $\Theta(n)$.

in Section 2.2. In this chapter, the agents' utilities are additive. Following Caragiannis et al. [67], we normalize the utilities across agents by assuming that $u_i(M) = 1$ for all $i \in N$.

A *property* P is a function that maps every instance I to a (possibly empty) set of allocations $P(I)$. Every allocation in $P(I)$ is said to satisfy property P.

We are now ready to define the price of fairness concepts.

Definition 6.1. For any given property P of allocations and any instance, we define the *price of* P for that instance to be the ratio between the optimal social welfare and the maximum social welfare over allocations satisfying P:

$$\text{Price of } P \text{ for instance } I = \frac{\mathsf{OPT}(I)}{\max_{\mathcal{M}\in P(I)} \mathsf{SW}(\mathcal{M})}.$$

The overall *price of* P is then defined as the supremum price of fairness across all instances.

Similarly, the *strong price of* P for a given instance is the ratio between the optimal social welfare and the *minimum* social welfare over allocations satisfying P:

$$\text{Strong price of } P \text{ for instance } I = \frac{\mathsf{OPT}(I)}{\min_{\mathcal{M}\in P(I)} \mathsf{SW}(\mathcal{M})}.$$

The overall *strong price of* P is then defined as the supremum price of fairness across all instances.

We will only consider properties P such that $P(I)$ is non-empty for every instance I, so the (strong) price of fairness is always well-defined. Specifically, we will consider the following fairness properties: EF1, EFX, round-robin, balancedness, as well as several welfare maximizers including MNW, MEW, and leximin; see definitions in Section 2.2. For EFX, the existence question is still unresolved [69, 135]. As such, we will only consider EFX in the case of two agents, for which existence is guaranteed [132].[6] Moreover, even though Pareto optimality (Definition 2.12) is an efficiency notion rather than a fairness notion, we also consider it in this chapter as it is a fundamental property in the context of resource allocation. Finally, with the exception of Theorem 6.9, we will be interested in the price of fairness as a function of n, and assume that m can be arbitrary.

6.3 Envy-Freeness Relaxations

In this section, we consider envy-freeness relaxations and begin with a general lower bound on the price of EF1.

Theorem 6.2. *The price of EF1 is* $\Omega(\sqrt{n})$.

Proof. Let $m = n, r = \lfloor\sqrt{n}\rfloor$, and assume that the utilities are as follows:

- For $i = 1, 2, \ldots, r-1$: $u_i((i-1)r + j) = \frac{1}{r}$ for $j = 1, 2, \ldots, r$, and $u_i(j) = 0$ otherwise.

[6] Recently, Chaudhury et al. [72] showed that the existence is also guaranteed for three agents.

- $u_r(j) = \frac{1}{n-r(r-1)}$ for $j = r(r-1)+1, \dots, n$, and $u_r(j) = 0$ otherwise.
- For $i = r+1, \dots, n$: $u_i(j) = \frac{1}{n}$ for all j.

Consider the allocation that assigns goods $ir-r+1, \dots, ir$ to agent i for $i = 1, \dots, r-1$ and the remaining goods to agent r. The social welfare of this allocation is r. On the other hand, in any EF1 allocation, each of the agents $i = r+1, \dots, n$ must receive at least one good—otherwise some agent would receive at least two goods and agent i would envy her. This means that the social welfare is at most $r \cdot \frac{1}{r} + (n-r) \cdot \frac{1}{n} < 2$. Hence the price of EF1 is at least $\frac{r}{2} = \frac{\lfloor \sqrt{n} \rfloor}{2}$. □

For two agents, we establish an almost tight bound on the price of EF1 and a tight bound on the price of EFX. We start with a lower bound for EF1.

Theorem 6.3. *For $n = 2$, the price of EF1 is at least $\frac{8}{7} \approx 1.143$.*

Proof. Let $m = 3$ and $0 < \epsilon < 1/6$, and assume that the utilities are as follows:

- $u_1(1) = 1/3 - 2\epsilon$, $u_1(2) = 1/3 + \epsilon$, $u_1(3) = 1/3 + \epsilon$;
- $u_2(1) = 0$, $u_2(2) = 1/2$, $u_2(3) = 1/2$.

The optimal social welfare is $4/3 - 2\epsilon$, achieved by assigning the first good to agent 1 and the last two goods to agent 2. However, in any EF1 allocation the last two goods cannot both be given to agent 2. Hence the social welfare of an EF1 allocation is at most $(1/3 - 2\epsilon) + (1/3 + \epsilon) + 1/2 = 7/6 - \epsilon$. Taking $\epsilon \to 0$, we find that the price of EF1 is at least $\frac{4/3}{7/6} = 8/7$. □

We now turn to the upper bound. In order to construct an EF1 allocation with high welfare, we proceed in a similar manner to the *adjusted winner* procedure [55], which is used to allocate divisible goods between two agents. Specifically, we arrange the goods according to the ratios between the utilities that they yield for the two agents—the idea is that the agents will then prefer goods at different ends. Roughly speaking, we then let the agent who obtains a lower utility in an optimal allocation choose a minimal set of goods for which she is EF1 starting from her end.

Theorem 6.4. *For $n = 2$, the price of EF1 is at most $\frac{2}{\sqrt{3}} \approx 1.155$.*

Proof. Consider an arbitrary instance. Sort the goods so that $\frac{u_1(1)}{u_2(1)} \geq \frac{u_1(2)}{u_2(2)} \geq \cdots \geq \frac{u_1(m)}{u_2(m)}$; goods x such that $u_2(x) = 0$ are put at the front and those with $u_1(x) = 0$ at the back, with arbitrary tie-breaking within each group of goods. Goods that yield zero value to both agents can be safely ignored since they have no effect on the optimal welfare or the maximum welfare of an EF1 allocation. For ease of notation, for any $1 \leq k \leq m$ we write $L(k) := \{1, 2, \dots, k\}$ and $R(k) := \{k, \dots, m\}$. We also define $L(0) = R(m+1) = \emptyset$.

Let $S_1 := \left\{ i \mid \frac{u_1(i)}{u_2(i)} > 1 \right\} = L(s)$ for some $0 \leq s \leq m$ and $S_2 := M \setminus S_1 = R(s+1)$. It is easy to see that $s < m$. If $s = 0$, both agents have identical utilities and the price of EF1

Algorithm 7: EF1-Two-Agents(N, M, u_1, u_2)

```
Assume that in an optimal allocation, agent 1 obtains no higher utility than agent 2.
  (Otherwise, reverse the roles of the two agents.)
Sort the goods so that u1(1)/u2(1) ≥ u1(2)/u2(2) ≥ ··· ≥ u1(m)/u2(m).
for k = 1, 2, ..., m do
    L(k) ← {1, ..., k}
    R(k) ← {k, ..., m}
s ← 0
while s < m and u1(s+1)/u2(s+1) > 1 do s ← s + 1
f ← s
while u1(L(f)) < u1(R(f + 2)) do f ← f + 1
return (L(f), R(f + 1))
```

is 1, so we may assume that $s > 0$. The allocation $\mathcal{S} = (S_1, S_2)$ is an optimal allocation, and the optimal social welfare is $u_1(S_1) + u_2(S_2)$. Without loss of generality, assume that $u_1(S_1) \leq u_2(S_2)$. Note that we must have $u_2(S_2) \geq 1/2$, since otherwise both $u_1(S_1)$ and $u_2(S_2)$ are smaller than $1/2$ and switching S_1 and S_2 would yield a higher social welfare. We can further assume that $u_1(S_1) < 1/2$, because otherwise $\mathcal{S}$ is also an EF1 allocation and the price of EF1 is 1.

Next, we describe how to obtain a particular EF1 allocation $\mathcal{F}$. Let f be the smallest index such that $f \geq s$ and $u_1(L(f)) \geq u_1(R(f+2))$. Clearly, $f < m$. In the allocation $\mathcal{F} = (F_1, F_2)$, we assign the goods $F_1 := L(f)$ to agent 1, and $F_2 := R(f+1)$ to agent 2. The pseudocode for computing $\mathcal{F}$ is presented as Algorithm 7. See also Figure 6.1.

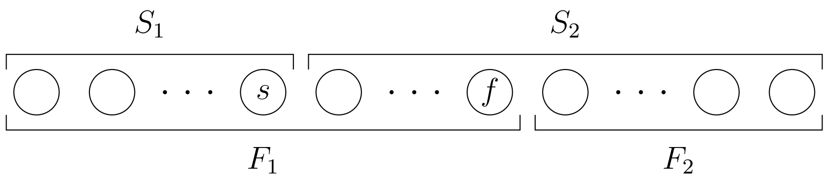


Figure 6.1: Illustration for the proof of Theorem 6.4.

Allocation $\mathcal{F}$ satisfies EF1 The EF1 condition is satisfied for agent 1, because $u_1(F_1) \geq u_1(F_2 \setminus \{f+1\})$ by definition.

For agent 2, since f is the smallest index such that $f \geq s$ and $u_1(L(f)) \geq u_1(R(f+2))$, we have either $f = s$ or $u_1(L(f-1)) < u_1(R(f+1))$. If $f = s$, then $\mathcal{F}$ coincides with the optimal allocation $\mathcal{S}$, and $u_2(F_2) = u_2(S_2) \geq 1/2$. Clearly EF1 is satisfied. Else, $f > s$, and we have $0 < u_1(L(f-1)) < u_1(R(f+1))$. Note also that $u_2(R(f+1)) > 0$. Therefore,

$$\frac{u_1(L(f-1))}{u_2(L(f-1))} \geq \frac{u_1(f-1)}{u_2(f-1)} \geq \frac{u_1(f+1)}{u_2(f+1)} \geq \frac{u_1(R(f+1))}{u_2(R(f+1))},$$

where we take a fraction to be infinite if it has denominator 0.[7] (None of the fractions can have both numerator and denominator 0.) Since $u_1(L(f-1)) < u_1(R(f+1))$, this implies

[7] To see the first and third inequalities, one may prove by induction that in general, if we have $\frac{a_1}{b_1} \geq \cdots \geq$

that

$$\frac{u_2(L(f-1))}{u_2(R(f+1))} \leq \frac{u_1(L(f-1))}{u_1(R(f+1))} < 1.$$

Thus,

$$u_2(F_2) = u_2(R(f+1)) > u_2(L(f-1)) = u_2(F_1 \setminus \{f\}),$$

implying that EF1 is again satisfied.

The price of EF1 for this instance is at most $\frac{2}{\sqrt{3}}$ Now we analyze the social welfare of the allocation $\mathcal{F}$ and compare it to the optimal social welfare.

If $f = s$, the price of EF1 is 1. Assume from now on that $f > s$. We have $u_1(F_2) > u_1(L(f-1)) \geq u_1(L(s)) = u_1(S_1)$ and $\frac{u_1(S_2)}{u_2(S_2)} \geq \frac{u_1(F_2)}{u_2(F_2)}$. Since $u_1(F_2) > 0$, we also have $u_1(S_2) > 0$. Moreover, $u_2(F_2), u_2(S_2) > 0$. Thus,

$$\begin{aligned}
u_1(F_1) + u_2(F_2) &\geq (1 - u_1(F_2)) + \frac{u_1(F_2)u_2(S_2)}{u_1(S_2)} \\
&= 1 + \left(\frac{u_2(S_2)}{u_1(S_2)} - 1\right) u_1(F_2) \\
&> 1 + \left(\frac{u_2(S_2)}{u_1(S_2)} - 1\right) u_1(S_1) \\
&= 1 - u_1(S_1) + \frac{u_2(S_2)}{u_1(S_2)} \cdot u_1(S_1) \\
&= 1 - u_1(S_1) + \frac{u_2(S_2)}{1 - u_1(S_1)} \cdot (1 + (u_1(S_1) - 1)) \\
&= 1 - u_1(S_1) + \frac{u_2(S_2)}{1 - u_1(S_1)} - u_2(S_2).
\end{aligned}$$

Therefore the ratio between the optimal social welfare and the social welfare of $\mathcal{F}$ is

$$\alpha := \frac{u_1(S_1) + u_2(S_2)}{u_1(F_1) + u_2(F_2)} < \frac{u_1(S_1) + u_2(S_2)}{\frac{u_2(S_2)}{1-u_1(S_1)} + 1 - u_2(S_2) - u_1(S_1)}.$$

We further analyze the last expression. First, taking its partial derivative with respect to $u_2(S_2)$ gives

$$\frac{(1 - u_1(S_1))(1 - 2u_1(S_1))}{(u_1(S_1)^2 + u_1(S_1)(u_2(S_2) - 2) + 1)^2},$$

which is always positive when $u_1(S_1) < 1/2$. This shows that the last expression is monotone increasing in $u_2(S_2)$. Thus

$$\alpha < \frac{u_1(S_1) + 1}{\frac{1}{1-u_1(S_1)} - u_1(S_1)}.$$

Finally, this expression is maximized when $u_1(S_1) = 2 - \sqrt{3}$ and yields a value of $\frac{2}{\sqrt{3}}$, completing the proof. $\square$

$\frac{a_k}{b_k}$, then $\frac{a_1}{b_1} \geq \frac{a_1 + \cdots + a_k}{b_1 + \cdots + b_k} \geq \frac{a_k}{b_k}$. The case $k = 2$ holds because $\frac{a_1}{b_1} \geq \frac{a_1+a_2}{b_1+b_2}$ is equivalent to $\frac{a_1}{b_1} \geq \frac{a_2}{b_2}$, and similarly for $\frac{a_1+a_2}{b_1+b_2} \geq \frac{a_2}{b_2}$.

The gap on the price of EF1 between Theorems 6.3 and 6.4 is only approximately 0.01. For EFX, we establish a tight bound in the case of two agents.

Theorem 6.5. *For $n = 2$, the price of EFX is $3/2$.*

Proof. **Lower bound:** Let $m = 3$ and $0 < \epsilon < 1/2$, and assume that the utilities are as follows:

- $u_1(1) = 1/2 + \epsilon$, $u_1(2) = 1/2 - \epsilon$, $u_1(3) = 0$;
- $u_2(1) = 1/2 + \epsilon$, $u_2(2) = 0$, $u_2(3) = 1/2 - \epsilon$.

The optimal social welfare is $3/2-\epsilon$, achieved by assigning the first two goods to agent 1 and the last good to agent 2. On the other hand, in any EFX allocation, no agent can get both of the goods that they positively value. Hence, the social welfare of an EFX allocation is at most 1. Taking $\epsilon \to 0$, we find that the price of EFX is at least $3/2$.

Upper bound: Consider an arbitrary instance. If in an optimal allocation both agents get utility at least $1/2$, this allocation is also envy-free and hence EFX, so the price of EFX is 1. Otherwise, the maximum welfare is at most $1 + 1/2 = 3/2$. Now we show that there always exists an EFX allocation with social welfare at least 1; this immediately yields the desired bound.

Let the first agent partition the goods into two bundles such that her values for the bundles are as equal as possible. Denote by x and $1 - x$ the values of the two bundles, where $x \geq 1 - x$. Suppose that all goods of zero value, if any, are in the second bundle. Let $y \geq 1 - y$ be the corresponding values for the second agent, and assume without loss of generality that $y \geq x$. Consider the partition of the first agent, and assume that the two bundles yield value z and $1 - z$ to the second agent, respectively. If $z \leq 1 - z$, by assigning the first bundle to the first agent and the second bundle to the second agent, we have an envy-free allocation with welfare at least 1. Else, $z \geq 1 - z$. By definition of y, we also have $z \geq y \geq x$. We assign the first bundle to the second agent and the second bundle to the first agent. The second agent is clearly envy-free. If the first agent still has envy after removing some good i from the first bundle, then by moving good i to the second bundle, we create a more equal partition, a contradiction. Hence the allocation is EFX to the first agent. The social welfare of this allocation is $z + (1 - x) \geq 1$. □

Next, we give a simple instance showing that EF1 and EFX allocations can have arbitrarily bad welfare.

Theorem 6.6. *The strong price of EF1 is ∞. For $n = 2$, the strong price of EFX is ∞.*

Proof. Let $m = n$, and assume that $u_i(i) = 1$ for all i and $u_i(j) = 0$ otherwise. The allocation that assigns good i to agent i for every i has social welfare n. On the other hand, the allocation that assigns good $i - 1$ to agent i for $i = 2, \ldots, n$ and good n to agent 1 is EF1 and EFX, but has social welfare 0. The conclusion follows. □

6.4 Round-Robin Algorithm

We now turn our attention to the round-robin algorithm, which always produces an EF1 allocation. We show that it is always possible to order the agents to obtain a welfare of 1.

Lemma 6.7. *For any instance, there exists an ordering of the agents such that the round-robin algorithm implemented with this ordering produces an allocation with social welfare at least 1, and this bound is tight.*

Proof. We claim that if we choose the ordering of the agents uniformly at random, the expected social welfare is at least 1. The desired bound immediately follows from this claim.

To prove the claim, consider an arbitrary agent i, and assume without loss of generality that $u_i(1) \geq u_i(2) \geq \cdots \geq u_i(m)$. Note that if the agent is ranked j-th in the ordering, her utility is at least $u_i(j) + u_i(n+j) + u_i(2n+j) + \cdots + u_i(kn+j)$, where $k = \lfloor (m-j)/n \rfloor$. Hence, the agent's expected utility is at least

$$\frac{1}{n} \cdot \sum_{j=1}^{n} \sum_{r=0}^{\lfloor (m-j)/n \rfloor} u_i(rn+j) = \frac{1}{n} \cdot \sum_{j=1}^{m} u_i(j) = \frac{1}{n}.$$

It follows from the linearity of expectation that the expected social welfare is at least $n \cdot \frac{1}{n} = 1$, as claimed.

The tightness of the bound follows from the instance where every agent has utility 1 for the same good. □

Lemma 6.7 yields a linear price of fairness for round-robin.

Theorem 6.8. *The price of round-robin is n. Consequently, the price of EF1 is at most n.*

Proof. **Upper bound:** Consider an arbitrary instance. Since every agent receives utility at most 1, the optimal social welfare is at most n. On the other hand, by Lemma 6.7, there exists an ordering of the agents such that the round-robin algorithm yields welfare at least 1. Hence the price of round-robin is at most n.

Lower bound: Let $m = x^n$ for some large x that is divisible by n, and assume that the utilities are such that for each agent i, $u_i(j) = 1/x^i$ for $j = 1, 2, \ldots, x^i$ and $u_i(j) = 0$ otherwise.

Consider the allocation that assigns goods $1, 2, \ldots, x$ to agent 1, and $x^{i-1}+1, \ldots, x^i$ to agent i for every $i \geq 2$. In this allocation, agent 1 gets utility 1, while each remaining agent gets utility $(x^i - x^{i-1})/x^i = 1 - 1/x$. The social welfare is therefore $n - (n-1)/x$. This converges to n for large x.

On the other hand, consider the round-robin algorithm with an arbitrary ordering of the agents, and assume without loss of generality that agents always break ties in favour of goods with lower numbers. Hence, regardless of the ordering, the goods get chosen in the order $1, 2, \ldots, m$. As a result, every agent gets exactly $1/n$ of their valued goods, so her utility is $1/n$, and the social welfare is 1. Hence the price of round-robin is n. □

The argument for the lower bound in Theorem 6.8 works even if we can choose a new ordering of the agents in every round. This means that the fixed order is not a barrier to obtaining a better price of fairness, but rather the "each agent picks exactly once in every round" aspect of the algorithm.

One may notice that the lower bound construction uses an exponential number of goods. This is in fact necessary to obtain an instance with a high price of round-robin. As we show next, the $\Omega(\sqrt{n})$ lower bound on the price of EF1 is almost tight as long as m is not too large compared to n. At a high level, our proof proceeds by considering an optimal allocation and choosing a range $\left[2^{-\ell-1}, 2^{-\ell}\right]$ that the largest number of agents' utilities for goods in this allocation fall into. In the case where a sufficiently large number of goods correspond to this range, we may choose an arbitrary round-robin ordering—we can lower bound the welfare resulting from the round-robin algorithm by observing that as long as we have not run out of goods from this range with respect to an agent, every pick must give the agent a utility at least the minimum utility that the agent obtains from this range. On the other hand, if only a small number of goods belong to this range, we need to be more careful in choosing the ordering.

Theorem 6.9. *The price of round-robin is $O(\sqrt{n}\log(mn))$. Consequently, the price of EF1 is $O(\sqrt{n}\log(mn))$.*

Proof. Consider any instance I. First, observe that if $\mathsf{OPT}(I) \leq 65\sqrt{n}\log_2(mn)$, then Lemma 6.7 immediately yields the desired result. We therefore focus on the case where $\mathsf{OPT}(I) > 65\sqrt{n}\log_2(mn)$. We claim that there exists an ordering for which the round-robin algorithm produces an allocation with social welfare at least $\frac{\mathsf{OPT}(I)}{65\sqrt{n}\log_2(mn)}$.

Fix an optimal allocation $\mathcal{M} = (M_1, M_2, \ldots, M_n)$, and let $r := \lceil \log_2(m\sqrt{n}) \rceil$. For each $i \in N$, let us partition M_i into $M_i^0 \cup M_i^1 \cup \cdots \cup M_i^r$, where M_i^ℓ is defined by

$$M_i^\ell = \begin{cases} \left\{ j \in M_i \mid u_i(j) \in \left(2^{-\ell-1}, 2^{-\ell}\right] \right\} & \text{if } \ell \neq r; \\ \left\{ j \in M_i \mid u_i(j) \in \left[0, 2^{-\ell}\right] \right\} & \text{if } \ell = r. \end{cases}$$

Furthermore, define $M^\ell := \bigcup_{i=1}^n M_i^\ell$ and $\mathsf{SW}_\ell(\mathcal{M}) := \sum_{i=1}^n u_i(M_i^\ell)$.

Let $\ell^* := \arg\max_{\ell \in \{0,\ldots,r-1\}} \mathsf{SW}_\ell(\mathcal{M})$. We have

$$\mathsf{SW}_{\ell^*}(\mathcal{M}) \geq \frac{1}{r}\left(\sum_{\ell=0}^{r-1} \mathsf{SW}_\ell(\mathcal{M})\right) = \frac{\mathsf{OPT}(I) - \mathsf{SW}_r(\mathcal{M})}{r}.$$

However, since agent i values each item in M_i^r at most $2^{-r} \leq \frac{1}{m\sqrt{n}}$, we have $u_i(M_i^r) \leq 1/\sqrt{n}$. This implies that $\mathsf{SW}_r(\mathcal{M}) \leq \sqrt{n}$, which is no more than $\mathsf{OPT}(I)/65$. Hence,

$$\mathsf{SW}_{\ell^*}(\mathcal{M}) \geq \frac{64}{65r} \cdot \mathsf{OPT}(I) \geq \frac{32 \cdot \mathsf{OPT}(I)}{65\log_2(mn)}. \tag{6.1}$$

Thus, it suffices to show the existence of an ordering such that round-robin produces an allocation with social welfare at least $\mathsf{SW}_{\ell^*}(\mathcal{M})/\sqrt{n}$.

Observe that Equation (6.1) implies that $\mathsf{SW}_{\ell^*}(\mathcal{M}) > 32\sqrt{n}$. We now consider two cases, based on $T := \left|M^{\ell^*}\right|$. Since $u_i(M_i^{\ell^*}) \leq 2^{-\ell^*}\left|M_i^{\ell^*}\right|$ for each i, we have $\mathsf{SW}_{\ell^*}(\mathcal{M}) \leq 2^{-\ell^*}T$.

Case 1: $T > 2n$. In this case, we will show that the round-robin algorithm with arbitrary ordering yields an allocation with social welfare at least $\mathsf{SW}_{\ell^*}(\mathcal{M})/\sqrt{n}$.

To see this, let us consider the round-robin procedure with arbitrary ordering, and consider the set of goods that are picked in the first $t := \lfloor T/(2n) \rfloor$ rounds; let $S_t \subseteq M$ denote this set. Now, observe that $\sum_{i=1}^n \left|M_i^{\ell^*} \setminus S_t\right| \geq T - |S_t| = T - n \cdot t \geq T/2$. This implies that

$$\sum_{i=1}^n u_i\left(M_i^{\ell^*} \setminus S_t\right) \geq \frac{T}{2} \cdot 2^{-\ell^*-1} \geq \frac{\mathsf{SW}_{\ell^*}(\mathcal{M})}{4} > 8\sqrt{n}.$$

Since $u_i\left(M_i^{\ell^*} \setminus S_t\right) \leq 1$, there must be more than $8\sqrt{n}$ agents such that $M_i^{\ell^*} \not\subseteq S_t$. Let N^* denote the set of such agents.

We claim that, in each of the first t rounds, every agent $i \in N^*$ must receive an item she values at least $2^{-\ell^*-1}$. The reason is that agent i picks her favourite good, which she must value at least as much as the good(s) left unpicked in $M_i^{\ell^*} \setminus S_t$. Moreover, she values the latter at least $2^{-\ell^*-1}$, so this must also be a lower bound of her utility for the former.

From the claim in the previous paragraph, we can conclude that the social welfare of the allocation produced is at least

$$|N^*| \cdot t \cdot 2^{-\ell^*-1} > 8\sqrt{n} \cdot \frac{T}{4n} \cdot 2^{-\ell^*-1} \geq \frac{\mathsf{SW}_{\ell^*}(\mathcal{M})}{\sqrt{n}}$$

as desired. Note that we use the assumption $T > 2n$ to conclude that $t \geq T/(4n)$ in the first inequality above.

Case 2: $T \leq 2n$. In this case, we will show that if we choose the ordering π in a careful manner, then the social welfare obtained in the first round alone already suffices.

Similarly to Case 1, observe that since $\sum_{i=1}^n u_i\left(M_i^{\ell^*}\right) = \mathsf{SW}_{\ell^*}(\mathcal{M}) > 8\sqrt{n}$, there are more than $8\sqrt{n}$ agents i whose $M_i^{\ell^*}$ is non-empty. Let N^* denote the set of such agents.

We will construct the ordering π step-by-step as follows. For $k = 1, 2, \ldots, \lceil 4\sqrt{n} \rceil$, we let $\pi(k)$ be any agent i such that (1) i is not yet in the ordering, and (2) not all goods in $M_i^{\ell^*}$ are already picked by $\pi(1), \ldots, \pi(k-1)$. Note that such an agent exists because, at each step k, at most two candidate agents become invalid: the agent $i = \pi(k)$, and the agent i' whose good in $M_{i'}^{\ell^*}$ is picked by $\pi(k)$. Since we start with $8\sqrt{n}$ valid candidates, even after $\lceil 4\sqrt{n} \rceil - 1$ steps, there are still valid candidate agents to be chosen from.

The remainder of the ordering can be chosen arbitrarily. We now argue that the resulting round-robin allocation has the desired social welfare. To see this, for $k = 1, \ldots, \lceil 4\sqrt{n} \rceil$, observe that agent $\pi(k)$ must pick a good that is worth at least $2^{-\ell^*-1}$ to her in the first round, since not all goods in $M_{\pi(k)}^{\ell^*}$ have been picked. As a result, the social welfare is at least

$$(4\sqrt{n}) \cdot 2^{-\ell^*-1} \geq (2T/\sqrt{n}) \cdot 2^{-\ell^*-1} \geq \frac{\mathsf{SW}_{\ell^*}(\mathcal{M})}{\sqrt{n}},$$

where the first inequality follows from $T \leq 2n$. $\square$

While Theorem 6.9 shows that the price of EF1 is close to $\Theta(\sqrt{n})$ unless the number of goods is huge, if we are only interested in the dependence on the number of agents, the gap still remains between $\Omega(\sqrt{n})$ and $O(n)$.

We end this section by establishing an exact bound on the strong price of round-robin.

Theorem 6.10. *The strong price of round-robin is n^2.*

Proof. **Upper bound:** Consider an arbitrary instance. Since every agent receives utility at most 1, the optimal social welfare is at most n. On the other hand, in the round-robin algorithm, the first agent gets to choose an item ahead of all other agents in every round and therefore does not envy any other agent in the resulting allocation. This implies that her utility, and hence the social welfare, is at least $1/n$. It follows that the strong price of round-robin is at most n^2.

Lower bound: Let m be a large number divisible by n, and assume that the utilities are as follows:

- $u_1(i) = \frac{1}{m}$ for all i;
- for $i = 2, \ldots, n$: $u_i(i-1) = 1$, and $u_i(j) = 0$ otherwise.

Consider the allocation that assigns good $i-1$ to agent i for every $i = 2, \ldots, n$, and the remaining goods to agent 1. In this allocation, every agent $i \geq 2$ receives utility 1. Agent 1 receives utility $\frac{m-n+1}{m}$, which converges to 1 for large m. Therefore, the social welfare converges to n.

On the other hand, consider the round-robin algorithm with the ordering of the agents $1, 2, \ldots, n$, and assume without loss of generality that agents always break ties in favour of goods with lower numbers. The first agent gets utility exactly $1/n$, while the remaining agents get zero utility since their only valuable good is "stolen" by the agent before them in the first round. Hence the social welfare is $1/n$. This means that the strong price of round-robin is n^2, as desired. $\square$

6.5 Balancedness

In this section, we consider balancedness. We begin by establishing an asymptotically tight bound on the price of balancedness.

Theorem 6.11. *The price of balancedness is $\Theta(\sqrt{n})$.*

Proof. Intuitively, for the upper bound, we divide the agents into two groups according to whether they receive a sufficiently large number of goods in an optimal allocation or not. There cannot be too many agents in the first group, and therefore they cannot make a significant contribution to the optimal welfare, so we may ignore them. For agents in the second group, we let each of them keep some number of goods that they like most; we choose this number so that it is possible to redistribute the remaining goods and obtain a balanced allocation.

Lower bound: Consider the instance in Theorem 6.2. The social welfare can be as high as $r = \lfloor\sqrt{n}\rfloor$, while a similar argument shows that the social welfare of any balanced allocation is at most 2. The conclusion follows.

Upper bound: If $\mathsf{OPT}(I) \leq 4\sqrt{n}$, the result follows immediately from Lemma 6.7. We therefore assume that $\mathsf{OPT}(I) > 4\sqrt{n}$. We claim that for any instance I, the maximum social welfare of a balanced allocation is always within a factor $4\sqrt{n}$ of the optimal social welfare; this claim implies the desired upper bound. In fact, we will show that there is a balanced allocation $\mathcal{M}$ such that $\mathsf{SW}(\mathcal{M}) \geq \frac{\mathsf{OPT}(I)-\sqrt{n}}{2\sqrt{n}}$; this suffices for our claim because $\frac{\mathsf{OPT}(I)-\sqrt{n}}{2\sqrt{n}} \geq \frac{\mathsf{OPT}(I)}{4\sqrt{n}}$. We consider two cases.

Case 1: $m \geq n$. Fix an optimal allocation, and let A be the set of agents who receive at least $\frac{m}{\sqrt{n}}$ goods in the optimal allocation, and B the complement set of agents. Since there are at most $\sqrt{n}$ agents in A, they contribute at most $\sqrt{n}$ to $\mathsf{OPT}(I)$, so the agents in B contribute at least $\mathsf{OPT}(I)-\sqrt{n}$. We let each agent in B keep her $\lceil\frac{m}{2n}\rceil$ most valuable goods (or all of her goods, if she has fewer than this number of goods). Note that each such agent keeps at least a $\lceil\frac{m}{2n}\rceil/\frac{m}{\sqrt{n}} \geq \frac{1}{2\sqrt{n}}$ fraction of her goods. Since the agents in B originally have a total utility of at least $\mathsf{OPT}(I) - \sqrt{n}$, the utility of the kept goods is at least $\frac{\mathsf{OPT}(I)-\sqrt{n}}{2\sqrt{n}}$. Moreover, since $\lceil\frac{m}{2n}\rceil \leq \lfloor\frac{m}{n}\rfloor$ due to the assumption $m \geq n$, the remaining goods can be reallocated to obtain a balanced allocation, which has social welfare at least $\frac{\mathsf{OPT}(I)-\sqrt{n}}{2\sqrt{n}}$.

Case 2: $m < n$. Fix an optimal allocation, and let A be the set of agents who receive at least $\sqrt{n}$ goods in the optimal allocation, and B the complement set of agents. Since there are at most $\sqrt{n}$ agents in A, they contribute at most $\sqrt{n}$ to $\mathsf{OPT}(I)$, so the agents in B contribute at least $\mathsf{OPT}(I) - \sqrt{n}$. We let each agent in B keep her most valuable good (if she receives at least one good). By a similar reasoning as in Case 1, this yields a total utility of at least $\frac{\mathsf{OPT}(I)-\sqrt{n}}{\sqrt{n}}$. The remaining goods can be reallocated to obtain a balanced allocation, which has social welfare at least $\frac{\mathsf{OPT}(I)-\sqrt{n}}{\sqrt{n}} \geq \frac{\mathsf{OPT}(I)-\sqrt{n}}{2\sqrt{n}}$. □

For two agents, we give an exact bound on the welfare that can be lost due to imposing balancedness.

Theorem 6.12. *For $n = 2$, the price of balancedness is $4/3$.*

Proof. **Lower bound:** Let m be a large even number, and assume that the utilities are as follows:

- $u_1(1) = 1$ and $u_1(i) = 0$ otherwise;
- $u_2(i) = \frac{1}{m}$ for all i.

Consider the allocation that assigns the first good to the first agent and the remaining goods to the second agent. The social welfare is $1 + (1 - 1/m)$, which converges to 2 for large m. On the other hand, in any balanced allocation, the first agent gets utility at most 1 while the second agent gets utility $\frac{m}{2} \cdot \frac{1}{m} = \frac{1}{2}$, so the social welfare is at most $3/2$. Hence the price of balancedness is at least $4/3$.

Upper bound: Consider an arbitrary instance. If m is odd, we may add a dummy good that yields zero utility to both agents—this does not change the optimal social welfare or the maximum social welfare of a balanced allocation. We may therefore assume that m is even.

Sort the goods so that $u_1(1) - u_2(1) \geq u_1(2) - u_2(2) \geq \cdots \geq u_1(m) - u_2(m)$. Let s be the last good such that $u_1(s) - u_2(s) \geq 0$, and assume without loss of generality that $s \geq m/2$. An optimal allocation assigns the set of goods $S_1 = \{1, 2, \ldots, s\}$ to the first agent and the complement set S_2 to the second agent, yielding social welfare $u_1(S_1) + u_2(S_2) = u_1(S_1) + (1 - u_2(S_1)) = 1 + \Delta$, where $\Delta := u_1(S_1) - u_2(S_1) \geq 0$. On the other hand, consider the balanced allocation that assigns goods $1, 2, \ldots, m/2$ to the first agent and the remaining goods to the second agent. Note that at most half of the goods in S_1 are reallocated to the second agent, and these are the goods with the lowest difference in utility between the two agents. Hence, the utility loss going from the first to the second allocation is at most $\Delta/2$, implying that the social welfare of the second allocation is at least $1 + \frac{\Delta}{2}$. The price of balancedness is therefore at most

$$\sup_{0 \leq \Delta \leq 1} \frac{1 + \Delta}{1 + \frac{\Delta}{2}}.$$

This ratio is increasing in Δ and reaches the maximum at $\Delta = 1$, where its value is $4/3$, completing the proof. □

Finally, the same construction as in Theorem 6.6 shows that balanced allocations can have arbitrarily bad welfare.

Theorem 6.13. *The strong price of balancedness is* ∞.

6.6 Welfare Maximizers

In this section, we consider allocations that maximize different measures of welfare. To start with, we show that every MNW and leximin allocation yields a decent welfare.

Lemma 6.14. *For any instance, every MNW allocation and every leximin allocation has social welfare at least* 1*, and both bounds are tight.*

Proof. We first establish the bound for MNW. Consider any MNW allocation where agent i receives bundle M_i, and assume for contradiction that $\sum_{k=1}^{n} u_k(M_k) < 1$. Fix any agent i. Since the agent has utility 1 for the entire set of items, we have $\sum_{k=1}^{n} u_i(M_k) = 1$. If

$u_i(M_k) \leq u_k(M_k)$ for all $k = 1, 2, \ldots, n$, we would have

$$1 = \sum_{k=1}^{n} u_i(M_k) \leq \sum_{k=1}^{n} u_k(M_k) < 1,$$

a contradiction, so there exists $j \neq i$ such that $u_i(M_j) > u_j(M_j)$. Construct a directed graph with vertices $1, 2, \ldots, n$, and add an edge from i to j if $u_i(M_j) > u_j(M_j)$. From the above observation, every vertex has at least one outgoing edge, implying that the graph consists of a directed cycle. For every edge $i \to j$ in the cycle, we give M_j to agent i instead of agent j. If we consider the change in the multiset of the n utilities between the old and new allocations, at least one number increases while others remain the same. This means that either we have decreased the number of agents who get zero utility, or keep this number fixed and increase the product of utilities of the agents who get nonzero utility. Either case contradicts the definition of an MNW allocation.

To show the bound for leximin, we apply the same argument. An improvement in the multiset of utilities as described in the last step contradicts the definition of leximin.

Finally, the tightness of the bounds follows from the instance where every agent has utility 1 for the same good. □

Lemma 6.14 allows us to show that the price of MNW and the strong price of MNW, the price of MEW, and both prices of leximin are of linear order.

Theorem 6.15. *The price of MNW, the strong price of MNW, the price of MEW, the price of leximin, and the strong price of leximin are* $\Theta(n)$.

Proof. We start with MNW. It suffices to show that the price of MNW is $\Omega(n)$ and the strong price of MNW is $O(n)$.

Lower bound: Let $m = n$ and $0 < \epsilon < 1$, and assume that the utilities are as follows:

- $u_1(1) = 1$ and $u_1(j) = 0$ otherwise;
- for $i = 2, \ldots, n$: $u_i(i-1) = 1 - \epsilon$, $u_i(i) = \epsilon$, and $u_i(j) = 0$ otherwise.

Consider the allocation that assigns good $i-1$ to agent i for $i = 2, \ldots, n$, and good n to agent 1. The social welfare of this allocation is $(n-1)(1-\epsilon)$. On the other hand, the unique MNW allocation assigns good i to agent i for every i. The social welfare of this allocation is $1 + (n-1)\epsilon$. Taking $\epsilon \to 0$, we find that the price of MNW is $\Omega(n)$.

Upper bound: Consider an arbitrary instance. Since every agent receives utility at most 1, the optimal social welfare is at most n. On the other hand, by Lemma 6.14, the social welfare of any MNW allocation is at least 1. The conclusion follows.

Notice that in the lower bound instance above, the unique MNW allocation is also the unique MEW allocation as well as the unique leximin allocation. This means that the price of MEW, the price of leximin, and the strong price of leximin are all $\Omega(n)$.

It remains to show that the price of MEW and the strong price of leximin are $O(n)$. For leximin, this follows from Lemma 6.14 and the fact that the optimal social welfare is at most n. We claim that for any instance, there exists a MEW allocation with social welfare at least 1. To prove this claim, we apply the same argument as in Lemma 6.14, but starting with a MEW allocation with maximum social welfare. An improvement in the multiset of utilities as described in the argument does not decrease the egalitarian welfare and strictly increases the social welfare, which gives us the desired contradiction. □

Surprisingly, MEW allocations can be arbitrarily bad when there are at least three agents.

Theorem 6.16. *For $n > 2$, the strong price of MEW is infinite.*

Proof. Let $m = n$, and assume that the utilities are as follows:

- $u_1(1) = 1$ and $u_1(j) = 0$ otherwise;
- for $i = 2, \ldots, n$: $u_i(i-1) = 1$ and $u_i(j) = 0$ otherwise.

Observe that in any allocation, some agent does not get a desired good. This means that every allocation has egalitarian welfare 0, and all allocations are MEW. Now, there exists an allocation with social welfare 0, for example the allocation that assigns good $i+1$ to agent i for $i = 1, 2, \ldots, n-1$, and assigns good 1 to agent n. Since there also exists an allocation with positive social welfare, the strong price of MEW is infinite. □

We now turn to the case of two agents. For MNW, we establish almost tight bounds on both prices of fairness.

Theorem 6.17. *For $n = 2$, the price of MNW and the strong price of MNW are at least $27/23 \approx 1.174$ and at most $5/4 = 1.25$.*

Proof. It suffices to show that the price of MNW is at least $27/23$ and the strong price of MNW is at most $5/4$.

Lower bound: Let $m = 3$ and $0 < \epsilon < 1/7$, and assume that the utilities are as follows:

- $u_1(1) = 2/3$, $u_1(2) = 1/3$, $u_1(3) = 0$;
- $u_2(1) = 4/7 - \epsilon$, $u_2(2) = 1/7 + \epsilon$, $u_2(3) = 2/7$.

The optimal social welfare is $9/7$, obtained by assigning the first two goods to the first agent and the last good to the second agent. On the other hand, one can check that the maximum Nash welfare is $2/7 + 2\epsilon/3$, obtained (uniquely) by assigning the first good to the first agent and the last two goods to the second agent. This allocation yields social welfare $23/21 + \epsilon$. Taking $\epsilon \to 0$, we find that the price of MNW is at least $27/23$.

Upper bound: Consider an arbitrary instance. Suppose that the optimal social welfare is x. If $x \leq 5/4$, then Lemma 6.14 immediately implies that the price of MNW of this instance is at most $5/4$.

We now focus on the case where $x \geq 5/4$. Let us assume further that, in an optimal allocation, the first agent has utility x_1 and the second has utility x_2, where $x_1 \geq x_2$ and $x_1 + x_2 = x$. Since $x_1 \leq 1$, we have $x_1/x_2 \leq 1/(x-1) \leq 4$. Next, consider any MNW allocation. Suppose that in this allocation the first agent has utility y_1 and the second has utility y_2. Since the Nash welfare of this allocation must be at least that of the optimal allocation, we have $y_1 y_2 \geq x_1 x_2$. As a result, the social welfare of this allocation is $y_1 + y_2 \geq 2\sqrt{y_1 y_2} \geq 2\sqrt{x_1 x_2}$, where the first inequality follows from $(\sqrt{y_1} - \sqrt{y_2})^2 \geq 0$. Thus, the price of MNW of this instance is at most

$$\frac{x_1 + x_2}{2\sqrt{x_1 x_2}} = 1 + \frac{1}{2} \cdot \left(\sqrt[4]{\frac{x_1}{x_2}} - \sqrt[4]{\frac{x_2}{x_1}}\right)^2 \leq 1 + \frac{1}{2} \cdot \left(\sqrt[4]{4} - \sqrt[4]{\frac{1}{4}}\right)^2 = 5/4,$$

where the inequality follows from $1 \leq x_1/x_2 \leq 4$. □

Finally, we derive the exact bound for MEW and leximin with two agents. Note that since all leximin allocations are MEW, Theorem 6.18 immediately implies Theorem 6.19.

Theorem 6.18. *For $n = 2$, the price of MEW and the strong price of MEW are $3/2$.*

Proof. It suffices to show that the price of MEW is at least $3/2$ and the strong price of MEW is at most $3/2$.

Lower bound: Let $m = 3$ and $0 < \epsilon < 1/2$, and assume that the utilities are as follows:

- $u_1(1) = 1/2$, $u_1(2) = 1/2 - \epsilon$, $u_1(3) = \epsilon$;
- $u_2(1) = 1/2$, $u_2(2) = \epsilon$, $u_2(3) = 1/2 - \epsilon$.

The optimal social welfare is $3/2 - 2\epsilon$, obtained by assigning the first two goods to the first agent and the last good to the second agent. On the other hand, the maximum egalitarian welfare is $1/2$, which can be obtained only by assigning the first good to one agent and the remaining two goods to the other agent. This allocation has social welfare 1. Taking $\epsilon \to 0$, we find that the price of MEW is at least $3/2$.

Upper bound: Consider an arbitrary instance, and denote by x the maximum egalitarian welfare. The optimal social welfare is at most $1 + x$, and the social welfare of any MEW allocation is at least $2x$. Consider any MEW allocation, and suppose that agent 1 receives utility x and agent 2 receives utility $y \geq x$. In the allocation where the bundles of the two agents are swapped, the utilities are $1 - x$ and $1 - y \leq 1 - x$. Since x is the maximum egalitarian welfare, we have $x \geq 1 - y$, or $x + y \geq 1$. This means that the social welfare of the original allocation is at least 1, so the social welfare of any MEW allocation is at least $\max\{2x, 1\}$.

The strong price of MEW is therefore at most $\frac{1+x}{\max\{2x,1\}}$. If $x \leq 1/2$, this quantity is at most $\frac{1+x}{1} \leq \frac{3}{2}$. On the other hand, if $x > 1/2$, this quantity is at most $\frac{1+x}{2x} = \frac{1}{2x} + \frac{1}{2} < \frac{3}{2}$. The conclusion follows. □

Theorem 6.19. *For $n = 2$, the price of leximin and the strong price of leximin are $3/2$.*

6.7 Pareto Optimality

In this section, we consider Pareto optimality. Since any allocation that maximizes social welfare is necessarily Pareto optimal, the price of Pareto optimality is trivially 1. By establishing a tight lower bound on the welfare of a Pareto optimal allocation, we show that the strong price of Pareto optimality is quadratic. Our result indicates that while Pareto optimality is sometimes referred to as 'efficiency', it does not necessarily fare well if efficiency is measured in terms of social welfare.

Lemma 6.20. *For any instance, every Pareto optimal allocation has social welfare at least $1/n$, and this bound is tight.*

Proof. To establish the bound, it suffices to show that in any Pareto optimal allocation, some agent receives utility at least $1/n$. Suppose that this is not the case. Since the utility of each agent for the entire set of goods is 1, every agent envies at least one other agent. This implies that the *envy graph*, which has the n agents as its vertices and in which there is a directed edge from one agent to another if the former agent envies the latter, contains a directed cycle. By giving agent j's bundle to agent i for every edge $i \to j$ in the cycle, we obtain a Pareto improvement, a contradiction.

The tightness of the bound follows from the instance in Theorem 6.21. □

Theorem 6.21. *The strong price of Pareto optimality is $\Theta(n^2)$.*

Proof. **Upper bound:** Consider an arbitrary instance. Since every agent receives utility at most 1, the optimal social welfare is at most n. On the other hand, by Lemma 6.20, every Pareto optimal allocation has social welfare at least $1/n$. The conclusion follows.

Lower bound: Assume that $n \geq 2$. Let $m = n$, $0 < \epsilon < 1/n$, and assume that the utilities are as follows:

- $u_1(1) = \frac{1}{n} + \epsilon$ and $u_1(j) = \frac{1}{n} - \frac{\epsilon}{n-1}$ otherwise;
- for $i = 2, \ldots, n$: $u_i(i-1) = 1 - \epsilon$, $u_i(i) = \epsilon$, and $u_i(j) = 0$ otherwise.

Consider the allocation that assigns good $i - 1$ to agent i for $i = 2, \ldots, n$, and good n to agent 1. The welfare of this allocation is $(n-1)(1-\epsilon) + \left(\frac{1}{n} - \frac{\epsilon}{n-1}\right) = n - 1 + \frac{1}{n} - \left(n - 1 + \frac{1}{n-1}\right)\epsilon$. On the other hand, the allocation that assigns good i to agent i for $i = 1, 2, \ldots, n$ is Pareto optimal. This is because in any Pareto improvement, agent 1 must receive good 1, and it follows that agent i must receive good i for every i. The social welfare of this allocation is $\left(\frac{1}{n} + \epsilon\right) + (n-1)\epsilon = \frac{1}{n} + n\epsilon$. Taking $\epsilon \to 0$ yields the desired result. □

We also show an exact bound for the case of two agents.

Theorem 6.22. *For $n = 2$, the strong price of Pareto optimality is 3.*

Proof. The instance in Theorem 6.21 shows that the strong price of Pareto optimality is at least 3. To show that this is tight, consider an arbitrary instance and an optimal allocation in this instance. Assume that the two agents receive utility x and y in this allocation, where $x \geq y$. In any Pareto optimal allocation, at least one agent must receive utility at least y; otherwise the optimal allocation is a Pareto improvement. In combination with Lemma 6.20, this implies that the social welfare of every Pareto optimal allocation is at least $\max\{y, 1/2\}$.

The strong price of Pareto optimality is therefore at most $\frac{x+y}{\max\{y,1/2\}} \leq \frac{1+y}{\max\{y,1/2\}}$. If $y \leq 1/2$, this quantity is at most $2(1+y) \leq 3$. On the other hand, if $y > 1/2$, this quantity is at most $\frac{1+y}{y} = \frac{1}{y} + 1 < 3$. The conclusion follows. □

6.8 Conclusion and Future Work

In this chapter, we have studied the price of fairness for *indivisible* goods using several fairness notions that can always be satisfied. For most cases, we exhibited tight or asymptotically tight bounds on the worst-case efficiency loss that can occur due to fairness constraints. Interestingly, both the round-robin and MNW allocations, which are EF1, can have social welfare a linear factor away from the optimum—however, round-robin performs significantly better than this worst-case bound as long as the number of goods is not huge compared to the number of agents. The linear bound that we obtained for MNW stands in contrast to Bertsimas et al. [40]'s result in the divisible goods setting, where the price of MNW is $\Theta(\sqrt{n})$.

A potential direction for future work is to perform similar analyses but using egalitarian welfare instead of utilitarian welfare as the benchmark. This has been done, e.g., in the context of contiguous allocations [11, 150].

One could also study the price of fairness for the *chore division* problem, where chores refer to items that yield negative utility for the agents. Indeed, almost all of the notions that we consider in the goods setting have direct analogues in the chore setting, and it would be interesting to see whether the corresponding bounds in the two settings turn out to be similar as well. Recently, Sun et al. [151] studied this problem; besides EF1 and EFX, they also considered maximin share (MMS) guarantee (see Definition 2.5), pairwise MMS guarantee (see, e.g., [69, Definition 4.3]) and their relaxations. Among other results, Sun et al. provided tight bounds on the price of these notions in the case of two agents and showed the price of EF1 and EFX is infinite in the case of any number of agents. Their results in the chore setting are in sharp contrast to ours in the goods setting.

Chapter 7

Price of Connectivity [†]

7.1 Introduction

Our focus in this chapter is still on the setting where we allocate *indivisible* goods. This pertains to the allocation of houses, cars, artworks, electronics, and many other common items. Perhaps the most well-known fair division protocol is the *cut-and-choose protocol*, which dates back to at least the Bible and can be used to allocate a *divisible* good between two agents. In this protocol, the first agent divides the good into two equal parts, which is possible because the good is divisible, and the second agent chooses the part that she prefers. The cut-and-choose protocol has a direct analogue in the indivisible goods setting: since an equal partition may no longer exist, the first agent now divides the goods into two parts that are as equal as possible in her view. The resulting allocation is guaranteed to satisfy both maximin share guarantee and EF1. In fact, it also satisfies EFX, which is stronger than EF1. However, these guarantees rely crucially on the assumption that any allocation of the goods to the two agents can be chosen—in reality, there are often constraints on the allocations that we desire. One common type of constraints is captured by a model of Bouveret et al. [51], where the goods are vertices of a connected undirected graph and each agent must be allocated a connected subgraph. For instance, the goods could represent offices in a university building that we wish to divide between research groups, and it is desirable for each group to receive a connected set of offices in order to facilitate communication within the group. To what extent do the fairness guarantees continue to hold when connectivity constraints are imposed, and how does the answer depend on the underlying graph? Put another way, what is the price in terms of fairness that we have to pay if we desire connectivity?

In this chapter, we make several contributions to the active line of work on fairly allocating indivisible goods under connectivity constraints. We survey this line of work in Section 7.1.1. While we also provide fairness guarantees for any number of agents, the majority of our results concern the setting of two agents. We emphasize here that this setting is fundamental in fair division. Indeed, a number of fair division applications including divorce settlements, inheritance division, and international border disputes often fall into this

[†]This chapter has been published in a paper by Bei, Igarashi, Lu, and Suksompong [35].

Class of graphs	$n = 2$	$n \geq 2$
Paths	1 if $m = 2$ 2 if $m \geq 3$	1 if $m < n$ $m - n + 1$ if $n \leq m < 2n - 1$ n if $m \geq 2n - 1$
Stars	$m - 1$	$m - n + 1$
Vertex connectivity 1	k (see caption)	$\leq m - n + 1$
Vertex connectivity 2	$4/3$	$\leq m - n + 1$
Vertex connectivity ≥ 3	$\leq 4/3$	$\leq m - n + 1$

Table 7.1: Summary of our PoC bounds, where n and m denote the number of agents and goods, respectively. For $n = 2$ and graphs with vertex connectivity 1 (which include all trees), the parameter k denotes the maximum number of components that can result from deleting a single vertex from the graph.

setting, and numerous prominent works in the field deal exclusively with the two-agent case, e.g., [53, 57, 58, 104]. See also [133] for further discussion on the importance of the two-agent setting. In addition, as we will see, under connectivity constraints the setting with two agents is already surprisingly rich and gives rise to several mathematically deep and challenging questions.

We begin by studying maximin share guarantee for agents with additive utilities. We define the *price of connectivity (PoC)* of a graph to be the largest multiplicative gap between the maximin share defined over all possible partitions and the *graph maximin share (G-MMS)*, which is defined over all partitions that respect the connectivity constraints of the graph. For any graph and any number of agents, it follows from the definitions that if the PoC is α and one can give each agent β times her G-MMS, then it is also possible to guarantee all agents a β/α fraction of their MMS. Moreover, in cases where giving every agent their full G-MMS is possible (i.e., $\beta = 1$), we observe in Section 7.2 that the resulting factor $1/\alpha$ is tight—in other words, the PoC is the reciprocal of the optimal MMS approximation that can be achieved. Since it is known from prior work that $\beta = 1$ for two agents and arbitrary graphs as well as for any number of agents and trees, our PoC notion precisely captures the best possible MMS guarantee in these cases. Hence, determining the PoC, whose definition only involves a single utility function, allows us to identify the optimal MMS guarantee for agents with possibly different utility functions.

With this relationship in hand, we proceed to determine the PoC of various graphs; our results are summarized in Table 7.1. In the two-agent case (Section 7.3.1), we show that the PoC is related to the *vertex connectivity* of the graph, i.e., the minimum number of vertices whose deletion disconnects the graph. For graphs with connectivity exactly 1, including all trees and being significantly richer than the class of trees, we show that the PoC is equal to the maximum number of connected components that result from deleting one vertex. As a consequence, the PoC is at least 2 for any graph in this class. On the other hand, we show an upper bound of $4/3$ for all graphs with connectivity at least 2—this bound is tight for all graphs with connectivity exactly 2 and, perhaps surprisingly, for certain graphs with connectivity up to 5. In addition, we pose an intriguing conjecture that the PoC of any graph with connectivity at least 2 is closely related to its "linkedness"—the two-agent case would be completely solved if the conjecture holds—and verify our conjecture when the graph is a complete graph with an arbitrary matching removed. For any number of agents (Section 7.3.2), we establish a general upper bound of $m - n + 1$ on the PoC (where m and n denote the number of goods and agents, respectively), and show that this implies the existence of a connected allocation that gives every agent at least a $1/(m - n + 1)$ fraction of her maximin share with respect to any graph. We also derive the exact PoC for paths and stars. Notably, in order to establish the PoC for paths, we introduce a new relaxation of proportionality that we call the *indivisible proportional share (IPS) property*. This notion strengthens a number of relaxations of proportionality in the literature while maintaining guaranteed existence, so we believe that it may be of independent interest as well.

Next, in Section 7.4 we turn our attention to envy-freeness relaxations and allow agents to have arbitrary monotonic utilities. In the case of two agents, Bilò et al. [45] characterized the graphs for which an EF1 allocation always exists as the graphs that admit a "bipolar ordering" (defined in Section 7.2). While the characterization yields a strong fairness guarantee for this class of graphs, it does not give any guarantee for the remaining graphs. We generalize this result by establishing the optimal relaxation of envy-freeness for every graph—specifically, for each graph, we determine the smallest k for which an allocation that is EFk always exists with two agents. Intuitively, the less connected the graph is, the weaker the fairness guarantee we can make, i.e., the higher the price we have to pay. As a corollary, an EF$(m - 2)$ allocation exists for any connected graph, and the bound $m - 2$ is tight for stars. By contrast, we show that EFX can only be guaranteed for complete graphs with two agents. We then address the case of three agents, where we characterize the set of trees and complete bipartite graphs that admit an EF1 allocation for arbitrary utilities.

From a technical point of view, our work makes extensive use of tools and concepts from graph theory, including vertex connectivity, linkedness, ear decomposition, bipolar ordering, and block decomposition. While bipolar ordering and block decomposition have been used by Bilò et al. [45] in the EF1 characterization, the other concepts have not previously appeared in the fair division literature to the best of our knowledge. We believe that establishing these connections enriches the growing literature and lays the groundwork for fruitful collaborations between researchers across the two well-established fields.

Finally, we remark that with the exception of Theorem 7.15, all of our guarantees are constructive. In particular, we exhibit polynomial-time algorithms that produce allocations satisfying the guarantees.

7.1.1 Related Work

The papers most closely related to this chapter are the two papers that we mentioned, by Bouveret et al. [51] and Bilò et al. [45]. Bouveret et al. showed that for any number of agents with additive utilities, there always exists an allocation that gives every agent her maximin share when the graph is a tree, but not necessarily when the graph is a cycle. It is important to note that their maximin share notion corresponds to our G-MMS notion and is defined based on the graph, with only connected allocations with respect to that graph taken into account in an agent's calculation. As an example of a consequence, even though a cycle permits strictly more connected allocations than a path, it offers less guarantee in terms of the G-MMS. Our approach of considering the (complete-graph) MMS allows us to directly compare the guarantees that can be obtained for different graphs. Bilò et al. investigated the same model with respect to relaxations of envy-freeness. As we mentioned, they characterized the set of graphs for which EF1 can be guaranteed in the case of two agents with arbitrary monotonic utilities. Moreover, they showed that an EF1 allocation always exists on a path for $n \leq 4$. Intriguingly, the existence question for $n \geq 5$ remains open, although they showed that an EF2 allocation can be guaranteed for any n.

Besides [45, 51], a number of other works have recently studied fairness under connectivity constraints. Lonc and Truszczynski [114] investigated maximin share guarantee in the case of cycles, also using the G-MMS notion, while Suksompong [150] focused on paths and provided approximations of envy-freeness, proportionality, and equitability. Igarashi and Peters [102] considered fairness in conjunction with the economic efficiency notion of Pareto optimality. Bouveret et al. [52] studied the problem of *chore division*, and gave complexity results on deciding the existence of envy-free, proportional, and equitable allocations for paths and stars. Bei and Suksompong [30] proposed a similar model in which the resource is *divisible* and forms the edges of a graph (as opposed to the vertices).

Considering connected allocations can also be useful in settings where we are not interested in connectedness per se, or perhaps the goods do not even lie on any graph. A technique that has received interest recently is to arrange the goods on a path and compute a connected allocation with respect to the path. Variants of this technique have been used to devise algorithms that find a fair allocation using few queries [129] or divide goods fairly among groups of agents [110, 143].

A related line of work also combines graphs with resource allocation, but uses graphs to capture the connection between agents instead of goods. In particular, a graph specifies the acquaintance relationship among agents. Abebe et al. [1] and Bei et al. [33] defined graph-based versions of envy-freeness and proportionality with divisible resources where agents only evaluate their shares relative to other agents with whom they are acquainted. Beynier et al. [41] and Bredereck et al. [63] studied the graph-based version of envy-freeness with indivisible goods. Aziz et al. [20] introduced a number of fairness notions parameterized by the acquaintance graph.

7.2 Preliminaries

Consider an indivisible goods instance $\langle N, M, \mathcal{U} \rangle$, including the set of n agents N, the set of m indivisible goods M, and the agents' utility functions $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$, as described in Section 2.2. There is a bijection between the goods in M and the m vertices of a connected undirected graph G; we will refer to goods and vertices interchangeably. A bundle is called *connected* if the goods in it form a connected subgraph of G, and an allocation or a partition is *connected* if all of its bundles are connected. We assume in this chapter that allocations are required to be connected. As is in the most of the literature when studying maximin share guarantee [51, 96, 108, 114], we assume that utilities are additive. In this chapter, an *instance* consists of the agents, the goods and their underlying graph, and the agents' utilities for the goods.

We are ready to define maximin share guarantee considered in this chapter.

Definition 7.1 (G-MMS)**.** Given a graph G, an additive utility function u, and the number of agents n, the *graph maximin share (G-MMS)* for G, u, n is defined as

$$\text{G-MMS}(G, u, n) \coloneqq \max_{(M_1, M_2, \ldots, M_n)} \min_{i=1,2,\ldots,n} u(M_i),$$

where the maximum is taken over all partitions $(M_1, M_2, \ldots, M_n)$ that are connected with respect to G. The maximin share (MMS) defined in Definition 2.5 for u, n is thus

$$\text{MMS}(u, n) = \text{G-MMS}(K_m, u, n),$$

where K_m denotes the complete graph over the goods. When the parameters are clear from the context, we will refer to the graph maximin share and the maximin share simply as G-MMS and MMS, respectively. A partition for which the maximum is attained is called a *G-MMS partition* (resp., *MMS partition*).

It follows from the definition that

$$\text{G-MMS}(G, u, n) \leq \text{MMS}(u, n) \leq \frac{u(M)}{n}$$

for all G, u, n, and $\text{G-MMS}(G_1, u, n) \leq \text{G-MMS}(G_2, u, n)$ if G_1 is a subgraph of G_2. Moreover, $\text{G-MMS}(G, u, n) = \text{MMS}(u, n) = 0$ if $m < n$.

Next, we define the price of connectivity.

Definition 7.2. Given a graph G and the number of agents n, the *price of connectivity (PoC)* of G for n agents is defined as

$$\sup_u \frac{\text{MMS}(u, n)}{\text{G-MMS}(G, u, n)},$$

where the supremum is taken over all possible additive utility functions u.[1] We denote the PoC of a graph G for n agents by $\text{PoC}(G, n)$.

By definition of the PoC, we have

$$\text{PoC}(G, n) \cdot \text{G-MMS}(G, u, n) \geq \text{MMS}(u, n) \tag{7.1}$$

[1]We interpret $\frac{0}{0}$ in this context to be equal to 1. Note that $\text{MMS}(u, n) = 0$ if and only if $\text{G-MMS}(G, u, n) = 0$, because both conditions are equivalent to the condition that fewer than n goods yield a positive utility according to u.

for any G, u, n, and the factor PoC(G, n) cannot be replaced by any smaller factor. When G and n are clear from the context, we will refer to PoC(G, n) simply as PoC. Note that the PoC is always at least 1, and is exactly 1 for complete graphs of any size. Moreover, the PoC is 1 if $m \leq n$.

Suppose that for some graph G and number of agents n, there always exists a connected allocation that gives each agent at least β times her G-MMS. By Equation (7.1), this allocation also gives each agent at least $\beta/$PoC(G, n) times her MMS. Prior work has established that $\beta = 1$ when $n = 2$ and G is arbitrary [114, Corollary 2], as well as when G is a tree and n is arbitrary [51, Theorem 5.4]. Hence, in these cases, we can guarantee each agent at least $1/$PoC(G, n) times her MMS. The factor $1/$PoC(G, n) is also the best possible. To see this, consider n agents with the same utility function u. From the definition of G-MMS, any connected allocation gives some agent a value of at most G-MMS(G, u, n). By considering u such that G-MMS(G, u, n) is arbitrarily close to MMS$(u, n)/$PoC(G, n), this agent receives arbitrarily close to $1/$PoC(G, n) times her MMS. To summarize, we have the following proposition.

Proposition 7.3. *Let n be any positive integer and G be any graph. If $n = 2$ (and G is arbitrary), or if G is a tree (and n is arbitrary), then there always exists a connected allocation that gives each agent at least $1/$PoC(G, n) times her MMS. Moreover, the factor $1/$PoC(G, n) is tight in both cases.*

Proposition 7.3 implies that if there are two agents or G is a tree, in order to determine the optimal MMS approximation for agents with possibly different utilities, it suffices to determine the value PoC(G, n), which only concerns a single utility function.

We also consider relaxations of envy-freeness: EFk and EFX (see Definition 2.3).

All graphs considered in this chapter are assumed to be connected. The *vertex connectivity* (or simply *connectivity*) of a graph G is the minimum number of vertices whose deletion disconnects G. A graph with vertex connectivity at least k is said to be *k-connected*. By definition, every connected graph is 1-connected. A 2-connected graph is also called *biconnected*. A *bipolar ordering* (also called *bipolar numbering*) of a graph is an ordering of its vertices such that every prefix and every suffix of the ordering forms a connected graph.

7.3 Maximin Share Guarantee

In this section, we consider maximin share guarantee. Our goal is to derive bounds on the PoC for arbitrary graphs in the case of two agents, and for paths and stars in the general case. By Proposition 7.3, this also yields the optimal MMS approximation for each of these cases.

7.3.1 Two Agents

We first focus on the case of two agents and start by establishing the PoC for all graphs with connectivity 1.

Theorem 7.4. *Let G be a graph with connectivity exactly 1, and let $k \geq 2$ be the maximum number of connected components that can result from deleting a single vertex of G. Then $PoC(G, 2) = k$.*

Proof. First, we show that the PoC of G is at least k. Let v be a vertex of G whose deletion results in k components. Consider a utility function with value k for v, value 1 for an arbitrary vertex in each of the k components, and value 0 for all other vertices. The MMS is k. In any connected bipartition, the part that does not contain v is a subset of one of the k components, so this part has value at most 1. Hence the PoC is at least k.

Next, we show that the PoC of G is at most k. Take an arbitrary utility function u, and assume without loss of generality that $u(M) = 1$. Since $\text{MMS}(u, 2) \leq u(M)/2 = 1/2$, the desired claim follows if there is a connected bipartition such that both parts have value at least $1/(2k)$. Assume that no such bipartition exists.

Pick a spanning tree T of G, and let v be an arbitrary vertex. The removal of v results in a number of subtrees of T; clearly, at most one of these subtrees can have value more than $1/2$. If such a subtree exists, we move from v towards the adjacent vertex in that subtree and repeat the procedure with the new centre vertex. Note that we will never traverse back an edge—otherwise there are two disjoint subtrees with value more than $1/2$ each, contradicting $u(M) = 1$. Since the tree is finite, we eventually reach a vertex v such that all subtrees $T_1, T_2, \ldots, T_r$ resulting from the removal of v have value at most $1/2$ each.

Since T_i and $T \setminus T_i$ are both connected for every i, by our earlier assumption, each of the subtrees $T_1, T_2, \ldots, T_r$ has value less than $1/(2k)$. Recall that in the original graph G, removing v can result in at most k components. This means that if $r > k$, the r subtrees must be connected by some edges not belonging to T. If subtrees T_i and T_j are connected by such an edge, we can merge T_i and T_j into one component. Note that $T_i \cup T_j$ has value less than $1/(2k) + 1/(2k) = 1/k \leq 1/2$, so since $T_i \cup T_j$ and $T \setminus (T_i \cup T_j)$ are both connected, $T_i \cup T_j$ must again have value less than $1/(2k)$. Our procedure can be repeated until the components can no longer be merged, at which point we are left with at most k components. Each of these components has value less than $1/(2k)$, which implies that v has value more than $1 - k/(2k) = 1/2$. In this case, a bipartition with v as one part is an MMS partition, so $\text{MMS}(u, 2) = 1 - u(v)$. On the other hand, at least one of the (at most) k components has value at least $(1 - u(v))/k$, which is $1/k$ of the MMS. We can take a connected bipartition with such a component as one part and obtain the desired result. □

We remark that the proof of Theorem 7.4 also yields a polynomial-time algorithm for computing a bipartition such that both parts have value at least $1/k$ of the MMS. To compute an allocation between two agents such that both agents receive $1/k$ of their MMS, we simply let the first agent compute a desirable bipartition, and let the second agent choose the part that she prefers. Since $\text{MMS}(u, 2) \leq u(M)/2$, the second agent is always satisfied.

Before we move on to results about graphs with higher connectivity, we show the following lemma, which will help simplify our subsequent proofs. The lemma implies that in order to prove an upper bound on the PoC in the case of two agents, it suffices to establish the bound for utility functions such that in an MMS partition, the two parts are of equal value.

Lemma 7.5. *For $n = 2$ and any graph G, the PoC remains the same if instead of taking the supremum in Definition 7.2*

$$\sup_u \frac{MMS(u,2)}{G\text{-}MMS(G,u,2)}$$

over all utility functions u, we only take the supremum over all utility functions u such that in any MMS partition according to u, the two parts are of equal value.

Proof. Let u be an arbitrary utility function, and suppose that in an MMS partition, the two parts are of value $x \leq y$. We have $\text{MMS}(u,2) = x$. Let $\alpha := \frac{\text{MMS}(u,2)}{\text{G-MMS}(G,u,2)}$. In any connected bipartition, each part either has value at most x/α, or at least $(x+y)-x/\alpha = y+(1-1/\alpha)x$.

Consider a modified utility function u' where in the MMS partition above, we arbitrarily decrease the values of some goods in the part with value y so that the part has value x. It is clear that $\text{MMS}(u',2) = x$. With respect to u', in any connected bipartition, each part either has value at most x/α, or at least $y + (1-1/\alpha)x - (y-x) = (2-1/\alpha)x$. This means that $\text{G-MMS}(G,u',2) \leq x/\alpha = \text{MMS}(u',2)/\alpha$, or $\frac{\text{MMS}(u',2)}{\text{G-MMS}(G,u',2)} \geq \alpha$. Since the two parts in any MMS partition according to u' are of equal value, the proof is complete. □

Next, we consider biconnected graphs, i.e., graphs with connectivity at least 2. We show that the PoC is at most $4/3$ for all such graphs—this is in contrast to graphs with connectivity 1, which have PoC at least 2 according to Theorem 7.4. For this result, we will use a property of biconnected graphs which we state in the following proposition. An *open ear decomposition* of a graph consists of a cycle as the first *ear* and a sequence of paths as subsequent ears such that in each path, the first and last vertices (which must be different) belong to previous ears while the remaining vertices do not.

Proposition 7.6 (Whitney [157, 158])**.** *In a biconnected graph with at least three vertices, any two vertices belong to a common cycle, and there exists an open ear decomposition. Moreover, we may choose any cycle in the graph as the first ear.*[2]

Theorem 7.7. *Let G be a biconnected graph. Then PoC$(G,2) \leq 4/3$.*

Proof. The case $m \leq 2$ is trivial since $n = 2$ and the PoC is 1 in this case, so consider $m \geq 3$. Take an arbitrary utility function u, and assume without loss of generality that $u(M) = 1$. By Lemma 7.5, we may also assume that $\text{MMS}(u,2) = 1/2$. Call a good *heavy* if it has value strictly more than $1/4$. Since there can be at most one heavy good in each part of an MMS partition, there are at most two heavy goods in total. Pick goods g_1 and g_2 so that together they include all of the heavy goods. By Proposition 7.6, there is a cycle in G containing g_1 and g_2, and an open ear decomposition with this cycle as the first ear.

[2] There is also a linear-time algorithm for computing an open ear decomposition with an arbitrary cycle as the first ear [142].

We will construct a bipolar ordering of the vertices that begins with g_1 and ends with g_2. Assume that the first ear is a cycle with vertex order

$$g_1, h_1, \ldots, h_i, g_2, h_{i+1}, \ldots, h_j.$$

We arrange these vertices as

$$g_1, h_1, h_2, \ldots, h_i, h_j, h_{j-1}, \ldots, h_{i+1}, g_2.$$

For each subsequent ear, suppose that the two vertices belonging to previous ears are h and h', where h appears before h' in the current ordering. We insert the remaining vertices on the path from h to h' into the ordering directly after h, following the same order as in the path. One can check (for example, by induction on the number of ears) that the resulting ordering is a bipolar ordering beginning with g_1 and ending with g_2.

Consider first the case where $\max\{u(g_1), u(g_2)\} > 1/2$; assume without loss of generality that $u(g_1) > 1/2$. In this case, MMS$(u, 2) = 1 - u(g_1) < 1/2$, contradicting the assumption that MMS$(u, 2) = 1/2$.

Assume now that $\max\{u(g_1), u(g_2)\} \leq 1/2$, and recall that $u(g) \leq 1/4$ for all $g \notin \{g_1, g_2\}$. Since MMS$(u, 2) = 1/2$, it suffices to find a connected bipartition such that both parts have value at least $3/8$. Let $S = \{g_1\}$, so $u(S) \leq 1/2$. We add one good at a time to S following the bipolar ordering until $u(S) \geq 1/2$. Since $u(g_2) \leq 1/2$, we stop (not necessarily directly) before we add g_2. Moreover, since each good besides g_1 and g_2 has value at most $1/4$, at some point during this process we must have $3/8 \leq u(S) \leq 5/8$. In the bipartition with S as one part, both parts are connected and have value at least $3/8$, completing the proof. □

Unlike for Theorem 7.4, the proof of Theorem 7.7 does not directly lead to a polynomial-time algorithm for computing an allocation such that both agents receive at least $3/4$ of their MMS. The problematic step is when we apply Lemma 7.5, since computing the maximin share is NP-hard by a straightforward reduction from the partition problem. Woeginger [159] showed that a polynomial-time approximation scheme (PTAS) for the problem exists—using this PTAS, we can obtain a $(3/4 - \epsilon)$-approximation algorithm that runs in polynomial time for any constant $\epsilon > 0$. Nevertheless, we next show that by building upon the proof of Theorem 7.7, we can also achieve a polynomial-time $3/4$-approximation algorithm.

Proposition 7.8. *For biconnected graph and $n = 2$, there exists a polynomial-time algorithm for computing an allocation that gives both agents at least $3/4$ of their MMS.*

Proof. As in the remark following Theorem 7.4, it suffices to compute a bipartition such that the first agent has value at least $3/4$ of her MMS for both parts; the second agent can then choose the part that she prefers. To compute such a bipartition, we iterate over all pairs of goods g_1, g_2. For each pair, we construct a bipolar ordering that begins with g_1 and ends with g_2; this is possible as explained in the proof of Theorem 7.7. We then consider taking every possible prefix of the ordering as one part of the bipartition, and return the bipartition

Algorithm 8: Approximate MMS Algorithm for Biconnected Graphs

Input: Indivisible goods M, undirected graph G, and utility function u.

current-best $\leftarrow 0$
$M_1 \leftarrow \emptyset$
$M_2 \leftarrow M$
for $(g_1, g_2) \in M \times M$ *with* $g_1 \neq g_2$ **do**
    Construct a bipolar ordering σ of G starting with g_1 and ending with g_2.
    for $g \in M$ **do**
        $M_1' \leftarrow \{$all goods before g in $\sigma\}$
        $M_2' \leftarrow \{g$ and all goods after g in $\sigma\}$
        if $\min\{u(M_1'), u(M_2')\} >$ *current-best* **then**
            current-best $\leftarrow \min\{u(M_1'), u(M_2')\}$
            $M_1 \leftarrow M_1'$
            $M_2 \leftarrow M_2'$
return (M_1, M_2)

with the highest minimum between the two parts across all pairs g_1, g_2. The pseudocode of the algorithm is given as Algorithm 8.

Since constructing a bipolar ordering with a specific cycle as the first ear can be done in linear time [142], Algorithm 8 runs in polynomial time. We now establish the correctness of the algorithm. Assume without loss of generality that $\text{MMS}(u, 2) = 1/2$, so there exists a bipartition (M_1, M_2) of M such that $u(M_1) = 1/2 \leq u(M_2)$.[3] Consider a modified utility function u' where we start with u and arbitrarily decrease the values of some goods in M_2 so that $u'(M_2) = 1/2$. In the new instance, the proof of Theorem 7.7 implies that there exists a connected bipartition for which both parts have value at least $3/8$, and this bipartition corresponds to one of the bipartitions examined by Algorithm 8. Since the values in the original instance with utility function u can only be higher than in the new instance with utility function u', in the original instance both parts of this bipartition also have value at least $3/8$. It follows that both parts of the bipartition returned by Algorithm 8 have value at least $3/8$, which is $3/4$ of the MMS. □

In the light of Theorems 7.4 and 7.7, it is tempting to believe that for graphs with connectivity 3 or higher, the PoC is strictly less than $4/3$. Perhaps surprisingly, this is not the case: a counterexample is the wheel graph shown in Figure 7.1, which has connectivity 3. In the instance shown in the figure, the MMS is 4 while the G-MMS is 3, so the PoC of the graph is at least $4/3$ (and by Theorem 7.7, exactly $4/3$). The key point of this example is that the graph cannot be partitioned into two connected subgraphs in such a way that one subgraph contains the vertices with value 1 and 3, while the other subgraph contains the two vertices with value 2. This observation allows us to generalize the counterexample. A graph is said to be 2-*linked* if for any two disjoint pairs of vertices (a, b) and (c, d), there exist two vertex-disjoint paths, one from a to b and the other from c to d.

[3]Note that we are not assuming $u(M) = 1$ as in the proof of Theorem 7.7.

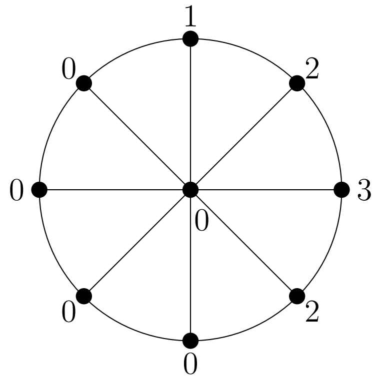


Figure 7.1: An instance showing that the PoC of a wheel graph is at least $4/3$.

Proposition 7.9. *Let G be a graph that is not 2-linked. Then PoC$(G, 2) \geq 4/3$.*

Proof. Suppose that G is not 2-linked, and let (a, b) and (c, d) be disjoint pairs of vertices such that there do not exist two disjoint paths, one from a to b and the other from c to d. Consider a utility function u such that $u(a) = u(b) = 2$, $u(c) = 3$, $u(d) = 1$, and $u(g) = 0$ for every other vertex g. We have MMS$(u, 2) = 4$. On the other hand, the graph cannot be partitioned into two connected subgraphs in such a way that one subgraph contains a and b while the other subgraph contains c and d—indeed, such a partition would give rise to two disjoint paths that cannot exist by our assumption. This means that G-MMS$(G, u, 2) \leq 3$. Hence PoC$(G, 2) \geq 4/3$. □

Every graph with connectivity at most 2 is not 2-linked.[4] And Figure 7.1 shows an example of a 3-connected graph that also does not satisfy the property. In fact, Mészáros [124, Figure 1] constructed a 5-connected graph that still fails to be 2-linked![5] Combining these facts with Theorem 7.7 yields the following corollaries.

Corollary 7.10. *For every graph G with connectivity 2, PoC$(G, 2) = 4/3$.*

Corollary 7.11. *For some graph G with connectivity 5, PoC$(G, 2) = 4/3$.*

While we have not been able to precisely determine the PoC for all graphs with connectivity 3 or above, we present a conjecture that, if settled in the affirmative, would complete the picture for the two-agent case. Before we can describe the conjecture, we need the following generalization of 2-linkedness [124].

Definition 7.12. Given positive integers a, b, a graph G is said to be (a, b)-*linked* if for any disjoint set of vertices M_1, M_2 with $|M_1| = a$ and $|M_2| = b$, there exist disjoint connected subgraphs G_1, G_2 of G such that M_i is contained in G_i for $i = 1, 2$.

[4] Indeed, given such a graph, let a, b be two vertices whose removal disconnects the graph, and let c, d be vertices from distinct components in the resulting graph. Then any path between c and d must go through either a or b.

[5] On the other hand, a 6-connected graph is always 2-linked [103].

For example, $(2,1)$-linkedness is equivalent to biconnectivity,[6] while $(2,2)$-linked graphs correspond to what we have so far called 2-linked graphs. The new definition allows us to extend the lower bound from Proposition 7.9.

Proposition 7.13. *Let k be a positive integer, and let G be a graph that is not $(2,k)$-linked. Then PoC$(G,2) \geq 2k/(2k-1)$.*

Proof. Suppose that G is not $(2,k)$-linked, and let $\{a_1,a_2\}$ and $\{b_1,b_2,\dots,b_k\}$ be sets of vertices for which there do not exist disjoint connected subgraphs separating them. Consider a utility function u such that $u(a_1) = u(a_2) = k$, $u(b_1) = k+1$, $u(b_2) = u(b_3) = \cdots = u(b_k) = 1$, and $u(g) = 0$ for every other vertex g. We have $\text{MMS}(u,2) = 2k$. On the other hand, the graph cannot be partitioned into two connected subgraphs in such a way that one subgraph contains a_1, a_2 while the other subgraph contains $b_1, b_2, \dots, b_k$. Since all vertex values are integers, this implies that $\text{G-MMS}(G,u,2) \leq 2k-1$. Hence $\text{PoC}(G,2) \geq 2k/(2k-1)$. □

Our conjecture is that for biconnected graphs, the PoC is exactly captured by $(2,k)$-linkedness.

Conjecture 7.14. *Let $k \geq 2$ be an integer, and let G be a graph that is $(2,k-1)$-linked but not $(2,k)$-linked. Then PoC$(G,2) = 2k/(2k-1)$.*

The case $k=2$ of Conjecture 7.14 holds by Corollary 7.10. We demonstrate next that the conjecture also holds for 'almost-complete' graphs, i.e., for complete graphs with a non-empty matching removed. These graphs have minimum degree $m-2$, where m is the number of vertices (i.e., goods). We show that the PoC of these graphs is always exactly $(2m-4)/(2m-5)$, with the only exception being the graph L_5 that results from removing two disjoint edges from the complete graph K_5 (Figure 7.2). The graph L_5 is not 2-linked, so Proposition 7.9 (or alternatively, the utilities in Figure 7.2) implies that its PoC is at least $4/3$ instead of $6/5$. In fact, since the graph has connectivity 3, Theorem 7.7 tells us that its PoC is exactly $4/3$.

Theorem 7.15. *Let G be a graph that results from removing a non-empty matching from a complete graph with at least three vertices, and assume that G is different from L_5. Then PoC$(G,2) = (2m-4)/(2m-5)$.*

To prove Theorem 7.15, we will use the following lemma.

[6]To see this, first consider a graph G that is not biconnected—suppose that removing a vertex x disconnects G. If y and z are vertices in different components of the resulting disconnected graph, then taking $M_1 = \{y,z\}$ and $M_2 = \{x\}$ yields a violation of Definition 7.12, meaning that G is not $(2,1)$-linked. Conversely, suppose that G is biconnected, and consider any disjoint set of vertices $M_1 = \{y,z\}$ and $M_2 = \{x\}$. By definition of biconnectivity, the graph G remains connected upon the removal of x. Hence, we may take G_2 to be the subgraph induced only on x and G_1 to be the subgraph induced on all vertices except x in Definition 7.12. This implies that G is $(2,1)$-linked.

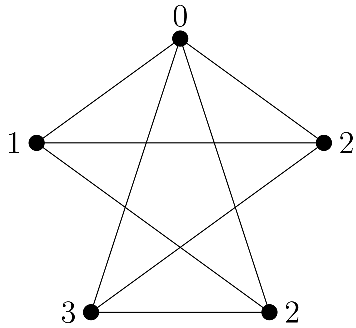


Figure 7.2: Graph L_5 and utilities showing that its PoC is at least $4/3$.

Lemma 7.16. *Let k be a positive integer, $2 \leq s \leq 2k$ be a real number, and $x_1, x_2, \ldots, x_k \geq 1$ be real numbers with sum s. For any real number $0 \leq r \leq s-2$, there exists a subset $J \subseteq \{1, 2, \ldots, k\}$ such that $r \leq \sum_{j \in J} x_j \leq r+2$.*

Proof. We proceed by induction on k. For the base case $k = 1$ we must have $s = 2$, $x_1 = 2$, $r = 0$, and the result holds trivially. Suppose now that the result holds for $k-1$; we will prove it for k. Assume without loss of generality that $x_1 = \max\{x_1, x_2, \ldots, x_k\}$.

First, assume that $x_1 \leq 2$. Define $y_i := x_1 + x_2 + \cdots + x_i$ for each i. The sequence $0, y_1, y_2, \ldots, y_k = s$ is strictly increasing and any two consecutive terms differ by at most 2, so one of the terms $x_1 + x_2 + \cdots + x_i$ must be between r and $r+2$. Hence we may take $J = \{1, 2, \ldots, i\}$ to fulfil the claim.

Assume from now on that $x_1 > 2$. We first prove the statement for $r \geq s/2 - 1$. If $x_1 > s/2 + 1$, then since $x_i \geq 1$ for all i, we have

$$s = x_1 + x_2 + \cdots + x_k > (s/2 + 1) + (k-1) = s/2 + k,$$

or $s > 2k$, a contradiction. So $x_1 \leq s/2 + 1 \leq r + 2$. If $x_1 \geq r$, we are done by choosing $J = \{1\}$, so assume that $x_1 < r$.

Let $t := x_2 + x_3 + \cdots + x_k$. Note that $0 \leq t \leq s - 2 \leq 2(k-1)$ and $0 < r - x_1 \leq s - 2 - x_1 = t - 2$. Applying the induction hypothesis on $x_2, x_3, \ldots, x_k$, we find that there is a set $L \subseteq \{2, 3, \ldots, k\}$ such that $r - x_1 \leq \sum_{l \in L} x_l \leq r - x_1 + 2$. Take $J = L \cup \{1\}$. We have $r \leq \sum_{j \in J} x_j \leq r + 2$, as desired.

Finally, suppose that $r < s/2 - 1$. We have

$$s - 2 \geq s - r - 2 > s - (s/2 - 1) - 2 = s/2 - 1,$$

so we know from the previous case ($r \geq s/2 - 1$) that there exists a subset $J \subseteq \{1, 2, \ldots, k\}$ for which $s - r - 2 \leq \sum_{j \in J} x_j \leq s - r$. Since $\sum_{j=1}^{k} x_j = s$, it follows that $r \leq \sum_{j \in \{1,2,\ldots,k\} \setminus J} x_j \leq r + 2$, completing the proof. □

We are now ready to establish Theorem 7.15.

Proof of Theorem 7.15. First, we show that the PoC of G is at least $(2m-4)/(2m-5)$. Let (v_1, v_2) be a missing edge. Consider a utility function with value $m-2$ for each of v_1 and v_2, value $m-1$ for another vertex v_3, and value 1 for each of the remaining $m-3$ vertices (so the total value is $4m-8$). The MMS is $2m-4$, attained by the bipartition with $\{v_1, v_2\}$ as one part. Take an arbitrary connected bipartition. If v_1 and v_2 are in the same part, this part must contain at least one other vertex, so the other part has value at most $2m-5$. On the other hand, if v_1 and v_2 are in different parts, the part that does not contain v_3 has value at most $2m-5$. In either case, there is a part with value no more than $2m-5$, so the G-MMS is at most $2m-5$. It follows that the PoC is at least $(2m-4)/(2m-5)$.

Next, we show that the PoC of G is at most $(2m-4)/(2m-5)$. Take an arbitrary utility function u, and assume without loss of generality that $u(M) = 4m-8$. By Lemma 7.5, we may also assume that $\text{MMS}(u, 2) = (4m-8)/2 = 2m-4$. It suffices to show that $\text{G-MMS}(G, u, 2) \geq 2m-5$. Consider any MMS partition. If the partition is connected, we have that the G-MMS is $2m-4$. Suppose therefore that the partition is not connected. Since G results from removing a non-empty matching from a complete graph, this means that (at least) one of the parts corresponds to a missing edge. Let v_1 and v_2 be the two vertices in that part (so $u(\{v_1, v_2\}) = 2m-4$), and $v_3, \ldots, v_m$ be the remaining vertices of G.

Assume first that there exists a vertex $v \notin \{v_1, v_2\}$ such that $u(v) \leq 1$. We have $2m-4 \leq u(\{v_1, v_2, v\}) \leq 2m-3$, and the vertices v_1, v_2, v form a connected subgraph. Moreover, since the graph G is different from L_5, the remaining vertices also form a connected subgraph; together these vertices have value at least $(4m-8)-(2m-3) = 2m-5$. Hence, in the connected bipartition with $\{v_1, v_2, v\}$ as one part, both parts have value at least $2m-5$. It follows that $\text{G-MMS}(G, u, 2) \geq 2m-5$ in this case.

Assume now that every vertex $v \notin \{v_1, v_2\}$ satisfies $u(v) > 1$. If $u(v_1) \geq 2m-5$, then taking the connected bipartition with v_1 alone as one part again yields $\text{G-MMS}(G, u, 2) \geq 2m-5$; an analogous argument applies if we have $u(v_2) \geq 2m-5$. Suppose therefore that $\max\{u(v_1), u(v_2)\} < 2m-5$. Since $u(v_1)+u(v_2) = 2m-4$, we have $1 < u(v_1) < 2m-5$, and so $0 < 2m-5-u(v_1) < 2m-6$. Applying Lemma 7.16 with $k = m-2$, $s = 2m-4$, $\{x_1, \ldots, x_k\} = \{u(v_3), \ldots, u(v_m)\}$, and $r = 2m-5-u(v_1)$, we find that there exists a subset of $\{u(v_3), \ldots, u(v_m)\}$ for which the sum of the elements belongs to the interval $[2m-5-u(v_1), 2m-3-u(v_1)]$. Letting S be the set of corresponding goods along with v_1, we have $2m-5 \leq u(S) \leq 2m-3$. Hence, in the connected bipartition with S as one part, both parts have value at least $2m-5$. Therefore $\text{G-MMS}(G, u, 2) \geq 2m-5$ in this case as well, and the proof is complete. □

One can check that any graph G satisfying the condition of Theorem 7.15 is $(2, m-3)$-linked but not $(2, m-2)$-linked, so Theorem 7.15 confirms Conjecture 7.14 for this class of graphs.

7.3.2 Any Number of Agents

We proceed to the general setting where the goods are divided among an arbitrary number of agents. In this setting, it is no longer true that the PoC alone captures the MMS approximation that can be guaranteed to the agents—this is evident in the case of a complete graph, where the PoC is 1 by definition, but an allocation that gives all agents their full MMS does not always exist [108]. At first glance, it may seem conceivable that certain graphs do not admit any useful MMS approximation. However, we provide a non-trivial guarantee for arbitrary graphs that depends only on the number of agents and goods and, in particular, not on the utilities (Theorem 7.18). We begin by establishing a general upper bound on the PoC.

Theorem 7.17. *For any graph G and number of agents n, we have*

$$\mathit{PoC}(G, n) \le \max\{1, m - n + 1\}.$$

Proof. If $m < n$, the PoC is 1. Assume that $m \ge n$, and consider an arbitrary utility function u. Let $(M_1, M_2, \ldots, M_n)$ be a (not necessarily connected) partition of M that maximizes $\min_{i=1,\ldots,n} u(M_i)$. We assume without loss of generality that $|M_i| \ge 1$ for each i, which also means that $|M_i| \le m - n + 1$ for every i.

For each i, let g_i be a good of highest value in M_i according to u, and let $M_i' = \{g_i\}$. As long as $\bigcup_{i=1}^n M_i' \ne M$, we add a good not already in $\bigcup_{i=1}^n M_i'$ to one of the bundles M_i' so that the bundle remains connected; this is always possible since G is connected. At the end of this process, $(M_1', M_2', \ldots, M_n')$ is a connected partition of M. By our choice of g_i, we have

$$u(M_i') \ge \frac{1}{m - n + 1} \cdot u(M_i)$$

for every i. It follows that

$$\text{G-MMS}(G, u, n) \ge \min_{i=1,\ldots,n} u(M_i') \ge \frac{1}{m - n + 1} \cdot \min_{i=1,\ldots,n} u(M_i) = \frac{1}{m - n + 1} \cdot \text{MMS}(u, n).$$

Hence $\text{PoC}(G, n) \le m - n + 1$. □

As we will see in Theorems 7.20 and 7.23, the bound $m - n + 1$ is tight for sufficiently short paths and all stars. We now give an approximate maximin share guarantee for arbitrary graphs.

Theorem 7.18. *For any graph G and any number of agents n, there exists a connected allocation that gives each agent at least $1/(m - n + 1)$ of her MMS.*

Proof. Take an arbitrary spanning tree H of G. By Theorem 7.17, $\text{PoC}(H, n) \le m - n + 1$. By Proposition 7.3, there exists a connected allocation with respect to H that gives each agent at least $1/(m - n + 1)$ times her MMS. Since any connected allocation with respect to H is also connected with respect to G, the conclusion follows. □

Next, we derive tight bounds on the PoC in the cases of paths and stars for any number of agents. By Proposition 7.3, this also yields the optimal MMS approximation for each of these cases. The following simple fact will be useful.

Lemma 7.19. *Let $m \ge n$, and let $M' \subseteq M$ be an arbitrary set of at least $m - n + 1$ goods. For an agent with utility function u, we have $u(M') \ge \mathit{MMS}(u, n)$.*

Proof. Observe that in any partition of the vertices into n parts, at least one of the parts is contained in M'. In particular, this holds for an MMS partition. It follows that $\text{MMS}(u,n) \le u(M')$, as claimed. $\square$

We begin with stars.

Theorem 7.20. *Let $n \ge 2$ and let G be a star. Then*

$$\textit{PoC}(G,n) = \begin{cases} m-n+1 & \textit{if } m \ge n; \\ 1 & \textit{if } m < n. \end{cases}$$

Proof. If $m < n$ the PoC is 1, so assume that $m \ge n$. We first show that the PoC is at least $m-n+1$. Consider a utility function u with value $m-n+1$ for the centre vertex and for $n-2$ of the leaves, and value 1 for each of the remaining $m-n+1$ leaves. We have $\text{MMS}(u,n) = m-n+1$. In any connected partition into n parts, at least $n-1$ parts contain a single leaf. This means that at least one of these parts contains a single leaf with value 1. Hence the PoC is at least $m-n+1$.

Next, we show that the PoC is at most $m-n+1$. Take an arbitrary utility function u, let v^* be the centre vertex, and let $v_1, v_2, \dots, v_{n-1}$ be the leaves with the highest value where $u(v_1) \ge \dots \ge u(v_{n-1})$. Consider a connected partition Π with each of these $n-1$ vertices as a part, and the remaining $m-n+1$ vertices as the last part.

Let $A := M \setminus \{v^*, v_1, \dots, v_{n-2}\}$. By Lemma 7.19, $\text{MMS}(u,n) \le u(A)$. Since there are $m-n+1$ vertices in A and v_{n-1} is a vertex with the highest value, we have

$$u(v_{n-1}) \ge \frac{1}{m-n+1} \cdot u(A) \ge \frac{1}{m-n+1} \cdot \text{MMS}(u,n).$$

It follows that $u(v_i) \ge \text{MMS}(u,n)/(m-n+1)$ for all $i = 1,2,\dots,n-1$, so the first $n-1$ parts of Π have value at least $\text{MMS}(u,n)/(m-n+1)$ each. The last part of Π is $B := M \setminus \{v_1, v_2, \dots, v_{n-1}\}$. By Lemma 7.19 again, we have $\text{MMS}(u,n) \le u(B)$. This means that all parts of Π have value at least $\text{MMS}(u,n)/(m-n+1)$, as desired. $\square$

Remark. We note that Theorem 7.17 already implies that the PoC is at most $m-n+1$. Nevertheless, the above proof gives rise to a polynomial-time algorithm for computing a connected allocation for n agents on a star such that each agent receives at least $\frac{1}{m-n+1}$ of her maximin share: Let each of the first $n-1$ agents pick a favourite leaf from the remaining leaves in turn, and let the last agent take the remaining $m-n+1$ vertices.

To address the more involved case of paths, we introduce an approximation of proportionality that can be of interest even in the absence of connectivity considerations. Recall that an allocation is said to be *proportional* if it gives every agent at least her *proportional share*, which is defined as $u(M)/n$. Even though a proportional allocation always exists for *divisible* goods, as we explained in the introduction, this is not the case for indivisible goods—our definition of *indivisible proportional share* therefore adapts proportionality to the setting of indivisible goods. In order to ensure a non-trivial approximation, we will need to hypothetically remove up to $n-1$ goods from the entire bundle. Indeed, when there are $n-1$ goods overall, in any allocation, one of the agents is necessarily left empty-handed. If this agent is only allowed to hypothetically remove at most $n-2$ goods, then she cannot guarantee any positive (multiplicative) approximation of her utility for the entire bundle. Thus, we are interested in the optimal approximation of each agent's utility after $n-1$ goods are removed. When the number of goods is large, this approximation is $1/n$, which is reasonable because there are n agents. However, for smaller number of goods, we will be able to achieve a better approximation, which is captured by our IPS factor in the following definition.

Definition 7.21 (IPS)**.** For positive integers n, m, define

$$\text{IPS}(n, m) = \begin{cases} \frac{1}{n} & \text{if } m \geq 2n-1; \\ \frac{1}{m-n+1} & \text{if } n \leq m < 2n-1; \\ 0 & \text{if } m < n. \end{cases}$$

Given n agents and m goods, a bundle A is said to satisfy the *indivisible proportional share (IPS) property* for an agent with utility function u if there exists a (possibly empty) set $B \subseteq M \setminus A$ with $|B| \leq n-1$ such that

$$u(A) \geq \text{IPS}(n, m) \cdot u(M \setminus B).$$

An allocation is said to satisfy the IPS property if every agent receives a bundle that satisfies the IPS property. For brevity, we will refer to a bundle or allocation that satisfies the IPS property as being IPS.

We remark that IPS is a stronger property than PROP*$(n-1)$ considered by Segal-Halevi and Suksompong [143], which corresponds to taking $\text{IPS}(n, m) = 1/n$ for $m \geq n$ and 0 for $m < n$. It is also stronger than PROP1 considered by Conitzer et al. [76] and Aziz et al. [21], as well as a proportionality relaxation studied by Suksompong [150]. Despite its strength, we show that an IPS allocation always exists. Moreover, we can obtain a connected IPS allocation if the graph is a path.

Proposition 7.22. *Let $n \geq 2$ and let G be a path. There exists a connected IPS allocation of the m goods to the n agents.*

Proof. If $m < n$, each agent needs utility 0 in an IPS allocation, so the claim holds trivially. Assume that $m \geq n$. Starting with an empty bundle, we process the goods along the path (say, from left to right) and add them one at a time to the current bundle until the bundle is IPS to at least one of the agents. We then allocate the bundle to one such agent, and repeat the procedure with the remaining goods and agents. Any leftover goods are allocated to the agent who receives the last bundle.

We claim that this procedure always results in an IPS allocation. Notice from Definition 7.21 that if a bundle is IPS for an agent, then so is any superset of the bundle. Hence it suffices to show that after $n-1$ bundles are allocated, the last agent still finds the remaining bundle to be IPS. Assume without loss of generality that the bundles are allocated to agents $1, 2, \ldots, n$ in this order, and let u be the utility function of agent n. The claim holds trivially if the empty bundle is IPS for agent n, so assume that it is not. For $1 \le i \le n-1$, let the bundle allocated to agent i be $M_i = X_i \cup Y_i$, where Y_i consists of the last good added to M_i (if M_i is non-empty), and X_i consists of the remaining goods. Let $X = \bigcup_{i=1}^{n-1} X_i$ and $Y = \bigcup_{i=1}^{n-1} Y_i$. In particular, $|Y| \le n-1$.

Let M_n be the bundle allocated to agent n.

- Case 1: $m \ge 2n-1$. By definition of the procedure, agent n does not find any of the bundles $X_1, \ldots, X_{n-1}$ to be IPS. In particular, noting that $Y \subseteq M \setminus X_i$ for each $1 \le i \le n-1$ and taking $B = Y$ in Definition 7.21, we have $u(X_i) < \text{IPS}(n, m) \cdot u(M \setminus Y) = u(M \setminus Y)/n$ for all i. Hence,
$$\begin{aligned} u(M_n) &= u(M) - \sum_{i=1}^{n-1} u(X_i) - \sum_{i=1}^{n-1} u(Y_i) \\ &> u(M) - \frac{n-1}{n} \cdot u(M \setminus Y) - u(Y) = \frac{1}{n} \cdot u(M \setminus Y). \end{aligned}$$
Since $Y \subseteq M \setminus M_n$, bundle M_n is IPS for agent n.

- Case 2: $n \le m \le 2n-1$. First, we show that at most $m-n$ of the first $n-1$ agents can receive at least two goods. Assume for contradiction that at least $m-n+1$ of these agents receive at least two goods, and suppose that the first $m-n+1$ of them are agents $a_1, \ldots, a_{m-n+1}$ in this order. Let j be the first good in agent a_{m-n+1}'s bundle. We claim that the bundle consisting of good j alone is IPS for agent n; this is sufficient for the desired contradiction because agent n should have taken this bundle ahead of agent a_{m-n+1}.

  Before agent a_{m-n+1} receives her bundle, the goods in X allocated to earlier agents are precisely those in the set $X' := \cup_{i=1}^{m-n} X_{a_i}$. Let $Z = M \setminus (X' \cup \{j\})$. Since $|X'| \ge m-n$, we have $|Z| \le m-(m-n)-1 = n-1$. By definition of the procedure, agent n does not find any of the bundles $X_{a_1}, \ldots, X_{a_{m-n}}$ to be IPS. In particular, noting that $Z \subseteq M \setminus X_{a_i}$ and taking $B = Z$ in Definition 7.21, we have $u(X_{a_i}) < u(M \setminus Z)/(m-n+1)$ for all $1 \le i \le m-n$. Hence,
$$\begin{aligned} u(\{j\}) &= u(M) - u(X') - u(Z) \\ &= u(M \setminus Z) - \sum_{i=1}^{m-n} u\left(X_{a_i}\right) \\ &> u(M \setminus Z) - \frac{m-n}{m-n+1} \cdot u(M \setminus Z) \\ &= \frac{1}{m-n+1} \cdot u(M \setminus Z). \end{aligned}$$
Since $Z \subseteq M \setminus \{j\}$, bundle $\{j\}$ is IPS for agent n, so agent n should indeed have taken this bundle ahead of agent a_{m-n+1}. This contradiction means that at most $m-n$ of the first $n-1$ agents can receive at least two goods.

We now proceed in a similar way as in Case 1. By definition of the procedure, agent n does not find any of the bundles $X_1, \dots, X_{n-1}$ to be IPS. In particular, noting that $Y \subseteq M \setminus X_i$ for each $1 \leq i \leq n-1$ and taking $B = Y$ in Definition 7.21, we have $u(X_i) < \text{IPS}(n,m) \cdot u(M \setminus Y) = u(M \setminus Y)/(m-n+1)$ for all i. Hence,

$$\begin{aligned} u(M_n) &= u(M) - \sum_{i=1}^{n-1} u(X_i) - \sum_{i=1}^{n-1} u(Y_i) \\ &> u(M) - \frac{m-n}{m-n+1} \cdot u(M \setminus Y) - u(Y) = \frac{1}{m-n+1} \cdot u(M \setminus Y), \end{aligned}$$

where the inequality holds because at most $m-n$ of the sets X_i are non-empty. Since $Y \subseteq M \setminus M_n$, bundle M_n is IPS for agent n.

The two cases together complete the proof. □

Proposition 7.22 allows us to establish the PoC for paths, which we do next in Theorem 7.23. Conversely, the instances that we use to show the upper bound on the PoC in Theorem 7.23 also show that the factor IPS(n,m) in the existence guarantee of Proposition 7.22 cannot be improved.

Theorem 7.23. *Let $n \geq 2$ and let G be a path. Then*

$$\textit{PoC}(G,n) = \begin{cases} n & \textit{if } m \geq 2n-1; \\ m-n+1 & \textit{if } n \leq m < 2n-1; \\ 1 & \textit{if } m < n. \end{cases}$$

Proof. If $m < n$ the PoC is 1, so assume that $m \geq n$. We will show that PoC$(G,n) = 1/\text{IPS}(n,m)$.

First, we show that PoC$(G,n) \leq 1/\text{IPS}(n,m)$. Take an arbitrary utility function u. Applying Proposition 7.22 to n agents who have the same utility function u, we find that there exists a connected IPS allocation. This means each agent i receives a bundle M_i for which there exists a set $B_i \subseteq M \setminus M_i$ with $|B_i| \leq n-1$ such that $u(M_i) \geq \text{IPS}(n,m) \cdot u(M \setminus B_i)$. Since $|M \setminus B_i| \geq m-n+1$, Lemma 7.19 implies that $u(M \setminus B_i) \geq \text{MMS}(u,n)$. Consequently, we have

$$u(M_i) \geq \text{IPS}(n,m) \cdot u(M \setminus B_i) \geq \text{IPS}(n,m) \cdot \text{MMS}(u,n)$$

for all agents i. Hence $(M_1, \dots, M_n)$ is a connected partition with each part having value at least IPS$(n,m) \cdot$ MMS(u,n). It follows that PoC$(G,n) \leq 1/\text{IPS}(n,m)$.

Next, we show that PoC$(G,n) \geq 1/\text{IPS}(n,m)$. We consider two cases.

- Case 1: $m \geq 2n-1$. Consider a utility function u with value $1, n, 1, \dots, n, 1$ for the first $2n-1$ vertices on the path (so exactly n vertices have value 1), and value 0 for the remaining vertices. We have MMS$(u,n) = n$. On the other hand, one can check that in any connected partition into n parts, at least one of the parts has value at most 1. Hence the PoC is at least $n = 1/\text{IPS}(n,m)$.

- Case 2: $n \leq m < 2n-1$. Consider a utility function u with value $1, m-n+1, 1, \dots, m-n+1, 1$ for the first $2m-2n+1$ vertices on the path (so $m-n+1$ vertices have value 1 while $m-n$ vertices have value $m-n+1$), and value $m-n+1$ for the remaining $2n-1-m$ vertices. In total, $n-1$ vertices have value $m-n+1$, and $m-n+1$ vertices have value 1. We have $\text{MMS}(u,n) = m-n+1$. On the other hand, one can check that in any connected partition into n parts, at least one of the parts has value at most 1. Hence the PoC is at least $m-n+1 = 1/\text{IPS}(n,m)$.

In both cases we have $\text{PoC}(G,n) \geq 1/\text{IPS}(n,m)$, completing the proof. □

Note that in order to compute a connected allocation for n agents on a path such that every agent receives at least a $1/\text{PoC}(G,n) = \text{IPS}(n,m)$ fraction of their MMS, we can use the algorithm in Proposition 7.22, which runs in polynomial time, to compute a connected IPS allocation. The first part in the proof of Theorem 7.23 implies that this allocation fulfils the desired guarantee.

7.4 Envy-Freeness Relaxations

Having extensively studied maximin share guarantees in the presence of connectivity requirements in the previous section, we now turn our attention to relaxations of envy-freeness. We again determine the price that we have to pay in order to maintain connectivity—intuitively, the less connected the graph is, the higher this price becomes. Unless specified otherwise, we allow agents to have arbitrary monotonic utilities in this section.

We say that a graph G *guarantees* EFk for n agents if for all permitted utilities of the n agents, there exists a connected EFk allocation.

7.4.1 Two Agents

For two agents, Bilò et al. [45] characterized the set of graphs that always admits an EF1 allocation regardless of the agents' utilities. Their characterization is based on the observation that such graphs necessarily admit a vertex ordering to which a discrete variant of the cut-and-choose protocol can be applied—in other words, the ordering is bipolar. The family of graphs that admit a bipolar ordering can be characterized using the block decomposition of a graph. A *block* is a maximal biconnected subgraph of a graph, and a *cut vertex* is a vertex whose removal increases the number of connected components in the graph. The *block decomposition* of a graph G is a bipartite graph $B(G)$ with all blocks of G on one side and all cut vertices of G on the other side; there is an edge between a block and a cut vertex in $B(G)$ if and only if the cut vertex belongs to the block in G.[7]

Proposition 7.24 (Bondy and Murty [48, Proposition 5.3]). *For any connected graph G, each pair of blocks share no edge and at most one cut vertex, and the block decomposition $B(G)$ is a tree.*

[7] We refer to the paper of Bilò et al. for examples of graphs and their block decompositions.

Bilò et al. [45] showed that a connected graph G guarantees EF1 for two agents if and only if a bipolar ordering exists in G, i.e., the blocks of G can be arranged into a path.

Proposition 7.25 (Bilò et al. [45, Theorem 10])**.** *The following four conditions are equivalent for every connected graph G:*[8]

1. *The block decomposition $B(G)$ is a path;*
2. *G admits a bipolar ordering;*
3. *G guarantees EF1 for two agents with arbitrary monotonic utilities;*
4. *G guarantees EF1 for two agents with identical binary utilities.*

Bilò et al.'s characterization allows us to identify graphs for which an EF1 allocation always exists in the case of two agents. However, for the remaining graphs, it does not provide any fairness guarantee. Our next result generalizes their characterization by giving the best possible EFk guarantee that can be made for each specific graph. In particular, we will show that a graph G guarantees EFk for two agents if and only if G admits a bipolar ordering over a subset of the vertices where each vertex in the ordering has at most $k-1$ vertices 'hanging' from it.

To formalize this idea, it will be useful to define the following notions. For each path P of $B(G)$, we denote by $C(P)$ the set of cut vertices that belong to some block in P. Given a path P in the block graph $B(G)$ of a graph G, for any vertex v of G that is not contained in any block in P, we define its *guardian* to be the cut vertex v' closest to v in $B(G)$ that belongs to some block in P (see Figure 7.3 for an example); we say that v is a *dependent* of v'. For a given graph, we define a *merge* on a subset V of vertices forming a connected subgraph to be an operation where we replace the vertices in V by a single vertex v, and there is an edge between v and another vertex w in the new graph exactly when w is adjacent to at least one vertex of V in the original graph. A path in a tree is said to be *maximal* if each of its end vertices is a leaf of the tree.

Theorem 7.26. *For any connected graph G and positive integer k, the following four conditions are equivalent:*

(1) *There exists a path P in the block decomposition $B(G)$ such that each cut vertex that belongs to some block in P has at most $k-1$ dependents;*

(2) *The vertices of G can be partitioned into disjoint subsets $V_1, V_2, \ldots, V_r$ such that each V_j forms a connected subgraph of size at most k in G, and if we merge the vertices in every set V_j separately, the resulting graph admits a bipolar ordering;*

(3) *G guarantees EFk for two agents with arbitrary monotonic utilities;*

[8] Bilò et al. [45] used a slightly stronger definition of EF1 that they called "envy-freeness up to one *outer* good". In their definition, one is only allowed to remove a good if doing so leaves the remaining bundle connected. It can be verified that their result also holds for the standard definition of EF1.

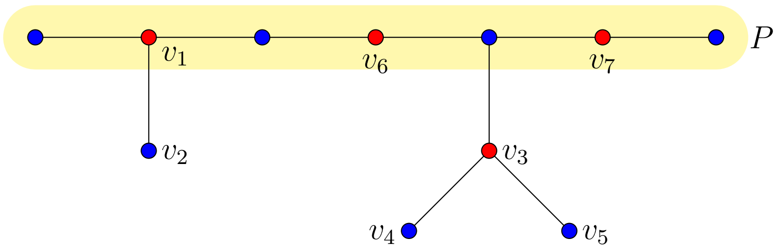


Figure 7.3: An example of a block decomposition $B(G)$ in the proof of Theorem 7.26. Blue vertices correspond to blocks in G and red vertices correspond to cut vertices in G. Here, $C(P) = \{v_1, v_3, v_6, v_7\}$. In this example, v_1 is the guardian of all vertices in block v_2 except itself, v_3 is the guardian of all vertices in blocks v_4 and v_5 except itself, while v_6 and v_7 are not guardians of any vertices.

(4) *G guarantees EFk for two agents with identical binary utilities.*

Proof. Consider the block decomposition $B(G)$, which is a tree due to Proposition 7.24.

To show (1) $\implies$ (2), suppose that there exists a path P in the block decomposition $B(G)$ such that each cut vertex in $C(P)$ has at most $k-1$ dependents. Take each set V_j in the theorem statement to consist of a vertex in $C(P)$ along with all of its dependents. Clearly, at most k vertices belong to each V_j. Also, each V_j is connected since the vertices in V_j form a connected subgraph of the block decomposition. Let G' be the graph resulting from the merge operations on each V_j separately. The block decomposition of G' is a path, and hence G' admits a bipolar ordering by Proposition 7.25.

To show (2) $\implies$ (3), suppose that the vertices of G can be partitioned into disjoint subsets $V_1, V_2, \ldots, V_r$ as defined in the statement of the theorem. We will show that G guarantees EFk for two agents. Consider arbitrary monotonic utilities of the two agents u_i for $i = 1, 2$. Let G' be the graph resulting from the merge operations on each V_j for $j = 1, 2, \ldots, r$. We define the utility functions u'_i on G' for $i = 1, 2$, where the value of an agent for each bundle M' is equal to her value for all vertices of G that are merged into the vertices of M'. Specifically, for each $i = 1, 2$ and each bundle M' in G',

$$u'_i(M') = u_i\left(\bigcup_{V_j \in M'} V_j\right).$$

Note that each u'_i remains monotonic and, by our assumption, G' admits a bipolar ordering. Thus, by Proposition 7.25, G' admits a connected EF1 allocation (M'_1, M'_2) with the utilities u'_i, so an agent's envy can be eliminated by removing a vertex of G' from the other agent's bundle. Consider the corresponding allocation (M_1, M_2) of G, where $M_i = \bigcup_{V_j \in M'_i} V_j$ for $i = 1, 2$. Since each vertex of G' is a merge of at most k vertices, any envy that results from this allocation can be eliminated by removing at most k vertices, and so the allocation (M_1, M_2) is a connected EFk allocation of G.

The implication (3) $\implies$ (4) is immediate. To show (4) $\implies$ (1), suppose that for every path P of $B(G)$, there exists some cut vertex in $C(P)$ with at least k dependents. We will show that there exist identical binary utility functions for which the graph G does not admit an EFk allocation.

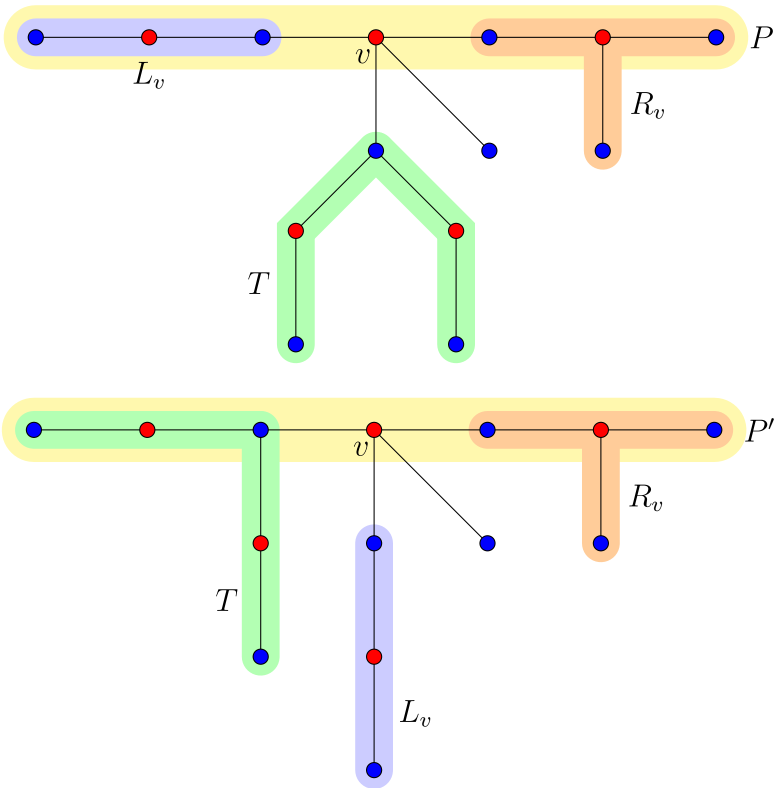


Figure 7.4: An example of a switch operation on the tree $B(G)$ in Case 1 of the proof of Theorem 7.26. The top and bottom figures are the trees before and after the operation, respectively.

Let $k^* \geq 1$ be the smallest number for which there exists a *maximal* path P in $B(G)$ such that each cut vertex in $C(P)$ is the guardian of at most $k^* - 1$ vertices in G. Choose a maximal path P in $B(G)$ where each cut vertex in $C(P)$ has at most $k^* - 1$ dependents; if several such paths exist, choose one that minimizes the number of vertices in $C(P)$ with exactly $k^* - 1$ dependents. By definition of k^*, we have $k^* - 1 \geq k$. Hence it suffices to show the existence of identical binary utility functions for which the graph G does not admit an EF$(k^* - 1)$ allocation.

Let $v \in C(P)$ be a cut vertex with $k^* - 1$ dependents. It could be that v is on the path P itself (e.g., vertices v_1, v_6, and v_7 in Figure 7.3), or v is not on the path P but belongs to some block in P (e.g., vertex v_3 in Figure 7.3). We consider the two cases separately.

Case 1: v is in the path P itself. Let L_v and R_v be the subtree of the tree $B(G)$ rooted at v starting with each of the two blocks adjacent to v on the path P, respectively. For each subtree besides L_v and R_v of the tree $B(G)$ rooted at v, with a block adjacent to v in $B(G)$ as the root of the subtree, define its *size* to be the number of dependents of v in G belonging to at least one block in the subtree. Note that the size can be different from the number of vertices in the subtree in $B(G)$.

Suppose that T is a largest subtree among such subtrees and has size $r \leq k^* - 1$. We claim that at least r vertices of G (excluding v) belong to some block in L_v. Assume for contradiction that there are at most $r - 1$ such vertices. In $B(G)$, we switch L_v with T and

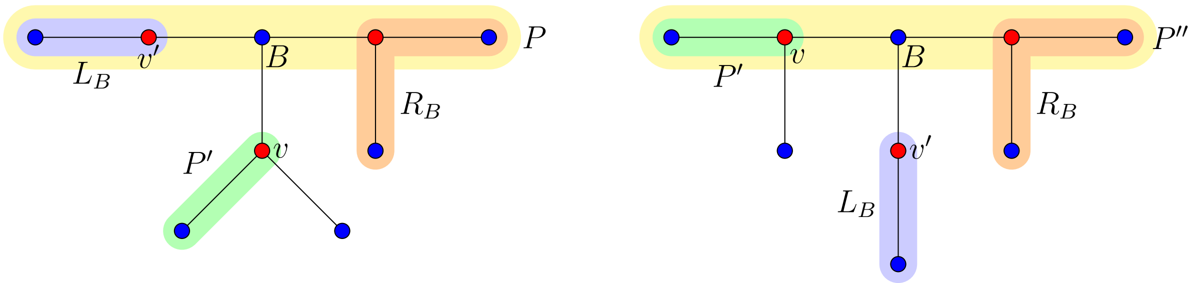


Figure 7.5: An example of a switch operation on the tree $B(G)$ in Case 2 of the proof of Theorem 7.26. The left and right figures are the trees before and after the operation, respectively.

choose an arbitrary path of T that contains a leaf of $B(G)$ to be on the main path P (see Figure 7.4). Let P' denote the new maximal path. Since v loses at least r dependents and gains at most $r-1$ new dependents, v now has at most k^*-2 dependents with respect to P'. Moreover, since T has size at most k^*-1, each of the new cut vertices in $C(P')$ has at most k^*-2 dependents. Hence we have decreased the number of cut vertices with k^*-1 dependents by at least 1. This gives the desired contradiction. The same argument shows that at least r vertices of G (excluding v) belong to some block in R_v.

Consider two agents who have the same binary utility function with value 1 for v, its k^*-1 dependents, r arbitrary vertices of G (besides v) belonging to some block in L_v, and r arbitrary vertices of G (besides v) belonging to some block in R_v, and value 0 for the remaining vertices. The total value of an agent is $2r+k^*$. In any connected allocation, one of the agents does not receive v. This agent receives value at most r, while the remaining goods are worth at least $r+k^*$. It follows that the allocation cannot be EF(k^*-1).

Case 2: v is not in the path P but belongs to some block B in P. Let L_B and R_B be the subtree of the tree $B(G)$ rooted at B starting with each of the two cut vertices adjacent to B on the path P, respectively. We claim that at least k^* vertices of G belong to some block in L_B. Assume for contradiction that there are at most k^*-1 such vertices. Let v' be the cut vertex in L_B adjacent to B. In $B(G)$, we switch L_B with v and its dependents, and choose an arbitrary path P' that starts with v and contains at least one of its dependents as well as a leaf of $B(G)$ to be on the main path P (see Figure 7.5). Let P'' denote the new maximal path. Since L_B contains at most k^*-1 vertices (which include v'), v' now has at most k^*-2 dependents with respect to P''. Moreover, the subtree that replaced L_B has at most k^* vertices. Among these vertices, v and at least one other vertex belong to P', which is now on the new path P'', so any new cut vertex has at most k^*-2 dependents. Hence we have decreased the number of cut vertices with k^*-1 dependents by at least 1. This gives the desired contradiction. The same argument shows that at least k^* vertices of G belong to some block in R_B.

Consider two agents who have the same binary utility function with value 1 for v, its k^*-1 dependents, k^* arbitrary vertices of G belonging to some block in L_B, and k^* arbitrary vertices of G belonging to some block in R_B, and value 0 for the remaining vertices. The total value of an agent is $3k^*$. In any connected allocation, one of the agents receives a bundle whose vertices of value 1 are contained in L_B, R_B, or the set with v and its dependents. This agent receives value at most k^*, while the remaining goods are worth at least $2k^*$. It follows that the allocation cannot be EF$(k^* - 1)$.

Hence, in both cases there exist identical binary utility functions for which the graph does not admit an EF$(k^* - 1)$ allocation, as claimed. □

Theorem 7.26 allows us to determine in polynomial time the optimal k such that a given graph always admits an EFk allocation, as well as to compute such an allocation. To do so, we compute the block decomposition $B(G)$ of the graph—this can be done in linear time [100]. We then determine the value of k^* in the proof of the theorem, which we have shown to be equal to the optimal value of k; this can be done by testing all pairs of vertices as endpoints of the path P. Finally, we compute a bipolar ordering of the vertices belonging to P—again, this takes linear time [84]—and apply the EF1 algorithm of Bilò et al. [45] on the merged vertices.

Theorem 7.26 also yields a short proof that every graph admits an EF$(m-2)$ allocation. Moreover, we show that the bound $m - 2$ is tight for stars.

Proposition 7.27. *Let $n = 2$, and let G be any graph with at least three vertices. There exists a connected EF$(m - 2)$ allocation to the two agents.*

Proof. Since the graph contains at least three vertices, it has a path of length 2; let the three vertices on this path be v_1, v_2, v_3 and $V_1 = \{v_1\}, V_2 = \{v_2\}, V_3 = \{v_3\}$. Add the remaining vertices to these sets arbitrarily so that each set remains connected. Clearly, each set contains at most $m - 2$ vertices. Theorem 7.26 then implies that an EF$(m - 2)$ allocation exists. □

Proposition 7.28. *Let $n = 2$, and let G be a star with at least two edges. There exist identical binary utility functions of the two agents such that a connected EF$(m - 3)$ allocation does not exist.*

Proof. Consider two agents who have value 1 for every good. In any connected allocation, one of the agents receives at most one good, while the other agent receives at least $m - 1$ goods. Hence the allocation cannot be EF$(m - 3)$. □

Next, we consider a stronger fairness notion, EFX. It is known that for two agents with arbitrary monotonic utilities, an EFX allocation always exists [132]. We show that if we consider connected allocations, the statement remains true only if the graph is complete.

Theorem 7.29. *Let $n = 2$, and let G be a non-complete graph. There exist identical additive utility functions of the two agents such that no connected allocation is EFX.*

Proof. Pick an arbitrary missing edge of G, and let $\epsilon > 0$ be a sufficiently small constant. Suppose that the two agents have value 2 for each of the two vertices with a missing edge (call them v_1 and v_2), and value $3, \epsilon, \epsilon, \ldots, \epsilon$ for the remaining vertices (call the first vertex v_3).

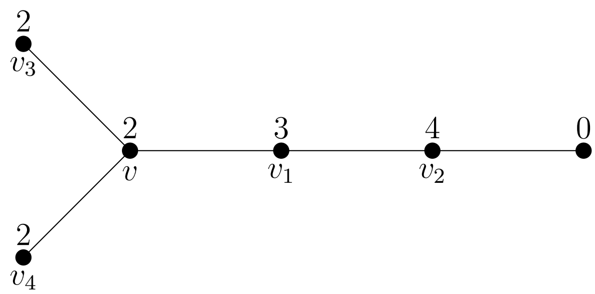


Figure 7.6: Example of an instance in the proof of Theorem 7.30.

Assume for contradiction that there exists a connected EFX allocation. In this allocation, neither of the agents can receive v_3 together with one (or both) of v_1, v_2. So one of the agents must receive v_1 and v_2, while the other agent receives v_3. If the first agent also receives one of the remaining vertices, the allocation cannot be EFX. So the second agent receives all of the remaining vertices. However, the resulting allocation is not connected, a contradiction. □

7.4.2 Three Agents

We now address the case of three agents. Bilò et al. [45] showed that in this case, an EF1 allocation is guaranteed to exist if the graph contains a Hamiltonian path[9] or if it is a star with three edges. We extend this result by characterizing all trees and complete bipartite graphs that always admit an EF1 allocation. Recall that we allow agents to have arbitrary monotonic utilities in this section.

Theorem 7.30. *Let G be a tree. Then G guarantees EF1 for three agents if and only if G is either a path, or a star with three edges.*

Proof. The 'if' direction was already shown by Bilò et al. [45]; we establish the 'only if' direction. Assume that G is neither a path, nor a star with three edges. Suppose first that there is a vertex v with degree at least 4. Consider three agents who have identical utilities with value 1 on v and four of its neighbours, and 0 on all other vertices. In any connected allocation, an agent who does not get v receives value at most 1, while the bundle of the agent who gets v has value at least 3 to her. Hence the allocation is not EF1.

Suppose now that every vertex has degree at most 3. Since G is not a path, there is a vertex v with degree 3. Moreover, since G is not a star, one of the branches from v contains at least two vertices, say a branch starting with a neighbour v_1 of v followed by another vertex v_2. Let v_3, v_4 be the two other vertices adjacent to v. Consider three agents who have identical utilities with value 2 for v, v_3, v_4, value 3 for v_1, value 4 for v_2, and value 0 for all other vertices (see Figure 7.6). Consider any connected allocation; in what follows, we will only be concerned with goods of non-zero value. First, assume that one of the agents receives either only v_3 or only v_4, and obtains value at most 2. If another agent receives at least three goods, the allocation is clearly not EF1. So each of the other two agents receives

[9] Clearly, it suffices to prove the claim when the graph is a path.

exactly two goods, which means one of them receives v_1 and v_2. This bundle is worth 3 to the first agent even after removing the most valuable good, so the allocation cannot be EF1. Hence one of the agents receives v_3, v_4, and v. But then the agent who does not receive v_2 will envy this agent even after removing one good. □

Next, we consider complete bipartite graphs. Denote by $K_{a,b}$ the complete bipartite graph with a vertices on the left (call this set of vertices L) and b vertices on the right (call this set of vertices R). We start by showing that if $a, b \geq 3$, there always exists a connected EF1 allocation. In fact, we present a generalization that holds for any number of agents.

Proposition 7.31. *Let $n \geq 2$, and let G be a complete bipartite graph $K_{a,b}$ with $a, b \geq n$. Then G guarantees EF1 for n agents.*

Proof. We enhance the *envy cycle elimination algorithm* of Lipton et al. [113], which computes an EF1 allocation for any number of agents. The algorithm works by allocating one good at a time in arbitrary order—we will exploit this freedom in choosing the order. It also maintains an *envy graph*, which has the agents as its vertices, and a directed edge $i \to j$ if agent i envies agent j with respect to the current (partial) allocation. At each step, the next good is allocated to an agent with no incoming edge, and any cycle that arises as a result is eliminated by giving j's bundle to i for each edge $i \to j$ in the cycle. This allows the algorithm to maintain the invariant that the envy graph is cycle-free, and so there exists an agent with no incoming edge before each allocation of a good.

We apply the envy cycle elimination algorithm by choosing a careful order of the goods to allocate. Since $a \geq n$ and every agent is unenvied at the beginning, we can first pick n goods from L and allocate one of them to each agent. After this point, we may no longer have control over which agent to choose next. Take an agent with no incoming edge in the envy graph. If the agent has already received a good from R, allocate to her a good from L if one still remains, otherwise allocate a good from R. Else, the agent has not received a good from R. In this case, allocate to her a good from R if one still remains, otherwise allocate a good from L. The pseudocode is presented as Algorithm 9.

The resulting allocation is EF1 [113]; we now show that it is connected. Every agent receives a good from L in the first phase of the algorithm. Note that if an agent receives at least one good from both L and R, her bundle is guaranteed to be connected. So it suffices to show that an agent will never receive more than one good from L without receiving a good from R. By construction, an agent who already has a good from R will take goods from L unless L is already empty. Since $b \geq n$, this means that as long as some agent has not received a good from R and the algorithm has not terminated, there is at least one good from R left. This establishes the desired claim.

Note that since the envy cycle elimination algorithm runs in time polynomial in the number of agents and goods [113], the proof of Proposition 7.31 also yields a polynomial-time algorithm that computes a connected EF1 allocation for any number of agents. □

Algorithm 9: Enhanced Envy Cycle Elimination Algorithm

```
Input: Agents $N$, complete bipartite graph with vertices $L$ and $R$ on the left and
       right respectively, and utility functions $u_1, u_2, \ldots, u_n$.
$M_1, M_2, \ldots, M_n \leftarrow \emptyset$
$r_1, r_2, \ldots, r_n \leftarrow$ false // Indicate whether agent $i$ has received a good
    from $R$
for $i \in N$ do Move an arbitrary good from $L$ to $M_i$.
while $L \cup R \neq \emptyset$ do
    Eliminate cycles in the envy graph.
    $i \leftarrow$ any agent with no incoming edge in the envy graph
    if $r_i = true$ then
        if $L \neq \emptyset$ then
            Move an arbitrary good from $L$ to $M_i$.
        else
            Move an arbitrary good from $R$ to $M_i$.
    else
        if $R \neq \emptyset$ then
            Move an arbitrary good from $R$ to $M_i$.
            $r_i \leftarrow$ true
        else
            Move an arbitrary good from $L$ to $M_i$.
return $(M_1, M_2, \ldots, M_n)$
```

If the agents have additive utilities, we can also obtain an EF1 allocation via a simple polynomial-time algorithm—the "double round-robin algorithm". The algorithm proceeds by running the classical round-robin algorithm twice, once on L and once on R, with opposite orderings of the agents. The pseudocode is shown as Algorithm 10.

Since $a, b \geq n$, every agent receives at least one good from each of L and R, so the resulting allocation is connected. We claim that it is EF1. To see this, consider two agents i, i' with $i < i'$. When allocating each of the sets L and R, we consider a *round* to begin when i picks a good, and end just before the next time i picks a good (or when the set runs out of goods). During the allocation of L, in each round i picks before i'. Since the utilities are additive, i does not envy i' with respect to the goods in L. Similarly, i does not envy i' in each round during the allocation of R. The only possible source of envy is before the first round starts, when i' picks her first good. However, this means that the envy can be eliminated if we remove this good from the bundle of i'. Hence i does not envy i' up to one good in total; an analogous argument shows that i' also does not envy i up to one good. Since i and i' are arbitrary, the allocation is EF1.

With Proposition 7.31 in hand, we now proceed with the characterization for complete bipartite graphs.

Theorem 7.32. *Let a, b be positive integers with $a \leq b$. The graph $K_{a,b}$ guarantees EF1 for three agents if and only if one of the following holds:*

Algorithm 10: Double Round-Robin Algorithm

$M_1, M_2, \ldots, M_n \leftarrow \emptyset$
$i = 1$
while $L \neq \emptyset$ **do**
  $j \leftarrow$ highest-valued good in L according to u_i.
  Move j from L to M_i.
  $i \leftarrow i + 1$
  if $i = n + 1$ **then** $i = 1$
$i = n$
while $R \neq \emptyset$ **do**
  $j \leftarrow$ highest-valued good in R according to u_i
  Move j from R to M_i.
  $i \leftarrow i - 1$
  if $i = 0$ **then** $i = n$
return $(M_1, M_2, \ldots, M_n)$

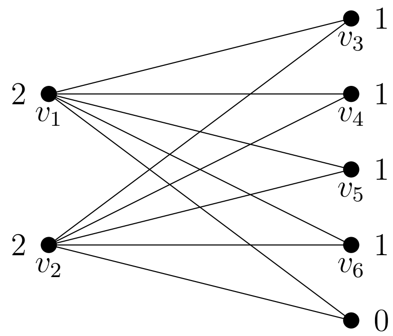


Figure 7.7: Example of an instance in the proof of Theorem 7.32.

1. $a = 1$ *and* $b \leq 3$*;*

2. $a = 2$ *and* $b \leq 3$*;*

3. $a, b \geq 3$*.*

Proof. The case $a = 1$ is covered by Theorem 7.30 and the case $a \geq 3$ by Proposition 7.31, so assume that $a = 2$. If $b \leq 3$, then G contains a Hamiltonian path, so the existence of an EF1 allocation follows from the result of Bilò et al. [45]. Else, let $b \geq 4$. Consider three agents who have identical utilities with value 2 on each of the two vertices $v_1, v_2 \in L$, value 1 on four of the vertices $v_3, v_4, v_5, v_6 \in R$, and value 0 for the remaining vertices (see Figure 7.7). Consider any connected allocation. If v_1 and v_2 are allocated to the same agent, this agent must also receive at least one of the vertices from R, and the allocation is not EF1. Else, one agent receives v_1 and another agent receives v_2. Now, the third agent can get at most one vertex from R and therefore receives value at most 1. This means that one of the first two agents receives one of v_1 and v_2 along with at least two of v_3, v_4, v_5, v_6. This agent is envied by the third agent even after we remove a good. It follows that the allocation cannot be EF1. □

7.5 Conclusion and Future Work

In this chapter, we have studied the fair allocation of indivisible goods under connectivity constraints and provided an extensive set of results on the guarantees that can be achieved via maximin share guarantee and relaxations of envy-freeness for various classes of graphs. For maximin share guarantee, we establish a link between the graph-specific maximin share and the well-studied maximin share through our price of connectivity (PoC) notion. We present a number of bounds on the PoC, several of which are tight, and leave a tempting conjecture that would settle the two-agent case if it holds. On the envy-freeness front, we classify all connected graphs based on the strongest relaxation with guaranteed existence in the case of two agents—thereby also quantifying the price that we have to pay with respect to fairness for each graph—and characterize the set of trees and complete bipartite graphs that always admit an EF1 allocation for three agents. Extending our results beyond three agents is a challenging problem: even when the graph is a path, the only known proof of EF1 existence for four agents employs arguments based on Sperner's lemma, and the corresponding question remains open when there are at least five agents [45].

Our results on envy-freeness relaxations hold for agents with arbitrary monotonic utilities. On the other hand, as is the case in most of the literature, our results on maximin share guarantee rely on the assumption that the agents' utility functions are additive. Maximin share guarantee beyond additive utilities has been studied by Barman and Krishnamurthy [27] and Ghodsi et al. [93]; for example, they showed that a constant approximation of the maximin share can be achieved for any number of agents with submodular utilities when the graph is complete. Since complementarity and substitutability are common in practice, it would be interesting to see how the graph-based approximations that we obtain in this chapter change as we enlarge the class of utility functions considered. Indeed, as Plaut and Roughgarden [132] noted, there is a rich landscape of problems to explore in fair division with different classes of utility functions, and the graphical setting is likely to be no exception.

Finally, while our results in this work provide fairness guarantees that hold regardless of the agents' utilities, better guarantees can be obtained in many instances if we take the utilities into account. For example, even though an envy-free allocation does not always exist, it is known that such an allocation exists most of the time when utilities are drawn at random [78, 117]. On a complete graph, deciding the existence of an envy-free allocation is NP-hard even for two agents with identical utilities [113]. By contrast, this problem can be solved efficiently on a tree or a cycle for any constant number of agents, since we can simply go through all of the (polynomially many) connected allocations; yet, the problem again becomes NP-hard even on a path if the number of agents is non-constant [51]. Similar computational questions can be asked for other combinations of graphs and fairness notions without guaranteed existence, and we believe that these questions constitute an important direction that deserves to be pursued in future work.

Part III

Other Settings

Chapter 8

Truthful Cake Sharing

8.1 Introduction

In previous chapters, we have studied various aspects of fair division, a fundament problem in social choice theory. When the resource is heterogeneous and divisible, this problem is commonly known as *cake cutting*, with the cake serving as a metaphor for the heterogeneous resource. Cake cutting has been extensively studied for over half a century in mathematics and economics, and more recently in computer science [55, 112, 134, 138].

In this chapter, we consider a variant of the classic cake cutting problem where instead of competing with one another for the cake, the agents all share the same subset of the cake, which must be chosen subject to a length constraint. We refer to this setting as *cake sharing*. The cake sharing problem captures many real-world scenarios, such as when a group of agents need to decide the time periods for which they should reserve a sports facility or a conference room for collective use given their limited budget, or when a group of users seek to agree upon the files to store in a shared cache memory. Our goal is to design cake sharing mechanisms that are both *truthful* and *fair*. Truthfulness requires that it should be in every agent's best interest to report her true underlying preferences to the mechanism. A truthful mechanism makes it easy for agents to participate in, as they do not have to act strategically and reason about beneficial manipulations; it also simplifies the job of the mechanism designer when reasoning about the possible behaviour of the agents. Note that truthfulness by itself is easy to obtain, for example by ignoring the agents' reports completely and allocating a prespecified subset of the cake. However, this is a patently unfair mechanism because it leaves any agent who has no value for that subset empty-handed. Is there a mechanism that is truthful and at the same time satisfies a certain degree of fairness for all agents?

Two mechanisms that have been used in a variety of resource allocation settings and often shown to exhibit attractive fairness properties are the *maximum Nash welfare (MNW) solution* and the *leximin solution*. The MNW solution chooses an allocation that maximizes the product of the agents' utilities among all feasible allocations. It is known this solution satisfies envy-freeness with divisible resources, and envy-freeness up to one good with indivisible resources [69]. The leximin solution considers all feasible allocations that maximize the minimum among the agents' utilities; among all such allocations, it considers those maximizing the second smallest utility, and so on. Leximin is also shown to satisfies proportionality and envy-freeness for a wide range of settings [107]. Due to their optimization nature, both solutions fulfil an important economic efficiency criterion of *Pareto optimality*: there is no other feasible outcome that makes some agent better off and no agent worse off compared to the chosen outcome. Indeed, any such improved outcome would also be an improvement with respect to the corresponding optimization objective. Given the broad appeal of the two mechanisms, are they appropriate choices for our cake sharing setting, especially from the truthfulness perspective?

As is standard in the cake cutting literature, we model the cake as an interval $[0, 1]$; for a given parameter $\alpha \in [0, 1]$, a subset of length at most α of the cake can be collectively allocated to the agents. We assume that the agents have *piecewise uniform* utilities, meaning that each agent has a desired subset of the cake which she values uniformly. Except in Section 8.6, we also assume that once a mechanism chooses a subset of the cake, it can "block" each agent from accessing certain parts of the cake, usually those that the agent does not desire according to her report. We remark here that blocking can be easily implemented in the aforementioned applications by restricting access to the sports facility, conference room, or files in a cache memory.

In Section 8.3, we focus on the leximin solution. Our main technical result establishes the truthfulness of the solution for any number of agents with arbitrary piecewise uniform utilities. At a high level, our proof proceeds by showing that the leximin solution is immune to certain types of manipulations, and then arguing that this immunity is sufficient to protect the solution against all possible manipulations. Along the way, we introduce the notion of an ε*-change*—a tiny change from one utility vector or allocation towards another—which may be useful in related settings. We also show that each agent receives the same utility in all leximin allocations, which means that tie-breaking is inconsequential, and that such an allocation can be computed in polynomial time.

Since truthfulness by itself is trivial to achieve as we explained earlier, we consider in Section 8.4 the fairness of mechanisms. We measure fairness using the *egalitarian ratio*, which is defined as the worst-case egalitarian welfare over all instances, where we normalize the agents' utilities when computing their egalitarian welfare. We show that for any α and any number of agents n, the leximin solution achieves an egalitarian ratio of exactly $\frac{\alpha}{n-(n-1)\alpha}$. Moreover, we prove that this ratio is already optimal among all mechanisms that are truthful and *position oblivious* (see Definition 8.17). Our results in Sections 8.3 and 8.4 establish the leximin solution as an attractive mechanism in the setting of cake sharing.

Next, in Section 8.5, we turn our attention to the MNW solution. We show that the solution is equivalent to the leximin solution in the case of two agents, and is therefore truthful in that case. In general, however, a result of Aziz et al. [24, Theorem 3] implies the non-truthfulness of the MNW solution in our setting. We strengthen their result by showing that MNW is not truthful even when an agent is only allowed to report a subset of her true desired piece.[1] Moreover, in contrast to Aziz et al.'s example, the symmetry structure in our example allows us to provide a relatively short proof of the non-truthfulness that can be easily verified by hand.

Then, we demonstrate in Section 8.6 that the ability to block is crucial for the truthfulness of mechanisms. In particular, we show that no truthful, Pareto optimal, and position oblivious mechanism can achieve a positive egalitarian ratio when blocking is not allowed.

Finally in Section 8.7, we consider an extension of the cake sharing model where the cost of selecting the cake is piecewise constant and show that a generalization of the leximin solution is still truthful and achieves the optimal egalitarian ratio.

8.1.1 Related Work

While the model of cake sharing is new to the best of our knowledge, the selection of a collective subset from a given set subject to a size or budget constraint has been studied in several lines of work. In *multiwinner voting* (see the survey of Faliszewski et al. [85]), the goal is to choose a certain number of candidates to form a committee, where criteria can include excellence and diversity. In that setting, Peters [131] proved that no rule can simultaneously satisfy a form of fairness and a form of truthfulness when agents have approval preferences (analogous to piecewise uniform utilities in our setting). A key difference between multiwinner voting and cake sharing is that the candidates in the former are discrete and cannot be divided into arbitrarily small pieces. Variants where discrete items instead of candidates are selected have also been considered [116, 145].

In the last few years, a long list of papers have addressed the problem of *participatory budgeting*, where the citizens decide how a public budget should be spent on possible projects in their community (see the survey of Aziz and Shah [17]). Some models assume that projects are discrete (each project can either be fully completed or not at all), while others assume that they are divisible (partial completion of a project yields some utility to the citizens). In either case, there is a prespecified set of projects and the preference of an agent within a project is uniform, so participatory budgeting cannot capture our cake sharing model where there is no predetermined division of the cake into homogeneous units.

Aziz et al. [24] considered a probabilistic voting setting where agents have dichotomous preferences over alternatives and the goal is to output a probability distribution over the alternatives. Their model corresponds to a special case of our model where interval $[(j-1)/m, j/m]$ represents alternative j for each $j = 1, 2, \dots, m$, and $\alpha = 1/m$; in this special case, agents are not allowed to have "breakpoints" that are not multiples of $1/m$ in their utility functions (see the precise definition in Section 8.2). Like us, they showed that the leximin solution is truthful.[2] Our results on the leximin solution generalize and strengthen

[1] We remark that subset manipulation is a highly restricted form of manipulation. Indeed, Peters [131] noted that reporting a subset is a "particularly simple fashion" of manipulating, and used subset manipulation as the "official notion" of truthfulness.

[2] They called the notion *excludable strategyproofness*, which is equivalent to truthfulness with blocking in our setting.

theirs in two important ways. First, we allow agents to report *arbitrary* breakpoints—as discussed in the previous paragraph, this considerably enlarges the strategy space of the agents and introduces an aspect that cannot be captured by Aziz et al.'s or any other known participatory budgeting models. Second, we establish a tight bound on the egalitarian ratio and show that the leximin solution achieves this bound. Therefore, our results make a significantly stronger case in favour of the leximin solution.

Friedman et al. [89] studied a model in which agents share a cache memory unit, focusing on truthfulness and fairness like we do. In their model, each agent has a private file that no other agent is interested in, and there is a large public file that may be of interest to multiple agents. The challenge is to elicit the true ratio between each agent's utility for the public file and that for her private file. Similarly to participatory budgeting, the files are predetermined and the agents' preferences are uniform within each file, so their model does not encompass cake sharing. These authors also demonstrated that the ability to block can help mechanisms achieve better guarantees in their setting, in particular by preventing "free riding".

Truthfulness in cake *cutting* has been considered in several papers [18, 32, 34, 61, 75, 106, 121, 123]. Like our chapter, a number of these papers also address the case of piecewise uniform utilities. Two important fairness properties in cake cutting are *envy-freeness* and *proportionality*. Note that envy-freeness is always fulfilled in our setting (as long as the mechanism does not block any agent's valued cake), since all agents share the same subset of the cake. On the other hand, proportionality has a similar flavour as our egalitarian ratio notion, where we want to guarantee a certain level of utility for every agent.

Finally, both the leximin and MNW solutions have been examined in a variety of settings and often shown to exhibit desirable properties [24, 47, 59, 69, 98, 107, 132, 144].

8.2 Cake Sharing Model

Our setting includes a set of agents denoted by $N = [n]$ and a heterogeneous divisible good (or *cake*) represented by the normalized interval $[0, 1]$. A *piece of cake* is a union of finitely many disjoint (closed) intervals. Denote by $\ell(I)$ the length of an interval I, i.e., $\ell([a, b]) = b - a$. For a piece of cake S, we denote $\ell(S) = \sum_{I \in S} \ell(I)$. Each agent $i \in N$ is endowed with a *density function* $f_i \colon [0, 1] \to \mathbb{R}_{\geq 0}$, which captures how the agent values different parts of the cake. We assume that the agents have *piecewise uniform* utilities (see Definition 2.2) with the density function f_i taking on the value 1 for all desired parts and 0 for all undesired parts. Let $W_i \subseteq [0, 1]$ denote the piece of cake on which $f_i = 1$. The utility of agent i for any piece of cake S is given by $u_i(S) := \ell(S \cap W_i)$. We assume that $\ell(W_i) > 0$ for every agent i, since we can simply ignore an agent i with $\ell(W_i) = 0$.

Let $\alpha \in [0, 1]$ be a given parameter. We refer to a setting with agents, their density functions, and the parameter α as an *instance*. A *mechanism* $\mathcal{M}(R)$ chooses from any given instance R a piece of cake A with $\ell(A) \leq \alpha$. This, however, does not mean that all agents will have full access to A as we allow the mechanism to *block* each agent from accessing certain parts of the selected piece. That is, the mechanism can assign piece $A_i \subseteq A$ to agent i; we call $\mathcal{A} = (A, A_1, \ldots, A_n)$ an *allocation*. The utility of agent i from the allocation $\mathcal{A}$ is $u_i(A_i)$. Since the cases $\alpha = 0$ and $\alpha = 1$ are trivial, (the mechanism cannot allocate any cake and can always allocate the whole cake, respectively), we assume from now on that $\alpha \in (0, 1)$. Given an instance, every point that is a left or right endpoint of an interval in W_i for at least one i is called a *breakpoint*; the points 0 and 1 are also considered to be breakpoints. Observe that for any instance, the agents' utilities for a piece of cake S depend only on the amounts of cake between consecutive pairs of breakpoints included in S.

We now define the central property and the two main mechanisms of this chapter.

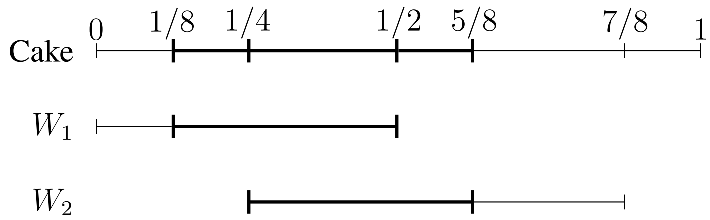
Definition 8.1 (Truthfulness)**.** A mechanism is *truthful* if for any instance R with $\mathcal{M}(R) = (A, A_1, \ldots, A_n)$ and any agent $i \in N$, if the agent reports $W_i' \neq W_i$ and the mechanism returns the allocation $\mathcal{A}' = (A', A_1', \ldots, A_n')$ on the modified instance, then $u_i(A_i) \geq u_i(A_i')$.

Definition 8.2 (MNW)**.** Given an instance, the *maximum Nash welfare (MNW) solution* chooses a piece of cake A with $\ell(A) \leq \alpha$ such that the product $\prod_{i \in N} u_i(A)$ is maximized. It then assigns $A_i = A \cap W_i$ for all $i \in N$.

Definition 8.3 (Leximin)**.** Given an instance, the *leximin solution* considers pieces of cake A with $\ell(A) \leq \alpha$ such that the minimum among the utilities $u_1(A), u_2(A), \ldots, u_n(A)$ is maximized; among all such pieces A, it considers those for which the second smallest utility is maximized, and so on, until after considering the largest utility, it chooses one of the pieces A that remain. It then assigns $A_i = A \cap W_i$ for all $i \in N$.

The following example illustrates some of our definitions.

Example 8.4. Let $\alpha = 1/2$. Consider an instance with two agents whose utility functions are $W_1 = [0, 1/2]$ and $W_2 = [1/4, 7/8]$.



Assume without loss of generality that the tie-breaking rule of both leximin and MNW returns the allocation $A = [1/8, 5/8]$.[3] Then, agent 1 has access to the piece $A_1 = A \cap W_1 = [1/8, 1/2]$ while agent 2 has access to the piece $A_2 = A \cap W_2 = [1/4, 5/8]$. Both agents receive utility $3/8$. $\triangle$

Since both of the mechanisms always choose $A_i = A \cap W_i$ for all i, we can represent an allocation $\mathcal{A}$ simply by the set A when we discuss these mechanisms. Note that

[3]We show in Theorem 8.20 that leximin and MNW are equivalent in the case of two agents.

$u_i(A_i) = \ell(A_i \cap W_i) = \ell(A \cap W_i) = u_i(A)$, so it also suffices to consider the agents' utilities with respect to A. By a standard compactness argument and our observation above that the agents' utilities depend only on the amounts of cake between breakpoints, both solutions are well-defined (i.e., the desired maxima are attained). There may be several maximizing allocations A to choose from, in which case we generally allow arbitrary tie-breaking—as we will see later, this tie-breaking does not influence the utility that each agent receives and therefore does not play a significant role. We call an allocation that is returned by the MNW solution (resp., leximin solution) under some tie-breaking an MNW allocation (resp., leximin allocation). By our assumptions that $\alpha > 0$ and $\ell(W_i) > 0$ for every i, all MNW allocations and leximin allocations give every agent a strictly positive utility.

8.3 Leximin Solution

In this section, we consider the leximin solution and start by establishing its basic properties. Our first result is that the utility of each agent is the same in all leximin allocations, meaning that tie-breaking is not an important issue. The proof proceeds by assuming for contradiction that two leximin allocations give some agent different utilities, and arguing that the "average" of these two allocations would have been a better choice with respect to the leximin ordering.

Proposition 8.5. *Given any instance, for each agent i, the utility that i receives is the same in all leximin allocations.*

Proof. Assume for contradiction that two leximin allocations, A and A', give some agent different utilities. Let A'' be an allocation such that for each pair of consecutive breakpoints, the amount of cake between those breakpoints included in A'' is the average of the corresponding amounts for A and A'. By linearity, A'' is a feasible allocation, and $u_i(A'') = \frac{1}{2}(u_i(A) + u_i(A'))$ for every $i \in N$.

Since the leximin ordering is a total order, the multiset of utilities that the n agents receive in A must be the same as the corresponding multiset in A'. Let j be an agent with the smallest $\min\{u_j(A), u_j(A')\}$ such that $u_j(A) \neq u_j(A')$, and assume w.l.o.g. that $u_j(A) < u_j(A')$. All agents k with $\min\{u_k(A), u_k(A')\} < \min\{u_j(A), u_j(A')\}$ have $u_k(A) = u_k(A')$, and this latter quantity is also equal to $u_k(A'')$. On the other hand, we have $u_j(A'') = \frac{1}{2}(u_j(A) + u_j(A')) > u_j(A)$, which means that the number of agents who receive utility exactly $u_j(A)$ in A'' is strictly less than the corresponding numbers for A and A'. Hence, A'' is a better allocation with respect to the leximin ordering than A and A', a contradiction. □

Next, we show a leximin allocation can be computed efficiently via a linear programming-based approach similar to the one used by Airiau et al. [2] in the context of portioning. Recall that in our setting, agents' utility functions can be described explicitly by the sets W_i.

Proposition 8.6. *There exists an algorithm that computes a leximin allocation in time polynomial in the input size.*

Algorithm 11: Computing a leximin allocation

Input: Agents N, a set M of cake intervals, and $\{W_i\}_{i\in[n]}$.

$N' \leftarrow \emptyset$
$t_i \leftarrow 0, \forall i \in N$
while $N' \neq N$ **do**
  Solve the following linear program:
  maximize t^* subject to
  $\sum_{j=1}^{m} \mathbf{1}_{I_j \subseteq W_i} \cdot x_j \geq t^* \quad \forall i \in N \setminus N'$
  $\sum_{j=1}^{m} \mathbf{1}_{I_j \subseteq W_i} \cdot x_j = t_i \quad \forall i \in N'$
  $\sum_{j=1}^{m} x_j \leq \alpha$
  $0 \leq x_j \leq \ell(I_j) \quad \forall j \in \{1, \dots, m\}$
  Set t^* to be the solution of the linear program.
  for $i' \in N \setminus N'$ **do**
    Solve the following linear program:
    maximize ε subject to
    $\sum_{j=1}^{m} \mathbf{1}_{I_j \subseteq W_{i'}} \cdot x_j \geq t^* + \varepsilon$
    $\sum_{j=1}^{m} \mathbf{1}_{I_j \subseteq W_i} \cdot x_j \geq t^* \quad \forall i \in N \setminus N'$
    $\sum_{j=1}^{m} \mathbf{1}_{I_j \subseteq W_i} \cdot x_j = t_i \quad \forall i \in N'$
    $\sum_{j=1}^{m} x_j \leq \alpha$
    $0 \leq x_j \leq \ell(I_j) \quad \forall j \in \{1, \dots, m\}$
    if $\varepsilon = 0$ **then**
      $N' \leftarrow N' \cup \{i'\}$
      $t_{i'} \leftarrow t^*$
return the solution $\mathbf{x}$ of the last linear program solved

Proof. First, we divide the cake into a set M of intervals $I_1, I_2, \dots, I_m$ using all breakpoints, so each interval is either desired in its entirety or not desired at all by each agent. Let x_j denote the length of I_j that we include in our allocation. Thus, the utility of agent i can be written as $\sum_{j=1}^{m} \mathbf{1}_{I_j \subseteq W_i} x_j$, where $\mathbf{1}_X$ denotes the indicator variable for event X.

We proceed by formulating linear programs. Initially, the set N' of agents whose utility we have already fixed is empty. We determine the smallest utility in a leximin allocation by solving for the maximum t^* such that the utility of every agent is at least t^*. We then determine an agent who receives utility t^* in a leximin allocation—to this end, for each agent, we solve for the maximum ε such that this agent receives utility at least $t^* + \varepsilon$ and every other agent receives utility at least t^*. We choose an agent who returns $\varepsilon = 0$, fix the utility of this agent i' by setting $t_{i'} = t^*$, and continue by finding the next smallest utility among the remaining agents. The pseudocode of our algorithm is given as Algorithm 11.

Since our algorithm requires solving $O(n^2)$ linear programs, it runs in polynomial time. We now establish its correctness. Consider the first iteration of the `while` loop, and the returned value t^* of the first linear program. We claim that for at least one $i' \in N$, the linear program for i' returns $\varepsilon = 0$. Indeed, if this is not the case, then for every i', there is a feasible allocation that gives i' a utility strictly greater than t^* and gives every other agent a utility of at least t^*; by taking the "average" of all such allocations similarly to the proof of

Proposition 8.5, we obtain a feasible allocation that gives every agent strictly greater than t^*, contradicting the definition of t^*. For i' such that $\varepsilon = 0$, we therefore have that the utility of i' is equal to t^* in every leximin allocation. We then apply a similar argument for the remaining $n-1$ iterations to conclude that the utility of each agent in the allocation that the algorithm returns is equal to the corresponding utility in every leximin allocation. It therefore follows that the returned allocation is a leximin allocation. □

We now come to our main result of this section, which establishes the truthfulness of the leximin solution.

Theorem 8.7. *For arbitrary tie-breaking, the leximin solution is truthful.*

At a high level, the proof of Theorem 8.7 proceeds by identifying specific types of manipulations, arguing that such manipulations cannot be beneficial when the leximin solution is used, and then showing that being immune to these manipulations implies being immune to all manipulations. We start by defining an ε-change, a useful concept in our proof.

Definition 8.8 (ε-change)**.** Given two vectors of real numbers $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{x}' = (x_1', x_2', \ldots, x_n')$, an ε-change from $\mathbf{x}$ towards $\mathbf{x}'$ refers to the following continuous operation: for each $i \in \{1, 2, \ldots, n\}$, x_i changes linearly to $x_i'' := x_i + \varepsilon(x_i' - x_i)$, where ε is sufficiently small so that if $x_i < x_j$, then $x_i'' < x_j''$.

For ease of expression, we will also use an ε-change to refer to the outcome of such an operation, i.e., the vector $\mathbf{x}''$. When we discuss ε-changes, we will not specify the exact value of ε: any ε satisfying the above condition works.

Lemma 8.9. *Given two vectors* $\mathbf{x}$ *and* $\mathbf{y}$*, if* $\mathbf{y}$ *is a better vector with respect to the leximin ordering than* $\mathbf{x}$*, then an* ε*-change from* $\mathbf{x}$ *to* $\mathbf{y}$ *is also a leximin improvement.*

Proof. Sort the numbers of $\mathbf{x}$ in non-descending order and group them into buckets so that numbers within each bucket are the same and those in different buckets are different. Observe that an ε-change improves $\mathbf{x}$ with respect to the leximin ordering if and only if for the lowest bucket where there is a change, some number increases and no number decreases.

Consider an ε-change from $\mathbf{x}$ towards a better leximin vector $\mathbf{y}$. If some number in the lowest bucket of $\mathbf{x}$ decreases, then $\mathbf{y}$ would not be a leximin improvement of $\mathbf{x}$, so no number in this bucket decreases. If some number in this bucket increases, we are done by the above observation. Else, there is no change in this bucket, and we move on to the next bucket and repeat the same argument. Because $\mathbf{x}$ and $\mathbf{y}$ are different, there must be a change in at least one bucket, which gives our desired conclusion. □

We now extend the definition of an ε-change to allocations. Recall that for the leximin solution, it suffices to consider the set A instead of the entire allocation $\mathcal{A}$. Given two allocations A and A', an ε-change from A towards A' can be captured by dividing the cake into intervals according to the breakpoints and changing A towards A' so that the length of cake included in the allocation in each interval changes linearly. Note that when we perform an

ε-change from A towards A', by linearity, we also obtain a corresponding ε-change from the vector $(u_1(A), u_2(A), \ldots, u_n(A))$ towards $(u_1(A'), u_2(A'), \ldots, u_n(A'))$, and any allocation obtained during the process is feasible.

Next, we present auxiliary lemmas used for proving the truthfulness of leximin solution. These lemmas discuss how the leximin allocation can change when an agent modifies her density function in various ways. For notational convenience, in these lemmas we assume that instance R (resp., R') contains the density functions corresponding to $W_1, W_2, \ldots, W_n$ (resp., $W'_1, W'_2, \ldots, W'_n$). Our first lemma says that whenever an agent shrinks her desired piece in such a way that it contains the entire portion she receives, then she should still receive the same portion in the new instance.

Lemma 8.10. *Given a leximin allocation A for instance R, let R' be an instance such that $A \cap W_i \subseteq W'_i \subseteq W_i$ for an agent $i \in N$ and $W'_j = W_j$ for all $j \in N \setminus \{i\}$. Then, A is also a leximin allocation for R'.*

Proof. Suppose for contradiction that A is not a leximin allocation for R'. Consider a leximin allocation A' for R'. Since $W'_i \subseteq W_i$, the utility of agent i for A' in R' is at most that for A' in R. Moreover, the utility of every agent $j \neq i$ for A' is the same in R' as in R. On the other hand, since $A \cap W_i \subseteq W'_i$, the utility of every agent for A is the same in R' as in R. By our assumption that A' is a better allocation with respect to the leximin ordering than A in R', the same must also hold for R, a contradiction. □

Our second lemma says that whenever an agent shrinks her desired piece, she should not get a higher utility than before.

Lemma 8.11. *Given a leximin allocation A for instance R, let R' be an instance such that $W'_i \subseteq W_i$ for an agent $i \in N$ and $W'_j = W_j$ for all $j \in N \setminus \{i\}$. Let A' be a leximin allocation for R'. Then, $\ell(A' \cap W'_i) \leq \ell(A \cap W_i)$.*

Proof. Suppose for contradiction that $\ell(A' \cap W'_i) > \ell(A \cap W_i)$; let x' and x denote the former and latter quantities, respectively. By Proposition 8.5, agent i receives the same utility in all leximin allocations, so A is a better allocation with respect to the leximin ordering than A' in R. When changing from A' to A, since $W'_i \subseteq W_i$, the utility of agent i decreases from at least x' to x with respect to R. Hence, even if the utility of agent i started at exactly x' and decreased to x, the change would still be a leximin improvement.

Now, consider the agents' utilities with respect to R'. When changing from A' to A, since $W'_i \subseteq W_i$, the utility of agent i decreases from x' to at most x. Since all changes are in the same direction as the previous change starting from x', which is a leximin improvement, by Lemma 8.9, an ε-change from A' towards A is also a leximin improvement with respect to R'. This contradicts the assumption that A' is a leximin allocation for R'. □

Our third lemma says that if an agent is already getting her entire desired piece, then whenever she shrinks her desired piece, she should still be at maximum happiness.

Lemma 8.12. *Given a leximin allocation A for instance R with $W_i \subseteq A$ for an agent $i \in N$, let R' be an instance such that $W_i' \subseteq W_i$ and $W_j' = W_j$ for all $j \in N \setminus \{i\}$. Let A' be a leximin allocation for R'. Then, $W_i' \subseteq A'$.*

Proof. Suppose for contradiction that A' does not contain the entire W_i', and let $x' = \ell(A' \cap W_i') < \ell(W_i')$. By Proposition 8.5, agent i receives the same utility in all leximin allocations, so A is a better allocation with respect to the leximin ordering than A' in R. By Lemma 8.9, an ε-change from A' towards A is also a leximin improvement with respect to R, so by the characterization of ε-change improvements in the proof of the lemma, in the lowest bucket where there is a change, some number increases and no number decreases. In this ε-change, the utility of agent i increases from at least x' towards $\ell(W_i)$.

Now, consider the same ε-change from A' towards A, but with respect to R'. The utility of agent i increases from exactly x' towards $\ell(W_i')$, while those of other agents change in the same way as before. Since agent i's utility starts no higher than before and still increases, one can check that in the lowest bucket where there is a change, again some number increases and no number decreases. Hence, the characterization of ε-change improvements implies that the change is also a leximin improvement with respect to R'. This contradicts the assumption that A' is a leximin allocation for R'. $\square$

We are now ready to prove Theorem 8.7.

Proof of Theorem 8.7. Suppose for contradiction that the leximin solution is not truthful. This means that there exists an instance R with leximin allocation A such that if agent i reports $\widehat{W}_i$ instead of W_i, a leximin allocation $\widehat{A}$ in the new instance $\widehat{R}$ satisfies $\ell(\widehat{A} \cap \widehat{W}_i \cap W_i) > \ell(A \cap W_i)$. We will keep the desired pieces W_j of agents $j \in N \setminus \{i\}$ unchanged throughout this proof.

First, consider an instance $\widehat{R}'$ where $\widehat{W}_i' = \widehat{W}_i \cap \widehat{A}$. By Lemma 8.10 applied to $\widehat{R}$ and $\widehat{R}'$, $\widehat{A}$ is also a leximin allocation for $\widehat{R}'$. Next, consider an instance $\widehat{R}''$ in which $\widehat{W}_i'' = \widehat{W}_i \cap \widehat{A} \cap W_i$. Since $\widehat{W}_i' \subseteq \widehat{A}$, by Lemma 8.12 applied to $\widehat{R}'$ and $\widehat{R}''$, any leximin allocation for $\widehat{R}''$ must contain the entire $\widehat{W}_i''$. Recall that $\ell(\widehat{W}_i'') > \ell(A \cap W_i)$.

Finally, consider the instances R and $\widehat{R}''$. From the former to the latter, agent i's desired piece shrinks from W_i to $\widehat{W}_i'' \subseteq W_i$. By Lemma 8.11, the agent should not get a higher utility through this shrinking. However, the agent's utility is $\ell(A \cap W_i)$ before the shrinking, and $\ell(\widehat{W}_i'')$ afterwards. This is a contradiction. $\square$

Observe that unlike MNW, the leximin solution depends on the normalization of the agents' utilities. Besides our normalization, another common choice in cake cutting is to normalize the utility of every agent for the whole cake to 1. We remark here that with this alternative normalization, the leximin solution is *not* truthful. To see this, consider two agents with $W_1 = [0, 1/3]$ and $W_2 = [1/3, 2/3]$, and let $\alpha = 1/3$. In this instance, the (alternative) leximin solution gives each agent length $1/6$ of the cake. However, if agent 2 misreports that $W_2 = [1/3, 1]$, then it is possible that the agent receives the interval $[1/3, 5/9]$ and therefore length $2/9 > 1/6$ of her valued cake. In the next section, we provide further evidence that our normalization is the appropriate one in cake sharing by showing that our version of the leximin solution achieves a strong fairness guarantee in terms of egalitarian welfare.

8.4 Egalitarian Ratio

As we mentioned in the introduction, truthfulness by itself is easy to achieve, for example by always allocating a fixed piece of cake of length α. However, this may leave certain agents with zero utility, a patently unfair outcome. A common measure of fairness is the *egalitarian welfare*, which corresponds to the minimum among the utilities of all agents. In order to perform meaningful interpersonal comparisons of utilities, we normalize the utilities in the following definition.

Definition 8.13 (Egalitarian ratio)**.** Given an instance R and an allocation $\mathcal{A}$, the *egalitarian ratio* of $\mathcal{A}$ is defined as

$$\textit{Egal-ratio-alloc}_R(\mathcal{A}) = \min_{i \in N} \frac{u_i(A_i)}{u_i([0,1])}.$$

For a mechanism $\mathcal{M}$ and parameters n and α, the *egalitarian ratio* of $\mathcal{M}$ with respect to n and α is defined as

$$\textit{Egal-ratio}_{n,\alpha}(\mathcal{M}) = \inf_R \textit{Egal-ratio-alloc}_R(\mathcal{M}(R)),$$

where the infimum is taken over all instances with n agents and parameter α.

In other words, the egalitarian ratio of $\mathcal{M}$ with respect to n and α is the smallest ratio between an agent's utility for her piece allocated by $\mathcal{M}$ and her utility for the entire cake, taken over all instances with parameters n and α. For example, if a mechanism always allocates a fixed piece of length α regardless of the agents' utility functions, then its egalitarian ratio with respect to any n and $\alpha \in (0,1)$ is 0. We first present a tight upper bound on the egalitarian ratio.

Proposition 8.14. *For all $n \geq 1$ and $\alpha \in (0,1)$,*

$$0 \leq \text{Egal-ratio}_{n,\alpha}(\mathcal{M}) \leq \alpha$$

for any mechanism $\mathcal{M}$. Moreover, for each inequality, there exists a mechanism $\mathcal{M}$ such that the inequality is tight.

Proof. The lower bound of 0 holds trivially, and is achieved by the mechanism discussed before the proposition.

For the upper bound, note that if some agent i values the whole cake (i.e., $W_i = [0,1]$), then $u_i([0,1]) = 1$ and $u_i(A_i) \leq \alpha$, so no mechanism can achieve egalitarian ratio larger than α. The tightness follows from a mechanism that, given any instance, divides the cake into intervals using all breakpoints and chooses an (arbitrary) α fraction from each interval—this results in $u_i(A_i) = \alpha \cdot u_i([0,1])$ for all i. □

Our next result gives the precise egalitarian ratio of the leximin solution.

Theorem 8.15. *For all $n \geq 1$ and $\alpha \in (0,1)$,*

$$\text{Egal-ratio}_{n,\alpha}(\text{leximin}) = \frac{\alpha}{n-(n-1)\alpha}.$$

Proof. For the upper bound, consider instance R with $W_i = \left[\frac{(i-1)\alpha}{n}, \frac{i\alpha}{n}\right]$ for $i = 1, \ldots, n-1$, and $W_n = \left[\frac{(n-1)\alpha}{n}, 1\right]$. Any leximin allocation A gives a cake of length $\frac{\alpha}{n}$ to every agent, so

$$\text{Egal-ratio-alloc}_R(A) \leq \frac{u_n(A)}{u_n([0,1])} = \frac{\alpha/n}{1-(n-1)\alpha/n} = \frac{\alpha}{n-(n-1)\alpha}.$$

We now consider the lower bound. Since the mechanism can allocate length α of the cake and there are n agents, it can give every agent i a utility of at least $\min\{\alpha/n, \ell(W_i)\}$. Hence, all leximin allocations give each agent i at least this much utility. If an agent has $\ell(W_i) \leq 1 - \frac{(n-1)\alpha}{n}$, the utility ratio for this agent is at least $\frac{\alpha/n}{1-(n-1)\alpha/n} = \frac{\alpha}{n-(n-1)\alpha}$. Else, suppose that $\ell(W_i) = 1 - \frac{(n-1)\alpha}{n} + x$ for some $x > 0$. In this case, no matter how the mechanism allocates length α of the cake, the utility of this agent is at least

$$\alpha - \left(1 - \left(1 - \frac{(n-1)\alpha}{n} + x\right)\right) = \alpha/n + x.$$

Hence, the utility ratio of this agent is at least

$$\frac{\alpha/n + x}{1-(n-1)\alpha/n + x} \geq \frac{\alpha}{n-(n-1)\alpha},$$

where the inequality follows from the fact that the expression on the left-hand side is non-decreasing for $x \in [0, \infty)$. □

Theorem 8.15 shows that the leximin solution achieves a non-trivial egalitarian ratio. However, it is unclear how good this ratio is. We will therefore show that the solution attains the highest possible ratio among all truthful mechanisms satisfying a natural condition. Given a vector of piecewise uniform density functions $\mathbf{f} = (f_1, f_2, \ldots, f_n)$, let $L_{\mathbf{f}}$ be a vector with 2^n components such that each component represents a distinct subset of agents and the value of the component is the length of the piece desired by exactly that subset of agents (and not by any agent outside the subset).

Example 8.16. Consider the instance in Example 8.4. The corresponding $L_{\mathbf{f}}$ of this instance is $(1/8, 1/4, 3/8, 1/4)$, where the components correspond to the lengths of the pieces desired by exactly the set of agents $\emptyset$, $\{1\}$, $\{2\}$, and $\{1,2\}$, respectively. △

Definition 8.17 (Position obliviousness)**.** A mechanism $\mathcal{M}$ is *position oblivious* if the following holds: Let $\mathbf{f}$ and $\mathbf{f}'$ be any vectors of density functions such that $L_{\mathbf{f}} = L_{\mathbf{f}'}$, and let R and R' be instances represented by these respective vectors and a given α; if $\mathcal{M}(R) = (A, A_1, \ldots, A_n)$ and $\mathcal{M}(R') = (A', A'_1, \ldots, A'_n)$, then $u_i(A_i) = u'_i(A'_i)$ for every $i \in N$.

Position obliviousness has previously been studied by Bei et al. [34]. Intuitively, for a position oblivious mechanism, the utility of an agent depends only on the lengths of the pieces desired by various subsets of agents and not on the positions of these pieces. It follows directly from the definition that the leximin solution is position oblivious.[4]

Theorem 8.18. *Let $\mathcal{M}$ be a truthful and position oblivious mechanism. Then, for all $n \geq 1$ and $\alpha \in (0,1)$,*

$$\text{Egal-ratio}_{n,\alpha}(\mathcal{M}) \leq \frac{\alpha}{n-(n-1)\alpha}.$$

Proof. Assume for the sake of contradiction that there exists a truthful and position oblivious mechanism $\mathcal{M}$ with *Egal-ratio*$_{n,\alpha}(\mathcal{M}) = \frac{\alpha}{n-(n-1)\alpha} + \delta$ for some $\delta > 0$. For each $i \in N$, let C_i be a piece of length $\ell(C_i) = \alpha/n + \varepsilon$ such that $C_i \cap C_j = \emptyset$ for every pair $i, j \in N$, where $\varepsilon > 0$ is such that

$$\varepsilon < \min\left\{\frac{1-\alpha}{n}, \frac{\delta(n-(n-1)\alpha)^2}{n(n-1)(\alpha+\delta(n-(n-1)\alpha))}\right\}.$$

Consider an instance R where $W_i = C_i$ for all $i \in N$. Since $\mathcal{M}$ can allocate length at most α of the cake, it must return an allocation for which some agent receives utility at most α/n. Assume without loss of generality that $\mathcal{M}$ returns an allocation $\mathcal{A}$ with $u_1(A_1) \leq \alpha/n$.

Next, consider an instance R' where $W_i = C_i$ for all $i \in N \setminus \{1\}$ and $W_1 = [0,1] \setminus \bigcup_{i \in N\setminus\{1\}} C_i$. For this instance, we have $\ell(W_1) = 1-(n-1)\cdot(\alpha/n+\varepsilon)$. Let $\mathcal{A}' = \mathcal{M}(R')$, and let $Y = A_1' \cap W_1$. By the definition of egalitarian ratio, we have $u_1(A_1')/u_1([0,1]) \geq$ Egal-ratio$_{n,\alpha}(\mathcal{M})$, that is,

$$\begin{aligned}
\ell(Y) &\geq \text{Egal-ratio}_{n,\alpha}(\mathcal{M}) \cdot \ell(W_1) \\
&= \left(\frac{\alpha}{n-(n-1)\alpha} + \delta\right) \cdot \left(1-(n-1)\cdot\left(\frac{\alpha}{n}+\varepsilon\right)\right) \\
&= \left(\frac{\alpha}{n-(n-1)\alpha} + \delta\right) \cdot \left(\frac{n-(n-1)\alpha}{n} - (n-1)\varepsilon\right) \\
&= \frac{\alpha}{n} + \frac{\delta(n-(n-1)\alpha)}{n} - \left(\frac{(n-1)(\alpha+\delta(n-(n-1)\alpha))}{n-(n-1)\alpha}\right)\varepsilon
\end{aligned}$$

which is greater than α/n by our choice of ε.

Finally, consider an instance R'' where $W_i = C_i$ for all $i \in N\setminus\{1\}$, while W_1 is a subset of $[0,1] \setminus \bigcup_{i \in N\setminus\{1\}} C_i$ of length $\ell(W_1) = \alpha/n + \varepsilon$ such that $\ell(W_1 \cap Y) > \alpha/n$. Since $\mathcal{M}$ is position oblivious, by comparing instances R'' to R, agent 1 must also get a utility of at most α/n in R''. However, if the agent reports $[0,1] \setminus \bigcup_{i \in N\setminus\{1\}} C_i$ as in R', she gets utility $\ell(W_1 \cap Y) > \alpha/n$, meaning that $\mathcal{M}$ is not truthful and yields the desired contradiction. □

Comparing this ratio with highest possible ratio of α without the truthfulness condition (Proposition 8.14),[5] one can see that adding the truthfulness requirement incurs a (multi-

[4] Bei et al. [32] considered a slightly stronger version of position obliviousness, which the leximin solution also satisfies.

[5] Note that the mechanism that achieves egalitarian ratio α in Proposition 8.14 satisfies position obliviousness.

plicative) "price" of $n-(n-1)\alpha$ on the best egalitarian ratio. This price can be as large as n when α is close to 0, and decreases to 1 as α approaches 1.

8.5 Maximum Nash Welfare

In this section, we address the MNW solution. We start by showing that like the leximin solution (Proposition 8.5), the utility that each agent receives is the same in all MNW allocations, thereby rendering the tie-breaking issue insignificant.

Proposition 8.19. *Given any instance, for each agent i, the utility that i receives is the same in all MNW allocations.*

Proof. We proceed in a similar manner as in the proof of Proposition 8.5. Assume for contradiction that two MNW allocations A, A' give some agent different utilities. Since the utility of every agent in an MNW allocation is strictly positive, we have $u_i(A), u_i(A') > 0$ for all $i \in N$. Let A'' be an allocation such that for each pair of consecutive breakpoints, the amount of cake between those breakpoints included in A'' is the average of the corresponding amounts for A and A'. By linearity, A'' is a feasible allocation, and $u_i(A'') = \frac{u_i(A)+u_i(A')}{2}$ for every $i \in N$. Recall that by the AM-GM inequality, it holds that $\frac{x+y}{2} \geq \sqrt{xy}$ for all positive real numbers x, y, with equality if and only if $x = y$. We therefore have

$$\prod_{i\in N} u_i(A'') = \prod_{i\in N}\left(\frac{u_i(A)+u_i(A')}{2}\right) > \prod_{i\in N}\sqrt{u_i(A)\cdot u_i(A')} = \sqrt{\prod_{i\in N} u_i(A)}\cdot\sqrt{\prod_{i\in N} u_i(A')},$$

where the inequality is strict because $u_i(A) \neq u_i(A')$ for at least one i. Since $\prod_{i\in N} u_i(A) = \prod_{i\in N} u_i(A')$, this implies that A'' has a higher Nash welfare than both A and A', yielding the desired contradiction. □

In the two-agent case, we show that MNW and leximin are equivalent. The high-level idea is that both solutions can be obtained via the following process. First, select portions of the cake desired by both agents. If the quota α has not been reached yet, let the agents "eat" their desired piece using the same speed, until either (i) one of the agents has no more desired cake, in which case we let the other agent continue eating, or (ii) we run out of quota.

Theorem 8.20. *Consider an instance with two agents. Any leximin allocation is an MNW allocation, and vice versa.*

Proof. Fix an instance with two agents, and let $X = W_1 \cap W_2$ and $x = \ell(X)$. If $x \geq \alpha$, then an allocation A is leximin if and only if $A \subseteq X$, and the same holds for MNW. Similarly, if $\ell(W_1 \cup W_2) \leq \alpha$, the relevant condition for both leximin and MNW is $W_1 \cup W_2 \subseteq A$.

Assume now that $x < \alpha < \ell(W_1 \cup W_2)$. Since both the leximin and MNW solutions satisfy Pareto optimality, we must have $\ell(A) = \alpha$ and $X \subseteq A$ in any leximin or MNW allocation A. In other words, the entire intersection of length x must be allocated, along with a further length $\alpha - x$ of the cake. Let $\Delta_1 = W_1 \setminus W_2$ and $\Delta_2 = W_2 \setminus W_1$, and consider two cases. The desired conclusion will follow from the analyses.

Case 1: $\min\{\ell(\Delta_1), \ell(\Delta_2)\} \geq (\alpha - x)/2$. In this case, for both leximin and MNW, the length $\alpha - x$ must be split equally between Δ_1 and Δ_2, otherwise the allocation can be improved with respect to both the leximin ordering and the Nash welfare by splitting the length equally. Conversely, any allocation that splits the length $\alpha - x$ equally between Δ_1 and Δ_2 is both leximin and MNW.

Case 2: $\min\{\ell(\Delta_1), \ell(\Delta_2)\} < (\alpha - x)/2$. Assume without loss of generality that $\ell(\Delta_1) < (\alpha - x)/2$. Since $\ell(\Delta_1) + \ell(\Delta_2) = \ell(W_1 \cup W_2) - x > \alpha - x$, we have $\ell(\Delta_2) > (\alpha - x)/2$. In this case, the entire Δ_1 must be allocated, otherwise the allocation A can be improved with respect to both the leximin ordering and the Nash welfare by allocating ε more of Δ_1 and ε less of Δ_2, for any $0 < \varepsilon < \ell(\Delta_1 \setminus A)$. Conversely, any allocation that allocates the entire Δ_1 and length $\alpha - x - \ell(\Delta_1)$ of Δ_2 is both leximin and MNW. □

Theorems 8.7 and 8.20 together imply the following:

Corollary 8.21. *For two agents and arbitrary tie-breaking, the MNW solution is truthful.*

When $n \geq 3$, the two mechanisms are no longer equivalent. This can be seen from the instance with $W_1 = [0, 1/2]$ and $W_i = [1/2, 1]$ for all $2 \leq i \leq n$, and $\alpha = 1/2$. The leximin solution selects length $1/4$ from each half of the cake, while MNW selects length $\frac{1}{2n}$ from the first half and $\frac{n-1}{2n}$ from the second half. For our main result of this section, we demonstrate that the MNW solution is not truthful even when an agent is only allowed to report a subset of her true desired piece—as discussed in Section 8.1, this strengthens the non-truthfulness result of Aziz et al. [24] where the manipulation is not of this simple nature. In particular, we construct an instance with six agents wherein one of the agents can obtain a higher utility by reporting a subset of her actual desired piece.

Theorem 8.22. *The MNW solution is not truthful regardless of tie-breaking.*

Proof. Assume for convenience that the cake is represented by the interval $[0, 8]$; this can be trivially scaled back down to $[0, 1]$. In our original instance, there are six agents whose utility functions are given as follows:

$$\begin{aligned} W_1 &= [0,1] \cup [2,8], \quad & W_2 &= [0,1] \cup [2,5], \\ W_3 &= [0,1] \cup [5,8], \quad & W_4 &= [1,3] \cup [5,6], \\ W_5 = [1,2] \cup [3,4] &\cup [6,7], \quad & W_6 &= [1,2] \cup [4,5] \cup [7,8], \end{aligned}$$

and let $\alpha = 2$. See Figure 8.1.

First, observe that in this instance, every (non-integer) point is valued by exactly three agents. Hence, for any subset A of the cake with $\ell(A) \leq 2$, we have $\sum_{i=1}^{6} u_i(A) \leq 6$. By the inequality of arithmetic and geometric means (AM-GM), it holds that $\prod_{i=1}^{6} u_i(A) \leq 1$. Moreover, by choosing $A = [0, 2]$, we obtain $u_i(A) = 1$ for each i, so this choice of A maximizes the Nash welfare as $\ell(A) = 2$ and $\prod_{i=1}^{6} u_i(A_i) = 1$, and gives agent 1 a utility of 1. By Proposition 8.19, agent 1 receives a utility of 1 in every MNW allocation.

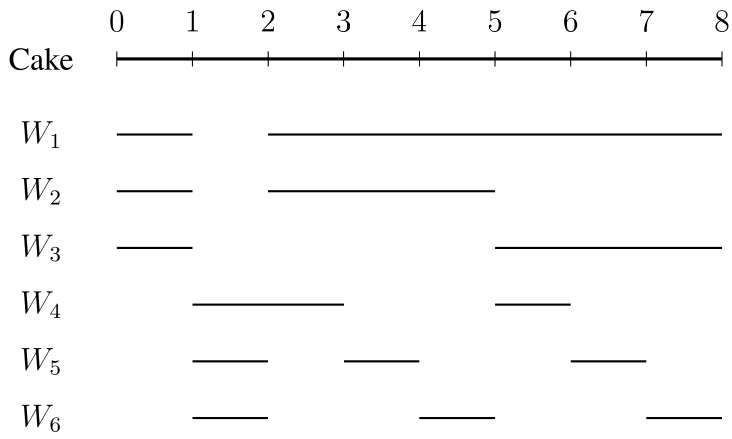


Figure 8.1: The original instance in the proof of Theorem 8.22.

Next, consider a modified instance where agent 1 reports $W_1 = [2, 8]$. Consider an MNW allocation A for this instance, and let $x := \ell(A \cap [2, 8])$, $y := \ell(A \cap [1, 2])$, and $z := \ell(A \cap [0, 1])$, so $x + y + z \leq 2$. Let A' be an allocation such that $\ell(A' \cap [0, 1]) = 2 - x - y \geq z$, $\ell(A' \cap [1, 2]) = y$, and $|A' \cap [j, j + 1]| = x/6$ for $j \in \{2, 3, \ldots, 7\}$. Notice that $\ell(A') = 2$, so A' is a feasible allocation. We claim that $\prod_{i=1}^{6} u_i(A') \geq \prod_{i=1}^{6} u_i(A)$. Indeed, letting $\tau := \ell(A \cap [2, 5])$, we have

$$u_2(A) \cdot u_3(A) = (z + \tau)(z + (x - \tau)) \leq \left(z + \frac{x}{2}\right)^2 \leq u_2(A') \cdot u_3(A'),$$

where the first inequality follows from the AM-GM inequality. Similarly, letting $\theta := \ell(A \cap ([2, 3] \cup [5, 6]))$ and $\rho := \ell(A \cap ([3, 4] \cup [6, 7]))$, it holds that

$$u_4(A)u_5(A)u_6(A) = (y + \theta)(y + \rho)(y + (x - \theta - \rho)) \leq \left(y + \frac{x}{3}\right)^3 = u_4(A')u_5(A')u_6(A').$$

Moreover, since $u_1(A) = u_1(A') = x$, it follows that $\prod_{i=1}^{6} u_i(A') \geq \prod_{i=1}^{6} u_i(A)$, as claimed. This means that A' is also an MNW allocation. The Nash welfare of A' is

$$\prod_{i=1}^{6} u_i(A') = x\left(2 - \frac{x}{2} - y\right)^2 \left(y + \frac{x}{3}\right)^3.$$

In order to show MNW is not truthful regardless of tie-breaking, by Proposition 8.19, it suffices to show that the maximum of this expression in the domain $x, y \geq 0, x + y \in [0, 2]$ is attained when $x > 1$, since this would imply that agent 1 has a profitable deviation.

Let $g(x, y) := x\left(2 - \frac{x}{2} - y\right)^2 \left(y + \frac{x}{3}\right)^3$, where $x, y \geq 0$ and $x + y \leq 2$. We have $g(1.5, 0.5) = 0.84375$. Now, from the AM-GM inequality,

$$\begin{aligned}
\frac{9}{4} \cdot g(x, y) &= x\left(3 - \frac{3x}{4} - \frac{3y}{2}\right)^2 \left(y + \frac{x}{3}\right)^3 \\
&\leq x\left(\frac{2(3 - 3x/4 - 3y/2) + 3(y + x/3)}{5}\right)^5 \\
&= x\left(\frac{6 - x/2}{5}\right)^5.
\end{aligned}$$

The derivative of the last expression is $\left(\frac{6-3x}{5}\right)\left(\frac{6-x/2}{5}\right)^4$, which is non-negative for $0 \le x \le 2$. This means that for $x \le 1$, we have

$$\frac{9}{4} \cdot g(x,y) \le 1 \cdot \left(\frac{6-1/2}{5}\right)^5 = \left(\frac{11}{10}\right)^5,$$

so $g(x,y) \le \frac{4}{9} \cdot (1.1)^5 < 0.72 < g(1.5, 0.5)$. It follows that the maximum of $g(x,y)$ is attained when $x > 1$, as desired. $\square$

We remark here that even if we allow the MNW solution to choose any A_i such that $A \cap W_i \subseteq A_i \subseteq A$ instead of always choosing $A_i = A \cap W_i$ (that is, the mechanism may give agent i some parts of A that she does not value, along with all parts of A that she values), the example in Theorem 8.22 still shows that any resulting mechanism is not truthful.

8.6 Without Blocking: Impossibility Result

As we have so far assumed that mechanisms can block agents from accessing certain parts of the resource, an important question is what guarantees the mechanisms can achieve without the ability to block. Indeed, while blocking can be easily implemented in our introductory applications by restricting access to the sports facility or files in a cache memory, it may be harder or more costly in other situations. In this section, we discuss mechanisms without the blocking ability. When no blocking is allowed, given an input instance, a mechanism $\mathcal{M}$ chooses a piece of cake A with $\ell(A) \le \alpha$, and each agent i receives a utility of $u_i(A) = \ell(A \cap W_i)$.

First, we observe that while the leximin solution is truthful if it has the ability to block (Theorem 8.7), this is no longer the case in the absence of blocking.

Example 8.23 (Leximin is not truthful without blocking)**.** Let $\alpha = 1/2$. First, consider an instance R with two agents whose utility functions are given as follows: $W_1 = [0, 1/2]$ and $W_2 = [1/2, 1]$. Assume without loss of generality that the tie-breaking rule chooses $A = [1/4, 3/4]$. Next, consider an instance R' with the following utility functions: $W_1 = [0, 3/4]$ and $W_2 = [1/2, 1]$. Agent 1 receives a utility of $3/8$ in every leximin allocation for R'. However, if agent 1 misreports that $W_1 = [0, 1/2]$, the instance becomes the same as R, and agent 1 receives a utility of $1/2$ from the allocation A. $\triangle$

Our main result of this section shows that Example 8.23 is in fact not a coincidence.

Theorem 8.24. *Without blocking, for every $\alpha \in (0, 1)$, no truthful, Pareto optimal, and position oblivious mechanism can achieve a positive egalitarian ratio even in the case of two agents.*

Proof. We assume for contradiction that there exists some $\alpha \in (0, 1)$ and a truthful, Pareto optimal, and position oblivious mechanism $\mathcal{M}$ with $\text{Egal-ratio}_{2,\alpha}(\mathcal{M}) > 0$. We consider a sequence of instances with two agents, which we illustrate in Figure 8.2. In the following,

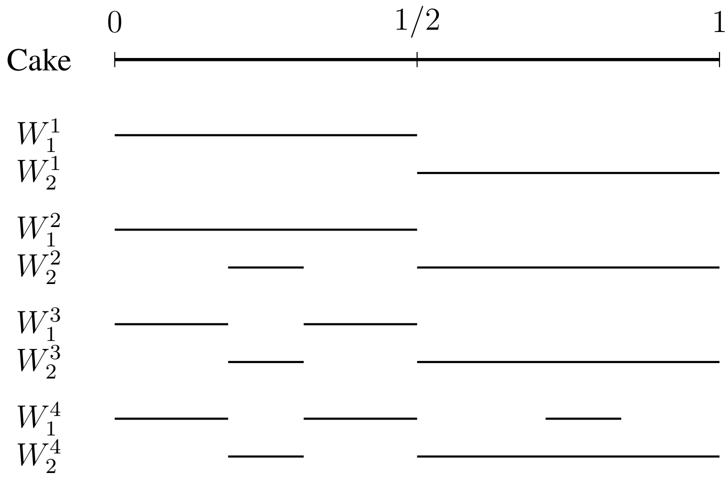


Figure 8.2: Example instances in the proof of Theorem 8.24.

the superscripts denote the indices of the instances. In all of the instances that we consider, every part of the cake is desired by at least one agent, so Pareto optimality implies that $\mathcal{M}$ must allocate exactly α of the cake.

Instance R^1**:** $W_1^1 = [0, 0.5], W_2^1 = [0.5, 1]$. Let $\mathcal{M}(R^1) = A^1$. Because $\alpha < 1$, at least one of the agents will not obtain her maximum utility of 0.5. Assume without loss of generality that $\ell(W_1^1 \cap A^1) = x < 0.5$; in other words, $W_1^1 \setminus A^1$ is non-empty. Since $\mathcal{M}$ has a positive egalitarian ratio, it must hold that $0 < x < \alpha$.

Instance R^2**:** $W_1^2 = [0, 0.5], W_2^2 = A^1 \cup [0.5, 1]$. Let $\mathcal{M}(R^2) = A^2$. We must have $A^2 \subseteq W_2^2$; in other words, agent 2 will receive utility α. This is because otherwise, agent 2 can benefit by reporting $W_2^{2\prime} = [0.5, 1]$ and the instance becomes R^1, in which case agent 2 will receive utility α from the output allocation A^1. Note that because A^2 is contained entirely in W_2^2, we still have $\ell(W_1^2 \cap A^2) \leq x$.

Instance R^3**:** $W_1^3 = [0, 0.5] \setminus A^1, W_2^3 = A^1 \cup [0.5, 1]$. Let $\mathcal{M}(R^3) = A^3$. By the positive egalitarian ratio, we have $\ell(W_1^3 \cap A^3) = y > 0$.

Instance R^4**:** $W_1^4 = ([0, 0.5] \setminus A^1) \cup B, W_2^4 = A^1 \cup [0.5, 1]$, where B is an interval of length x contained in W_2^3 with the largest intersection with A^3. That is,

- if $\ell(W_2^4 \cap A^3) \geq x$, let B be any subset of $W_2^4 \cap A^3$ of length x;
- if $\ell(W_2^4 \cap A^3) < x$, let B be any interval of length x that contains $W_2^4 \cap A^3$.

Let $\mathcal{M}(R^4) = A^4$. In this instance, we must have $u_1(A^4) > x$ because otherwise, agent 1 can benefit by reporting $W_1^{4\prime} = [0, 0.5] \setminus A^1$ and the instance becomes R^3, in which case agent 1 will obtain a utility of $x + y$ (when $\ell(W_2^4 \cap A^3) \geq x$) or a utility of α (when $\ell(W_2^4 \cap A^3) < x$). In both cases this value is strictly larger than x.

Finally, observe that instances R^2 and R^4 have the same $L_{\mathbf{f}}$ vector. In particular, we have $\ell(W_1^2) = \ell(W_1^4) = 1/2$, $\ell(W_2^2) = \ell(W_2^4) = 1/2 + x$, and $\ell(W_1^2 \cap W_2^2) = \ell(W_1^4 \cap W_2^4) = x$.

This means that each agent should receive the same utility in these two instances from our position oblivious mechanism $\mathcal{M}$. However, agent 1 receives utility at most x in R_2 and utility strictly larger than x in R^4. We have reached a contradiction. □

8.7 Non-Uniform Costs

In this section, we consider an extension of our model where the cost of selecting the cake may be non-uniform. Specifically, there is a (public) *cost function* $c\colon [0,1] \to \mathbb{R}_{\geq 0}$, which captures the cost for different parts of the cake. We assume that the cost function is piecewise constant, and without loss of generality that $\int_0^1 c \,\mathrm{d}x = 1$ (if the whole cake has cost 0, the mechanism can simply always choose the whole cake). Note that the main model of this chapter corresponds to the cost function being the constant 1 over the entire cake. We still consider piecewise uniform utility functions of the agents, and allow the mechanism to choose a piece of cake with cost at most a given parameter $\alpha \in (0,1)$.

We can generalize the leximin solution to this setting as follows. First, consider all breakpoints of the cost function, where the breakpoints are defined in the same way as for the utility functions. Then, for each piece of cake between two consecutive breakpoints, we choose a fraction of at most α of this cake by implementing the canonical leximin solution with the same α. The generalized leximin solution then returns the union of the chosen cake. By linearity, the chosen cake has cost at most α.

Theorem 8.25. *For all $n \geq 1$ and $\alpha \in (0,1)$, when the cost function is piecewise constant, the generalized leximin solution is truthful and has egalitarian ratio $\frac{\alpha}{n-(n-1)\alpha}$.*

Proof. We first establish truthfulness. The cost function is public and its breakpoints cannot be controlled by the agents, so we can consider the piece of cake between each pair of consecutive breakpoints separately. By Theorem 8.7, for each piece, reporting the utility function truthfully yields the highest utility to each agent. Since the utility for the whole cake is simply the sum of the utilities for different pieces, the mechanism is truthful.

The upper bound of the egalitarian ratio follows from Theorem 8.15 since the cost function in the current theorem is more general. For the lower bound, observe that by Theorem 8.15, for the piece of cake between each pair of consecutive breakpoints, each agent receives a utility of at least a fraction $\frac{\alpha}{n-(n-1)\alpha}$ of her utility for this entire piece of cake. The desired bound then follows by linearity. □

We remark that since we consider more general cost functions in this section, the egalitarian ratio $\frac{\alpha}{n-(n-1)\alpha}$ is still optimal by Theorem 8.18. However, unlike the canonical leximin solution, the generalized version is no longer Pareto optimal, since it may be possible to improve the utility of all agents by choosing more than an α fraction in certain parts of the cake and less in other parts. An interesting question is therefore whether we can obtain Pareto optimality while maintaining truthfulness and the egalitarian ratio.

8.8 Conclusion and Future Work

In this chapter, we have studied truthful and fair mechanisms in the cake sharing setting where all agents share the same subset of a divisible resource. We established the leximin solution as an attractive mechanism due to its truthfulness and its optimal egalitarian ratio among all truthful and position oblivious mechanisms. On the other hand, we constructed an intricate example showing that the maximum Nash welfare solution, which often exhibits desirable properties in other settings, fails to yield truthfulness in cake sharing. Moreover, we showed that in the absence of blocking, no truthful, Pareto optimal, and position oblivious mechanism can achieve a positive egalitarian ratio—in particular, this implies that the leximin solution is *not* truthful without blocking. An intriguing question is whether the impossibility still holds if we remove Pareto optimality or position obliviousness (or both), or whether there is a truthful mechanism that attains a non-trivial fairness guarantee even when blocking is not allowed.

In future research, it would be interesting to extend our cake sharing model to capture other practical scenarios. One natural direction is to allow agents to have more complex preferences beyond piecewise uniform utilities; the first step would be to consider *piecewise constant* utilities, where an agent's density function is constant over subintervals of the cake. Another direction is to allow non-uniform costs over the cake—this models, e.g., the fact that reserving a sports facility or a conference room can be more expensive during peak periods. Our preliminary result in Section 8.7 shows that a generalization of the leximin solution is still truthful and achieves the optimal egalitarian ratio for piecewise constant cost functions. Other questions addressed in cake cutting, such as the *price of truthfulness* and the *price of fairness* (i.e., the loss of social welfare due to truthfulness and fairness, respectively), are equally relevant in cake sharing as well.

Chapter 9

Conclusion and Open Problems

As algorithmic systems and the internet are now revolutionizing how society allocates its resources as well as start to inform or make important decisions in our society, it is crucial to make sure that the technologies are used in a responsible and beneficial manner. With this in mind, this dissertation contributes to the study of *fair division*, with a special emphasis on providing theoretical results for various fair division settings.

We have been concerned with fair division of a mixture of divisible and indivisible goods in Part I. The mixed goods setting not only captures deeply practical scenarios, but also gives rise to conceptually and mathematically challenging questions. To this end, we studied the envy-freeness for mixed goods (EFM), a generalization of envy-freeness and EF1, as well as the maximin share (MMS) guarantee in Chapters 4 and 5, respectively. We showed that an EFM allocation always exists for any number of agents with additive valuations. This result relies on the perfect allocation oracle, which, however, cannot be implemented in a bounded time in the RW model. It then leads to the following open question:

Question 9.1. Does there exist a bounded, or even finite, protocol in the Robertson-Webb model that computes an EFM allocation in the general setting for any number of agents?

We also provided bounded protocols to compute EFM allocations in special cases, and an ϵ-EFM allocation in the general setting in time poly$(n, m, 1/\epsilon)$. It remains open to design an algorithm that runs in time poly$(n, m, \log(1/\epsilon))$.

In addition, we present several preliminary results when considering EFM in conjunction with economic efficiency notions, but overall, some fundamental questions still remain open:

Question 9.2. Do weak EFM and PO allocations always exist in the mixed goods model?

En route, it would be tempting to first consider the special case with indivisible goods and a single *homogeneous* divisible good (e.g., money), since even though this case is well-studied in the literature when there is enough money, it remains an open question when it comes to fair and efficient allocations. Also relevant is to further generalize the setting where the resources to be allocated can be goods or chores and can be indivisible or divisible.

Question 9.3. Do EFM allocations always exist in this much more general setting?

On the MMS front, we analyzed the relation of the worst-case MMS approximation guarantees between mixed goods instances and indivisible goods instances, and presented an algorithm to produce an α-MMS allocation, where α monotonically increases in terms of the ratio between agents' values for the divisible goods and their maximin share.

Question 9.4. For future work, how to further improve the MMS approximation guarantee?

Besides EFM and MMS guarantee considered in this dissertation, one could also generalize and/or study other fairness notions when allocating mixed goods. How well would this notion behave with mixed goods in terms of its existence, approximation, and computation?

Another direction is to extend the mixed goods model to capture other practical scenarios. For instance, we assume additive valuations and our results for MMS guarantee rely on this assumption. Since substitute and complementary goods occur in practice, it would be interesting to design algorithms for fair division problems with more general valuations. Our EFM algorithmic framework (Section 4.2), as we remarked, works for any valuations over bundles of indivisible goods as long as the valuations over divisible goods are *additive*.

Question 9.5. Do EFM allocations exist when valuations for divisible goods don't add up?

Last but not least, extensions pointed out in the recent survey by Aziz [12] for indivisible or divisible items, e.g., unequal entitlements of the participants [71, 77], are relevant for mixed goods as well. Overall, we believe that fair division with mixed resources encodes a rich structure and creates a new research direction that deserves to be pursued for future work. For instance, we may analyze the price of fairness (see Chapter 6) for those notions in the mixed goods setting. Since both the price of EF for divisible goods and the price of EF1 for indivisible goods are settled [29, 38, 40, 67], it is natural and intriguing to resolve:

Question 9.6. What is the price of EFM?

In addition to economic efficiency we mentioned earlier, we have also studied the tradeoff between fairness and connectivity constraint in Part II, (both in a setting considers exclusively indivisible goods); we provide thorough discussions in Chapters 6 and 7. It would be interesting to further consider other aspects which are desirable, e.g., truthfulness, and to see if it fares well with fairness considerations. This dissertation has concerned truthfulness solely in Part III. Specifically, we presented yet another model—cake sharing—where all participants share the same selected subset of a cake which must be chosen subject to a length constraint, and focused on designing truthfulness and fairness mechanisms.

Besides the works related to truthfulness in cake cutting (see Chapter 8), truthfulness has also been concerned in fair division with indivisible goods [5, 6, 25, 94, 98]. While in general truthfulness and fairness are incompatible, positive results have been shown if agents' valuations are restricted, especially when valuations for indivisible goods are binary.

Question 9.7. In a restricted mixed goods setting where, for instance, agents have binary valuations, is it possible to design a mechanism which achieves truthfulness and fairness (e.g., EFM) simultaneously?

Bibliography

[1] Rediet Abebe, Jon Kleinberg, and David C. Parkes. Fair division via social comparison. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 281–289, 2017. URL https://dl.acm.org/doi/10.5555/3091125.3091171.

[2] Stéphane Airiau, Haris Aziz, Ioannis Caragiannis, Justin Kruger, Jérôme Lang, and Dominik Peters. Portioning using ordinal preferences: Fairness and efficiency. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 11–17, 2019. doi:10.24963/ijcai.2019/2.

[3] Ahmet Alkan, Gabrielle Demange, and David Gale. Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59(4):1023–1039, 1991. doi:10.2307/2938172.

[4] Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987. doi:10.1016/0001-8708(87)90055-7.

[5] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. On truthful mechanisms for maximin share allocations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 31–37, 2016. URL https://www.ijcai.org/Abstract/16/012.

[6] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 545–562, 2017. doi:10.1145/3033274.3085147.

[7] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)*, 13(4):52:1–52:28, 2017. doi:10.1145/3147173.

[8] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A. Voudouris. Maximum Nash welfare and other stories about EFX. *Theoretical Computer Science (TCS)*, 863:69–85, 2021. doi:10.1016/j.tcs.2021.02.020.

[9] Enriqueta Aragones. A derivation of the money rawlsian solution. *Social Choice and Welfare*, 12(3):267–276, 1995. doi:10.1007/BF00179981.

[10] Eshwar Ram Arunachaleswaran, Siddharth Barman, and Nidhi Rathi. Fully polynomial-time approximation schemes for fair rent division. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1994–2013, 2019. doi:10.1137/1.9781611975482.121.

[11] Yonatan Aumann and Yair Dombb. The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation (TEAC)*, 3(4):23:1–23:16, 2015. doi:10.1145/2781776.

[12] Haris Aziz. Developments in multi-agent fair allocation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 13563–13568, 2020. doi:10.1609/aaai.v34i09.7082.

[13] Haris Aziz. Achieving envy-freeness and equitability with monetary transfers. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5102–5109, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16645.

[14] Haris Aziz and Florian Brandl. Efficient, fair, and incentive-compatible healthcare rationing. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 103–104, 2021. doi:10.1145/3465456.3467531.

[15] 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 IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 416–427, 2016. doi:10.1109/FOCS.2016.52.

[16] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 454–464, 2016. doi:10.1145/2897518.2897522.

[17] Haris Aziz and Nisarg Shah. Participatory budgeting: Models and approaches. In Tamás Rudas and Gábor Péli, editors, *Pathways Between Social Science and Computational Social Science: Theories, Methods, and Interpretations*, pages 215–236. Springer International Publishing, 2021.

[18] Haris Aziz and Chun Ye. Cake cutting algorithms for piecewise constant and piecewise uniform valuations. In *Proceedings of the 10th International Conference on Web and Internet Economics (WINE)*, pages 1–14, 2014. doi:10.1007/978-3-319-13129-0_1.

[19] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. Algorithms for max-min share fair allocation of indivisible chores. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 335–341, 2017. URL https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14606.

[20] Haris Aziz, Sylvain Bouveret, Ioannis Caragiannis, Ira Giagkousi, and Jérôme Lang. Knowledge, fairness, and social constraints. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 4638–4645, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17230.

[21] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 53–59, 2019. doi:10.24963/ijcai.2019/8.

[22] Haris Aziz, Hau Chan, and Bo Li. Weighted maxmin fair share allocation of indivisible chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 46–52, 2019. doi:10.24963/ijcai.2019/7.

[23] Haris Aziz, Bo Li, and Xiaowei Wu. Strategyproof and approximately maxmin fair share allocation of chores. In *Proceedings of the 28th International Joint Conference on artificial intelligence (IJCAI)*, pages 60–66, 2019. doi:10.24963/ijcai.2019/9.

[24] Haris Aziz, Anna Bogomolnaia, and Hervé Moulin. Fair mixing: The case of dichotomous preferences. *ACM Transactions on Economics and Computation (TEAC)*, 8(4):18:1–18:27, 2020. doi:10.1145/3417738.

[25] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5119–5126, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16647`.

[26] Nikhil Bansal and Maxim Sviridenko. The Santa Claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 31–40, 2006. doi:10.1145/1132516.1132522.

[27] Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation (TEAC)*, 8(1):5:1–5:28, 2020. doi:10.1145/3381525.

[28] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 557–574, 2018. doi:10.1145/3219166.3219176.

[29] Siddharth Barman, Umang Bhaskar, and Nisarg Shah. Optimal bounds on the price of fairness for indivisible goods. In *Proceedings of the 16th International Conference on Web and Internet Economics (WINE)*, pages 356–369, 2020. doi:10.1007/978-3-030-64946-3_25.

[30] Xiaohui Bei and Warut Suksompong. Dividing a graphical cake. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5159–5166, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16652`.

[31] 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. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4980`.

[32] Xiaohui Bei, Ning Chen, Guangda Huzhang, Biaoshuai Tao, and Jiajun Wu. Cake cutting: Envy and truth. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3625–3631, 2017. doi:10.24963/ijcai.2017/507.

[33] 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. doi:10.24963/ijcai.2017/508.

[34] Xiaohui Bei, Guangda Huzhang, and Warut Suksompong. Truthful fair division without free disposal. *Social Choice and Welfare*, 55:523–545, 2020. doi:10.1007/s00355-020-01256-0.

[35] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. The price of connectivity in fair division. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5151–5158, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16651`.

[36] Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. Fair division of mixed divisible and indivisible goods. *Artificial Intelligence (AIJ)*, 293:103436, 2021. doi:10.1016/j.artint.2020.103436. Preliminary version in AAAI'20.

[37] Xiaohui Bei, Shengxin Liu, Xinhang Lu, and Hongao Wang. Maximin fairness with mixed divisible and indivisible goods. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 35(2):34, 2021. doi:10.1007/s10458-021-09517-7. Preliminary version in AAAI'21.

[38] Xiaohui Bei, Xinhang Lu, Pasin Manurangsi, and Warut Suksompong. The price of fairness for indivisible goods. *Theory of Computing Systems (TOCS)*, Forthcoming. doi:10.1007/s00224-021-10039-8. Preliminary version in IJCAI'19.

[39] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. (Almost full) EFX exists for four agents (and beyond). *CoRR*, abs/2102.10654, 2021. URL `https://arxiv.org/abs/2102.10654v2`.

[40] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. The price of fairness. *Operations Research*, 59(1):17–31, 2011. doi:10.1287/opre.1100.0865.

[41] Aurélie Beynier, Yann Chevaleyre, Laurent Gourvès, Ararat Harutyunyan, Julien Lesca, Nicolas Maudet, and Anaëlle Wilczynski. Local envy-freeness in house allocation problems. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 33:591–627, 2019. doi:10.1007/s10458-019-09417-x.

[42] Ivona Bezáková and Varsha Dani. Allocating indivisible goods. *ACM SIGecom Exchanges*, 5(3):11–18, 2005. doi:10.1145/1120680.1120683.

[43] Umang Bhaskar, AR 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)*, 2021. Forthcoming.

[44] Vittorio Bilò, Angelo Fanelli, Michele Flammini, Gianpiero Monaco, and Luca Moscardelli. The price of envy-freeness in machine scheduling. *Theoretical Computer Science (TCS)*, 613:65–78, 2016. doi:10.1016/j.tcs.2015.10.046.

[45] 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. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 14:1–14:21, 2019. doi:10.4230/LIPIcs.ITCS.2019.14.

[46] 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. doi:10.24963/ijcai.2018/13.

[47] Anna Bogomolnaia and Hervé Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004. doi:10.1111/j.1468-0262.2004.00483.x.

[48] John Adrian Bondy and U. S. R. Murty. *Graph Theory*. Springer, 1st edition, 2008.

[49] Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2016. doi:10.1007/s10458-015-9287-3.

[50] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In Brandt et al. [60], chapter 12, pages 284–310.

[51] 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. doi:10.24963/ijcai.2017/20.

[52] Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. Chore division on a graph. *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 33(5):540–563, 2019. doi:10.1007/s10458-019-09415-z.

[53] Steven J. Brams and Peter C. Fishburn. Fair division of indivisible items between two people with identical preferences: Envy-freeness, Pareto-optimality, and equity. *Social Choice and Welfare*, 17(2):247–267, 2000. doi:10.1007/s003550050019.

[54] Steven J. Brams and Alan D. Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995. doi:10.2307/2974850.

[55] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[56] Steven J. Brams and Alan D. Taylor. A procedure for divorce settlements. *Mediation Quarterly*, 13(3):191–205, 1996. doi:10.1002/crq.3900130305.

[57] Steven J. Brams, D. Marc Kilgour, and Christian Klamler. The undercut procedure: An algorithm for the envy-free division of indivisible items. *Social Choice and Welfare*, 39(2):615–631, 2012. doi:10.1007/s00355-011-0599-1.

[58] Steven J. Brams, Marc Kilgour, and Christian Klamler. Two-person fair division of indivisible items: An efficient, envy-free algorithm. *Notices of the AMS*, 61(2):130–141, 2014. doi:10.1090/noti1075.

[59] Florian Brandl, Felix Brandt, Dominik Peters, Christian Stricker, and Warut Suksompong. Funding public projects: A case for the Nash product rule. *CoRR*, abs/2005.07997, 2020. URL `https://arxiv.org/abs/2005.07997v1`.

[60] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[61] Simina Brânzei and Peter Bro Miltersen. A dictatorship theorem for cake cutting. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 482–488, 2015. URL `http://ijcai.org/Abstract/15/074`.

[62] Simina Brânzei and Noam Nisan. The query complexity of cake cutting. *CoRR*, abs/1705.02946, 2017. URL `https://arxiv.org/abs/1705.02946v3`.

[63] Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier. Envy-free allocations respecting social networks. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 283–291, 2018. URL `https://dl.acm.org/doi/10.5555/3237383.3237430`.

[64] Johannes Brustle, Jack Dippel, Vishnu V. Narayan, Mashbat Suzuki, and Adrian Vetta. One dollar each eliminates envy. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 23–39, 2020. doi:10.1145/3391403.3399447.

[65] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy (JPE)*, 119(6):1061–1103, 2011. doi:10.1086/664613.

[66] Ioannis Caragiannis and Stavros Ioannidis. Computing envy-freeable allocations with limited subsidies. *CoRR*, abs/2002.02789, 2020. URL https://arxiv.org/abs/2002.02789v1.

[67] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. The efficiency of fair division. *Theory of Computing Systems (TOCS)*, 50 (4):589–610, 2012. doi:10.1007/s00224-011-9359-y.

[68] Ioannis Caragiannis, Nick Gravin, and Xin Huang. Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 527–545, 2019. doi:10.1145/3328526.3329574.

[69] 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 (TEAC)*, 7(3):12:1–12:32, 2019. doi:10.1145/3355902.

[70] Katarína Cechlárová and Eva Pillárová. On the computability of equitable divisions. *Discrete Optimization*, 9(4):249–257, 2012. doi:10.1016/j.disopt.2012.08.001.

[71] Mithun Chakraborty, Ayumi Igarashi, Warut Suksompong, and Yair Zick. Weighted envy-freeness in indivisible item allocation. *ACM Transactions on Economics and Computation (TEAC)*, Forthcoming. Preliminary version in AAMAS'20.

[72] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. EFX exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 1–19, 2020. doi:10.1145/3391403.3399511.

[73] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2658–2672, 2020. doi:10.1137/1.9781611975994.162.

[74] Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, Ruta Mehta, and Pranabendu Misra. Improving EFX guarantees through rainbow cycle number. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 310–311, 2021. doi:10.1145/3465456.3467605.

[75] Yiling Chen, John K. Lai, David C. Parkes, and Ariel D. Procaccia. Truth, justice, and cake cutting. *Games and Economic Behavior (GEB)*, 77(1):284–297, 2013. doi:10.1016/j.geb.2012.10.009.

[76] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 629–646, 2017. doi:10.1145/3033274.3085125.

[77] Ágnes Cseh and Tamás Fleiner. The complexity of cake cutting with unequal shares. *ACM Transactions on Algorithms (TALG)*, 16(3):29:1–29:21, 2020. doi:10.1145/3380742.

[78] John P. Dickerson, Jonathan Goldman, Jeremy Karp, Ariel D. Procaccia, and Tuomas Sandholm. The computational rise and fall of fairness. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1405–1411, 2014. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8348`.

[79] Lester E. Dubins and Edwin Henry Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1):1–17, 1961. doi:10.2307/2311357.

[80] Jeff Edmonds and Kirk Pruhs. Balanced allocations of cake. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 623–634, 2006. doi:10.1109/FOCS.2006.17.

[81] Jeff Edmonds and Kirk Pruhs. Cake cutting really is not a piece of cake. *ACM Transactions on Algorithms (TALG)*, 7(4):51:1–51:12, 2011. doi:10.1145/2000807.2000819.

[82] Ulle Endriss, editor. *Trends in Computational Social Choice*. AI Access, 2017.

[83] Shimon Even and Azaria Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285–296, 1984. doi:10.1016/0166-218X(84)90005-2.

[84] Shimon Even and Robert Endre Tarjan. Computing an st-numbering. *Theoretical Computer Science (TCS)*, 2(3):339–344, 1976. doi:10.1016/0304-3975(76)90086-4.

[85] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner voting: A new challenge for social choice theory. In Endriss [82], chapter 2, pages 27–47.

[86] Alireza Farhadi, Mohammad Ghodsi, MohammadTaghi Hajiaghayi, Sébastien Lahaie, David Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods to asymmetric agents. *Journal of Artificial Intelligence Research (JAIR)*, 64(1):1–20, 2019. doi:10.1613/jair.1.11291.

[87] Uriel Feige, Ariel Sapir, and Laliv Tauber. A tight negative example for mms fair allocations. *CoRR*, abs/2104.04977v1, 2021. URL `https://arxiv.org/abs/2104.04977v1`.

[88] Duncan Karl Foley. Resource allocation and the public sector. *Yale Economics Essays*, 7(1):45–98, 1967.

[89] Eric J. Friedman, Vasilis Gkatzelis, Christos-Alexandros Psomas, and Scott Shenker. Fair and efficient memory sharing: Confronting free riders. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1965–1972, 2019. doi:10.1609/aaai.v33i01.33011965.

[90] Ya'akov (Kobi) Gal, Moshe Mash, Ariel D. Procaccia, and Yair Zick. Which is the fairest (rent division) of them all? *Journal of the ACM (JACM)*, 64(6):39:1–39:22, 2017. doi:10.1145/3131361.

[91] Jugal Garg and Setareh Taki. An improved approximation algorithm for maximin shares. *Artificial Intelligence (AIJ)*, 300:103547, 2021. doi:10.1016/j.artint.2021.103547.

[92] Jugal Garg, Peter McGlaughlin, and Setareh Taki. Approximating maximin share allocations. In *Proceedings of the 2nd Symposium on Simplicity in Algorithms (SOSA)*, pages 20:1–20:11, 2019. doi:10.4230/OASIcs.SOSA.2019.20.

[93] Mohammad Ghodsi, MohammadTaghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvement. *Mathematics of Operations Research*, Forthcoming. doi:10.1287/moor.2020.1096.

[94] Hiromichi Goko, Ayumi Igarashi, Yasushi Kawase, Kazuhisa Makino, Hanna Sumita, Akihisa Tamura, Yu Yokoi, and Makoto Yokoo. Fair and truthful mechanism with limited subsidy. *CoRR*, abs/2105.01801, 2021. URL `https://arxiv.org/abs/2105.01801v1`.

[95] Jonathan Goldman and Ariel D. Procaccia. Spliddit: Unleashing fair division algorithms. *SIGecom Exchanges*, 13(2):41–46, 2015. doi:10.1145/2728732.2728738.

[96] Laurent Gourvès and Jérôme Monnot. On maximin share allocations in matroids. *Theoretical Computer Science (TCS)*, 754:50–64, 2019. doi:10.1016/j.tcs.2018.05.018.

[97] Daniel Halpern and Nisarg Shah. Fair division with subsidy. In *Proceedings of the 12th International Symposium on Algorithmic Game Theory (SAGT)*, pages 374–389, 2019. doi:10.1007/978-3-030-30473-7_25.

[98] Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. Fair division with binary valuations: One rule to rule them all. In *Proceedings of the 16th International Conference on Web and Internet Economics (WINE)*, pages 370–383, 2020. doi:10.1007/978-3-030-64946-3_26.

[99] Sandy Heydrich and Rob van Stee. Dividing connected chores fairly. *Theoretical Computer Science (TCS)*, 593:51–61, 2015. doi:10.1016/j.tcs.2015.05.041.

[100] John Hopcroft and Robert Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM (CACM)*, 16(6):372–378, 1973. doi:10.1145/362248.362272.

[101] Xin Huang and Pinyan Lu. An algorithmic framework for approximating maximin share allocation of chores. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 630–631, 2021. doi:10.1145/3465456.3467555.

[102] 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. doi:10.1609/aaai.v33i01.33012045.

[103] Heinz A. Jung. Eine Verallgemeinerung des n-fachen Zusammenhangs für Graphen. *Mathematische Annalen*, 187(2):95–103, 1970. doi:10.1007/BF01350174.

[104] D. Marc Kilgour and Rudolf Vetschera. Two-player fair division of indivisible items: Comparison of algorithms. *European Journal of Operational Research (EJOR)*, 271(2):620–631, 2018. doi:10.1016/j.ejor.2018.05.057.

[105] Flip Klijn. An algorithm for envy-free allocations in an economy with indivisible objects and money. *Social Choice and Welfare*, 17(2):201–215, 2000. doi:10.1007/s003550050015.

[106] David Kurokawa, John K. Lai, and Ariel D. Procaccia. How to cut a cake before the party ends. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, pages 555–561, 2013. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6365.

[107] David Kurokawa, Ariel D. Procaccia, and Nisarg Shah. Leximin allocations in the real world. *ACM Transactions on Economics and Computation (TEAC)*, 6(3-4):11:1–11:24, 2018. doi:10.1145/3274641.

[108] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM (JACM)*, 65(2):8:1–8:27, 2018. doi:10.1145/3140756.

[109] Sascha Kurz. The price of fairness for a small number of indivisible items. In *Operations Research Proceedings*, pages 335–340, 2014. doi:10.1007/978-3-319-28697-6_47.

[110] Maria Kyropoulou, Warut Suksompong, and Alexandros A. Voudouris. Almost envy-freeness in group resource allocation. *Theoretical Computer Science (TCS)*, 841: 110–123, 2020. doi:10.1016/j.tcs.2020.07.008.

[111] A. A. Liapounoff. Sur les fonctions-vecteurs complètement additives. *Izvestiya Akademii Nauk SSSR. Seriya Matematicheskaya*, 4(6):465–478, 1940. URL http://mi.mathnet.ru/eng/izv3907.

[112] Claudia Lindner and Jörg Rothe. Cake-cutting: Fair division of divisible goods. In Jörg Rothe, editor, *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 7, pages 395–491. Springer-Verlag Berlin Heidelberg, 2016.

[113] Richard J. 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. doi:10.1145/988772.988792.

[114] Zbigniew Lonc and Miroslaw Truszczynski. Maximin share allocations on cycles. *Journal of Artificial Intelligence Research (JAIR)*, 69:613–655, 2020. doi:10.1613/jair.1.11702.

[115] Ryoga Mahara. Existence of EFX for two additive valuations. *CoRR*, abs/2008.08798, 2021. URL https://arxiv.org/abs/2008.08798v2.

[116] Pasin Manurangsi and Warut Suksompong. Computing a small agreeable set of indivisible items. *Artificial Intelligence (AIJ)*, 268:96–114, 2019. doi:10.1016/j.artint.2018.10.001.

[117] Pasin Manurangsi and Warut Suksompong. When do envy-free allocations exist? *SIAM Journal on Discrete Mathematics (SIDMA)*, 34(3):1505–1521, 2020. doi:10.1137/19M1279125.

[118] Pasin Manurangsi and Warut Suksompong. Closing gaps in asymptotic fair division. *SIAM Journal on Discrete Mathematics (SIDMA)*, 35(2):668–706, 2021. doi:10.1137/20M1353381.

[119] Evangelos Markakis. Approximation algorithms and hardness results for fair division with indivisible goods. In Endriss [82], chapter 12, pages 231–247.

[120] Eric S. Maskin. On the fair allocation of indivisible goods. In George R. Feiwel, editor, *Arrow and the Foundations of the Theory of Economic Policy*, chapter 11, pages 341–349. Palgrave Macmillan UK, 1987. doi:10.1007/978-1-349-07357-3_12.

[121] Avishay Maya and Noam Nisan. Incentive compatible two player cake cutting. In *Proceedings of the 8th International Workshop on Internet and Network Economics (WINE)*, pages 170–183, 2012. doi:10.1007/978-3-642-35311-6_13.

[122] Marc Meertens, Jos Potters, and Hans Reijnierse. Envy-free and Pareto efficient allocations in economies with indivisible goods and money. *Mathematical Social Sciences*, 44(3):223–233, 2002. doi:10.1016/S0165-4896(02)00064-1.

[123] Vijay Menon and Kate Larson. Deterministic, strategyproof, and fair cake cutting. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 352–358, 2017. doi:10.24963/ijcai.2017/50.

[124] Gábor Mészáros. *Linkedness and Path-Pairability in the Cartesian Product of Graphs*. PhD thesis, Department of Mathematics and its Applications, Central European University, 2015.

[125] Marcin Michorzewski, Dominik Peters, and Piotr Skowron. Price of fairness in budget division and probabilistic social choice. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2184–2191, 2020. doi:10.1609/aaai.v34i02.5594.

[126] Neeldhara Misra and Aditi Sethia. Fair division is hard even for amicable agents. In *Proceedings of the 21st Italian Conference on Theoretical Computer Science (ICTCS)*, pages 202–207, 2020.

[127] Hervé Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003.

[128] Hervé Moulin. Fair division in the internet age. *Annual Review of Economics*, 11(1): 407–441, 2019. doi:10.1146/annurev-economics-080218-025559.

[129] Hoon Oh, Ariel D. Procaccia, and Warut Suksompong. Fairly allocating many goods with few queries. *SIAM Journal on Discrete Mathematics (SIDMA)*, 35(2):788–813, 2021. doi:10.1137/20M1313349.

[130] Parag A. Pathak, Tayfun Sönmez, M. Utku Ünver, and M. Bumin Yenmez. Fair allocation of vaccines, ventilators and antiviral treatments: Leaving no ethical value behind in health care rationing. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 785–786, 2021. doi:10.1145/3465456.3467604.

[131] Dominik Peters. Proportionality and strategyproofness in multiwinner elections. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1549–1557, 2018. URL `https://dl.acm.org/doi/10.5555/3237383.3237931`.

[132] Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics (SIDMA)*, 34(2):1039–1068, 2020. doi:10.1137/19M124397X.

[133] Benjamin Plaut and Tim Roughgarden. Communication complexity of discrete fair division. *SIAM Journal on Computing (SICOMP)*, 49(1):206–243, 2020. doi:10.1137/19M1244305.

[134] Ariel D. Procaccia. Cake cutting algorithms. In Brandt et al. [60], chapter 13, pages 311–329.

[135] Ariel D. Procaccia. An answer to fair division's most enigmatic question: Technical perspective. *Communications of the ACM (CACM)*, 63(4):118, 2020. doi:10.1145/3382131.

[136] Ariel D. Procaccia and Junxing Wang. A lower bound for equitable cake cutting. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 479–495, 2017. doi:10.1145/3033274.3085107.

[137] Ariel D. Procaccia, Rodrigo A. Velez, and Dingli Yu. Fair rent division on a budget. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1177–1184, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16936`.

[138] Jack Robertson and William Webb. *Cake-Cutting Algorithm: Be Fair If You Can*. A K Peters/CRC Press, 1998.

[139] Jörg Rothe, editor. *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer-Verlag Berlin Heidelberg, 2016.

[140] Alexander Rubchinsky. Brams-Taylor model of fair division for divisible and indivisible items. *Mathematical Social Sciences*, 60(1):1–14, 2010. doi:10.1016/j.mathsocsci.2010.03.004.

[141] Fedor Sandomirskiy and Erel Segal-Halevi. Efficient fair division with minimal sharing. *CoRR*, abs/1908.01669, 2020. URL `https://arxiv.org/abs/1908.01669v2`.

[142] Jens M. Schmidt. A simple test on 2-vertex- and 2-edge-connectivity. *Information Processing Letters*, 113(7):241–244, 2013. doi:10.1016/j.ipl.2013.01.016.

[143] Erel Segal-Halevi and Warut Suksompong. Democratic fair allocation of indivisible goods. *Artificial Intelligence (AIJ)*, 277:103167, 2019. doi:10.1016/j.artint.2019.103167.

[144] Erel Segal-Halevi and Balázs Róbert Sziklai. Monotonicity and competitive equilibrium in cake-cutting. *Economic Theory*, 68(2):363–401, 2019. doi:10.1007/s00199-018-1128-6.

[145] Piotr Skowron, Piotr Faliszewski, and Jérôme Lang. Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artificial Intelligence (AIJ)*, 241:191–216, 2016. doi:10.1016/j.artint.2016.09.003.

[146] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948. URL `http://www.jstor.org/stable/1914289`.

[147] Walter Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980. doi:10.2307/2320951.

[148] Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *The American Mathematical Monthly*, 106(10):930–942, 1999. doi:10.1080/00029890.1999.12005142.

[149] Warut Suksompong. Approximate maximin shares for groups of agents. *Mathematical Social Sciences*, 92:40–47, 2018. doi:10.1016/j.mathsocsci.2017.09.004.

[150] Warut Suksompong. Fairly allocating contiguous blocks of indivisible items. *Discrete Applied Mathematics*, 260:227–236, 2019. doi:10.1016/j.dam.2019.01.036.

[151] Ankang Sun, Bo Chen, and Xuan Vinh Doan. Connections between fairness criteria and efficiency for allocating indivisible chores. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1281–1289, 2021. URL `https://dl.acm.org/doi/10.5555/3463952.3464100`.

[152] Lars-Gunnar Svensson. Large indivisibles: An analysis with respect to price equilibrium and fairness. *Econometrica*, 51(4):939–954, 1983. doi:10.2307/1912044.

[153] William Thomson. Introduction to the theory of fair allocation. In Brandt et al. [60], chapter 11, pages 261–283.

[154] Hal R. Varian. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1):63–91, 1974. doi:10.1016/0022-0531(74)90075-1.

[155] Rodrigo A. Velez. Equitable rent division. *ACM Transactions on Economics and Computation (TEAC)*, 6(2):9:1–9:25, 2018. doi:10.1145/3274528.

[156] Dietrich Weller. Fair division of a measurable space. *Journal of Mathematical Economics*, 14(1):5–17, 1985. doi:10.1016/0304-4068(85)90023-0.

[157] Hassler Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932. doi:10.2307/2371086.

[158] Hassler Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 34(2):339–362, 1932. doi:10.2307/1989545.

[159] Gerhard J. Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters (ORL)*, 20(4): 149–154, 1997. doi:10.1016/S0167-6377(96)00055-7.

[160] Hobart Peyton Young. *Equity: In Theory and Practice*. Princeton University Press, 1995.