FAIR SHARE ALLOCATION OF INDIVISIBLE CHORES

YU ZHOU

PhD

The Hong Kong Polytechnic University

2023

The Hong Kong Polytechnic University

Department of Computing

Fair Share Allocation of Indivisible Chores

Yu Zhou

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

May 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

__________________________ (Signed)

_________ Yu Zhou _________ (Name of student)

ABSTRACT

Since initiated by Steinhaus at a meeting in Washington D.C. [Econometrica, 1948], fair allocation has been broadly studied in the fields of economics, mathematics and computer science. A substantial body of works aimed at understanding the theory of fairly allocating a set of items to a set of agents have appeared consequently. In this thesis, we study the problem of fairly allocating m indivisible chores (i.e., undesired items with non-negative disutilities) to n agents, with particular focus on share-based fairness notions, where agents evaluate the fairness of an allocation by comparing their received disutilities with a benchmark *share* - a function only of her own disutility function and the number of agents. This share is called a *guarantee* if for any profile of disutility functions there is an allocation where every agent receives disutility no more than her own share.

We first consider the notion of MaxMinShare (MMS) proposed by Budish [J. Political Econ., 2011] for indivisible goods (i.e., desired items with non-negative utilities). For indivisible chores, this notion becomes MinMaxShare, which is also abbreciated to MMS for consistency. In the literature, the majority of effort on finding MMS fair allocations for chores is devoted to additive disutility funtions; however, beyond additivity, very little is known.

We prove that no algorithm can ensure better than $\min\{n, \frac{\log m}{\log\log m}\}$ approximation if the disutility functions are submodular. This result shows a sharp contrast to the allocation of goods where constant approximations exist as shown by Barman and Krishnamurthy [TEAC, 2020] and Ghodsi et al. [AIJ, 2022]. We then prove that for subadditive disutilities, there always exists an allocation that is $\min\{n, \lceil\log m\rceil\}$-approximation, and thus the approximation ratio is asymptotically tight. Besides multiplicative approximation, we also consider the ordinal relaxation, 1-out-of-d MMS, which was recently proposed by Hosseini et al. [JAIR and AAMAS, 2022]. Our result implies that for any $d \geq 2$, a 1-out-of-d MMS allocation may not exist. Due to these hardness results in the general subadditive setting, we study two specific problems, namely, job scheduling and bin packing. For both problems, we show that constant approximate allocations always exist for both multiplicative and ordinal relaxations of MMS.

Since exact MMS fairness cannot be guaranteed as shown by Feige et al. [WINE, 2021], we turn to another share-based notion proposed by Hill [Ann. Probab., 1987], which is the worst-case MaxMinShare over all utility functions with the same largest possible single-item utility. Although Hill's share is more conservative than the MaxMinShare, it can always be guaranteed and its computation is elementary, unlike that of the MaxMinShare which involves solving an NP-hard problem. We apply Hill's approach to the allocation of indivisible chores, and characterise the tight closed form of the worst-case MinMaxShare for a given disutility of the worst chore. We argue that Hill's share for allocating chores is effective in the sense of being close to the original MinMaxShare value, and there is much to learn about the guarantee an agent can be offered from the disutility of her worst single chore. Furthermore, we prove that the monotonic cover of Hill's share is the best guarantee that can be achieved in Hill's model for all allocation instances.

PUBLICATIONS

1. Haris Aziz, Bo Li, Shiji Xing, and Yu Zhou. "Possible Fairness for Allocating Indivisible Resources." In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, pp. 197-205. 2023. (published, authors are listed alphabetically)

2. Bo Li, Hervé Moulin, Ankang Sun, Yu Zhou. "On Hill's Worst-Case Guarantee for Indivisible Bads." In Conference On Web And InterNet Economics. 2023. (accepted, authors are listed alphabetically)

3. Bo Li, Fangxiao Wang, Yu Zhou. "Fair Allocation of Indivisible Chores: Beyond Additive Valuations." In Conference on Neural Information Processing Systems. 2023. (submitted, authors are listed alphabetically)

4. Zecheng Li, Yu Zhou, Songtao Guo, and Bin Xiao. "Solsaviour: A defending framework for deployed defective smart contracts." In Annual Computer Security Applications Conference, pp. 748-760. 2021. (published)

5. Yu Zhou, Shang Gao, Weiwei Qiu, Kai Lei, Bin Xiao. "On the Profitability of Selfish Mining Attack under the Checkpoint Mechanism."

In IEEE Global Communications Conference. 2023. (accepted)

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my dear supervisor—Dr. Bo Li, who gave me considerable encouragement and support not only in doing research but also in living my life. He was very patient when teaching me how to find a meaningful research problem and how to solve the problem step by step. He spent a lot of time in carefully polishing my papers and the responses to reviewers, even when he was quite busy. He was never stingy with his praise whenever I made some progress and was always generous with his encouragement whenever I felt down and became doubtful about myself. He was always there whenever I needed his help. He always put us students in the first place and was always thinking about how to make us better and better. To me, he is also a close friend besides a supervisor. A close friend with whom I could play together and laugh together; a close friend with whom I would like to share my own happiness; a close friend whom I could fully trust.

I am also very grateful to the teachers who offered me a lot of help during my PhD journey, including the thesis examiners Prof. Xujin Chen and Dr. Qi Qi, my co-supervisor Prof. Qing Li, DRC chair Prof. Wenjie Li, Prof. Xiaoming Sun and Prof. Jialin Zhang from Chinese Academy of Sciences, Dr. Jiarui Gan from University of Oxford, Prof. Minming Li from City University of Hong Kong, Prof. Haris Aziz from UNSW and Prof. Hervé Moulin from the University of Glasgow.

I also thank my friends very much, whose company kept me away from feeling alone and made my PhD journey full of happiness. We learned from each other, encouraged each other and helped each other. We had so many wonderful and unforgettable memories when we were together.

Finally, I owe an incredible debt of gratitude to my grandparents, who brought me up with their meticulous care. I also would like to express my heartfelt gratitude to my parents, my elder sister, my girlfriend, and other family members, for their constant love and encouragement that kept me brave in face of difficulties and supported me to get through all the tough periods.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In this chapter, we first review the history and development of the field of fair allocation and present some real-world examples and applications to familiarize the readers more with this field. We then briefly describe the fair allocation problem and its major components. We next present the research context of our works by reviewing the salient results in the literature and describe the problems we focus on in this thesis as well as their motivations. We also present the results we have discovered. This chapter is ended with a description of the structure of the remaining contents of this thesis.

1.1 Background

Fair allocation is an age-old problem with a long history. The oldest story can trace back to the Bible (Chapter 13 of the Book of Genesis): when Abraham and Lot came to a land named Canaan, they suggested that they divide it among them. Abraham first cut the land into a western part and an eastern part which were equally valuable to him. Then Lot chose the part that he preferred more and left Abraham with the other. At last, Lot got the eastern part which contains Sodom and Gomorrah, and Abraham got the western part which contains Beer Sheva, Hebron, Bethel, and Shechem. The way Abraham and Lot divided the land is the archetypal fair allocation algorithm "Divide-and-Choose", which admirably guaranteed that neither Abraham nor Lot was envious of the other.

Modern research on fair allocation was regarded to proliferate after a talk presented by a mathematician named Steinhaus at a meeting of the Econometric Society in Washington D.C [135]. In the talk, Steinhaus proposed extending the aforementioned problem in the Bible to arbitrary number of participants, which was positively answered by Banach and Knaster with a simple algorithm named "the last diminisher method". Later, from the late 1950s to the early 1990s, a lot of economists started to study the problem of fair allocation [134, 128, 69, 70, 130, 81, 50], albeit from different perspectives from computer scientists'. They studied fair allocation in the context of Arrow Debreu's economies and put forward many microeconomic theories. The conceptual insights of these theories closely followed the results of Distributive Justice in political philosophy. Nevertheless, the settings on which these economists focused were too general that they were short on applications.

In the past decades, considerable fresh energy from computer science has been poured into the field of fair allocation, largely stimulated by the emergence of the Internet. The ideas that computer scientists borrowed from computer science brought a fresh and promising perspective to the study of fair allocation – algorithmic fair allocation. Besides designing algorithms that solve specific problems of fair allocation, computer scientists also cared about the computational complexity of these algorithms, i.e., the amount of time, storage, communication cost and other resources to execute them. They introduced a diversity of advanced technologies into the field of fair allocation such as complexity theory, asymptotic analysis, approximation algorithms, etc. They also proposed many novel and practical settings that combine the characteristics of the Internet, and discovered many significant results regarding these settings. The effort by the computer scientists has made the theories of fair allocation much closer to the reality in the Age of Internet.

Nowadays, fair allocation plays a significant role in the real world, especially when more and more important decisions in our lives are made by computer systems which makes it imperative that these decisions are made transparently and fairly. To end this chapter and to further familiarize the readers with the problem of fair allocation, we present several concrete examples and applications of fair allocation.

Examples. The first example is allocating resources in cloud computing. Internet technology companies like Amazon, Google and Huawei use schedulers in clouds to allocate limited amount of scarce resources (e.g., servers, memory, GPU, CPU, etc.) among a number of self-interested users who want to maximize the utility of their own allocations [94, 141]; the scheduler's goal is to fairly allocate the resources to users and also maximize the utilization of resources. Fairness is also seen as a desired property in other areas of computer science such as computer networks and operating systems.

The second example is splitting assets, which is a common business in life.

When an elder dies, his assets (e.g., pensions, houses, savings, investments, etc) will be divided among his heirs; when a couple decides to divorce, their shared properties need to be distributed between them; when a company goes out of business, the company's assets and debts will be divided between the company's owners. More analogous scenarios can be seen in our daily life. For all these scenarios, the assets and properties should be divided in a fair way such that all involved parties are satisfied with the allocation.

The third and last example is combating climate change. Since 1800s, human activities have greatly accelerated the process of global warming, primary due to burning of fossil fuels like coal, oil and gas. To combat this issue, one of the consensuses reached by the world organization UNFCCC (i.e., the United Nations Framework Convention on Climate Change) is to stabilize the worldwide emissions of greenhouse gas. One critical concern in this consensus is how to fairly quantify the amount of greenhouse gas that each nation could emit, which should consider the nations' economic volumes, economic models, total amount of greenhouse gas emissions, etc.

Applications. The first application is Course Match, an original and innovative course registration system created by the Wharton School of the University of Pennsylvania. This system deploys advanced fair allocation algorithms to allocate courses to students based on students' preferences and course availability. These algorithms have made the process of allocating courses simpler and fairer. In use since Fall 2013, Course Match has been proven to increase student satisfaction and promote fairness in course allocation.

The second application is Spliddit (spliddit.org), a platform which provides online solutions to everyday fair allocation problems that are relevant to the society at large [90]. For example, people can use Spliddit to allocate rooms to housemates in the rent splitting problem, to distribute a set of goods in the good dividing problem, and to determine scientific credit of a paper or share credit for a project or divide a company bonus in the credit sharing problem. The solutions offered by Spliddit have been proven in the literature to be fair, equitable and efficient.

1.2 Literature Review

Fair allocation studies the problem of allocating a set of items to a set of agents in a fair manner [122]. The items are either goods (i.e., ones with non-negative utilities like natural resources) or chores (i.e., ones with non-positive utilities or non-negative disutilities like household duties). They can be divisible like lands which can be allocated fractionally, or indivisible like paintings each of which must be allocated as a whole. Agents have utility (or disutility) functions over the items, which could be binary, additive, submodular, subadditive, etc. For example, binary functions mean that the utility (or disutility) of each item is either 0 or 1 and additive functions mean that the utility (or disutility) of any set of items is the sum of the utilities (or disutilities) of these items. We say the agents are homogeneous if their utility (or disutility) functions over the items are the same, and we say they are heterogeneous if not. Fairness notions define the requirements for an allocation to be deemed fair by the agents. Take the two well-studied fairness notions (i.e., *envy-freeness* [83, 140] and *proportionality* [135]) for example, an envy-free allocation requires no agent to prefer the bundle of items allocated to any other agent more than her own, and a proportional one requires every agent to receive a utility no smaller than her average utility for all items. Besides fairness, efficiency is also a key concern in fair allocation. For example, one may desire that the sum of the utilities (or disutilities) received by the agents are maximized (or minimized), which is called *maximum utilitarian social welfare*. One may desire that no agent can improve her own utility (or disutility) without hurting other agents, which is called *Pareto optimality*.

The original study of fair allocation was concentrated on allocating divisible items, which is also known as the *cake-cutting problem* [49, 129]. Although this kind of problem is not the focus of this thesis, we start our literature review from it.

1.2.1 Divisible Items

In a cake-cutting problem, there is usually one divisible item which is represented by the interval $[0, 1]$. Agents may have different utilities (or disutilities) for different pieces of the item (even when these pieces are of the same length). As we have mentioned in the last chapter, the cake-cutting problem was first referred in the Bible where the "Divide-and-Choose" algorithm was deployed to allocate the divisible land to two agents. For the allocation of divisible goods, "Divide-and-Choose" remarkably guarantees both envy-freeness and proportionality for the setting with two agents. When the setting is extended from two agents to an arbitrary number of agents, the fairness notions become much harder to satisfy. Several elegant algorithms were proposed to compute allocations that satisfy proportionality [68, 80, 77, 71, 99, 27]. Nevertheless, the progress on envy-free allocations was much slower. Selfridge and Conway first proposed an algorithm that computes envy-free allocations for the three-agent setting (see [129]). After three decades, Brams and Taylor [48] made a significant breakthrough: they designed an algorithm that computes envy-free allocations for settings with arbitrary number of agents. One critical drawback of Brams and Taylor's algorithm is that its running time is unbounded. This drawback was finally resolved after another two decades by Aziz and Mackenzie [19] who proposed an algorithm that runs in a bounded number of steps. For the allocation of divisible chores, Dehghani et al. [66] proposed an algorithm that computes an envy-free allocation of divisible chores to an arbitrary number of agents in a bounded number of steps. Boodaghians et al. [42] designed a polynomial-time algorithm that computes an approximate competitive equilibrium. Chaudhury et al. [59] also studied the allocation of a mixture of divisible goods and chores.

Connectivity was a key concern when allocating divisible items. For example, people would not like to receive many disjoint pieces of a land. Dubins and Spanier [68] first proposed an algorithm that computes a connected proportional allocation for any number of agents. Even and Paz [77], Edmonds and Pruhs [72] subsequently improved the results by reducing the complexity of Dubins and Spanier's algorithm. While a connected proportional allocation is relatively easy to obtain, a connected envy-free one is not the case. It was first shown by Stromquist [136] and Edward Su [73] that a connected envy-free allocation indeed exists. However, such an allocation was proved by Stromquist [137] to require unbounded steps when there are more than two agents. Despite this negative result, Goldberg et al. [89] and Arunachaleswaran et al. [12] respectively designed a polynomial-time algorithm that computes a connected and approximately envy-free allocation for any number of agents.

Effort was also devoted to other sub-branches of fair allocation of divisible items. For example, Varian [140] provided a competitive equilibrium from equal incomes (CEEI) solution that simultaneously guarantees envy-freeness and Pareto-optimality. Brams et al. [47], Chen et al. [62], Mossel and Tamuz [121] studied the problem from a game-theoretic perspective where agents may have incentives to manipulate the algorithm like misreporting their true utility functions. Arzi et al. [13], Aumann and Dombb [14], Aumann et al. [15], Bertsimas et al. [37], Caragiannis et al. [56] analyzed the cost of social welfare it takes to guarantee fairness.

1.2.2 Indivisible Items

The recent focus in the literature is on indivisible items, which is motivated by the fact that most items in our daily life cannot be fractionally allocated. The cases with indivisible items are harder to deal with than those with divisible items, in the sense that absolutely fair allocations rarely exist. Consider the simple example where one indivisible item is allocated to two agents. The agent who receives the item is envied by the other agent which breaks envy-freeness, while the agent who does not receive the item gets a utility of zero which breaks proportionality. Consequently, exploring the extent to which the relaxations of envy-freeness and proportionality can be satisfied for indivisible items steps into the center of fair allocation.

Relaxations of Envy-freeness Two of the most notable relaxations of envy-freeness are envy-free up to one item (EF1) which allows the existence of an envy but requires that the envy could be eliminated by removing an item from the bundle of the envied agent, and envy-free up to any item (EFX) which requires the envy be eliminated by removing any item from the envied agent's bundle. Obviously, EFX is stronger than EF1 in the sense that any EFX allocation is also EF1. The notion of EF1 was first introduced by Budish [53] and first studied by Lipton et al. [115] who showed that an EF1 allocation is ensured to exist even when the functions are combinatorial and monotone.

The notion of EFX was first proposed by Caragiannis et al. [57]. Unlike EF1, EFX is hard to satisfy and the existence of EFX allocations is still unknown. For the case of goods, there are only results that show the existence of EFX allocations in some special cases. Plaut and Roughgarden [125] showed that EFX could be satisfied (1) when the utility functions are identical or combinatorial; (2) when the utility functions are IDO additive; (3) when there are two agents. Chaudhury et al. [60] and Amanatidis et al. [7] respectively extended the existence of EFX allocations to the cases (1) when there are three agents and (2) when the utility functions are bi-valued. Due to the hardness of finding exact EFX allocations, researchers relaxed the restriction by allowing a small number of items to be donated to a charity. Caragiannis et al. [54] showed the existence of an EFX partial allocation that achieves half the maximum Nash social welfare (i.e., the product of all agents' utilities). Chaudhury et al. [61] designed a pseudo-polynomial time algorithm that computes an EFX partial allocation where the charity receives no more than $n-1$ items and no agent envies the charity (n is the number of agents). Berger et al. [36] further improved this result by showing the existence of an EFX partial allocation with at most one unallocated item for $n = 4$ and $n-2$ unallocated items for $n \geq 5$. Researchers also studied the multiplicative approximations of EFX. Plaut and Roughgarden [125] showed that a 0.5-EFX allocation exists for every instance even with subadditive utility functions. Amanatidis et al. [11] further improved the approximation ratio to 0.618 for additive utility functions by proposing a polynomial-time algorithm. However, much less attention has been paid to the parallel problem of chores. The existence of EFX allocations is known for only a few special instances, e.g., IDO instances [114] and leveled preference instances [82]. For general instances, only $O(n^2)$-approximate EFX allocations are known to exist [145]. The existence of EFX allocations is still unknown even for simple cases with $n = 3$ or bi-valued disutility functions.

Relaxations of Proportionality One extensively studied relaxation of proportionality is MaxMinShare (MMS), which was proposed by Budish [53] for indivisible goods. When it comes to indivisible chores, the notion becomes MinMaxShare, which is also abbreviated to MMS for consistency. Intuitively, MaxMinShare is motivated by an imaginary experiment where an agent is to divide all items into n bundles but is the last one to select a bundle. The agent's best strategy, in the worst case, is to maximize the minimum utility of all bundles, and this utility is named her maximin share. Then an MMS allocation is defined so that every agent's utility is no smaller than her MMS. Researchers were optimistic about the existence of MMS allocations at the beginning. Bouveret and Lemaître [45] proved that if the utility functions are additive, envy-free allocations are also MMS fair. Kurokawa et al. [111] showed that MMS allocations exist with high probabilities through running random experiments. However, for the allocation of goods, it was first shown by Kurokawa et al. [111, 112] that there are instances where no allocation is MMS fair for all agents. Accordingly, considerable effort was devoted to designing (efficient) algorithms to compute approximately MMS fair allocations. Kurokawa et al. [112], Procaccia and Wang [127] proved there exists a 2/3-approximate MMS fair allocation for additive utilities, and then Amanatidis et al. [10] designed a polynomial-time algorithm with the same approximation guarantee. Later, Ghodsi et al. [87] improved the approximation ratio to 3/4, and Garg and Taki [86] further improved it to $3/4 + o(1)$. On the negative side, Feige et al. [79] proved that no algorithm can ensure better than 39/40 approximation. Beyond additive utilities, Barman and Krishnamurthy [31] initiated the study of approximate MMS fair allocation with submodular utilities, and proved that a 0.21-approximate MMS fair allocation can be computed by the round-robin algorithm. Ghodsi et al. [88] improved the approximation ratio to 1/3, and moreover, they gave constant and logarithmic approximation guarantees for XOS and subadditive utilities, respectively. The approximations for XOS and subadditive utilities are recently improved by Seddighin and Seddighin [131]. As we have seen, the majority of effort on finding MMS fair allocations is devoted to indivisible goods, but the parallel problem of chores has received less attention. Aziz et al. [21]

first pointed out this issue, and proved that the round-robin algorithm ensures 2-approximation for additive disutilities. Barman and Krishnamurthy [31] and Huang and Lu [104] respectively improved the approximation ratio to 4/3 and 11/9. Recently, Feige et al. [79] proved that with additive disutilities, no algorithm can be better than 44/43-approximate.

MaxMinShare bears some disadvantages. On the one hand, the definition is not trivial and computing its value involves solving an NP-hard problem. On the other hand, as we have seen, the MaxMinShare is not a feasible guarantee in some cases. Back to 1980s, Hill [100] also investigated how the indivisibility of the items affect the agent's guaranteed share by restricting attention to additive utility functions v such that $v(M) = 1$ (without loss of generality) and the most valuable item of v is worth α, $0 < \alpha < 1$; we write $\mathcal{V}(\alpha)$ for this subdomain of additive utility functions. Hill proposed to study the worst-case MaxMinShare among all utilities in $\mathcal{V}(\alpha)$, which is referred to as the *Hill's share*. In [100], Hill computed for every $n \geq 2$ a function $V_n : [0,1] \to [0, \frac{1}{n}]$, which lower-bounds Hill's share. By definition, $V_n(\alpha)$ is also a lower bound on the MaxMinShare of every utility in $\mathcal{V}(\alpha)$. Depending on α the guarantee $V_n(\alpha)$ may or may not improve upon the $\frac{3}{4}$-approximate MaxMinShare guarantee, but its great advantage is that whether a given allocation meets the guarantee for a given utility is immediately verifiable. Furthermore, Hill proved that if every agent's utility is in $\mathcal{V}(\alpha)$, it is always possible to simultaneously give each agent a share worth at least $V_n(\alpha)$, i.e., $V_n(\cdot)$ is a guarantee. Markakis and Psomas [117] proved a stronger result: the share $V_n(\alpha_i)$ where $\alpha_i = \max_{e \in M} v_i(e)$ is a bona fide guarantee over the full domain of additive and nonnegative utilities. Moreover, an allocation implementing these individual guarantees can be computed in polynomial time. Gourvès et al. [93] found that $V_n(\alpha)$ is not the tight characterisation of Hill's share and proved a tighter function. An interesting fact is that the tight function is not monotone in α, whereas its exact computation is still open.

Two other well-studied relaxations of proportionality are proportionality up to one item (PROP1) and proportionality up to any item (PROPX), which resemble the notions of EF1 and EFX. PROP1 allocations are known to always exist and can be computed in polynomial time for indivisible goods [64, 30], chores [51] and a mixed manna [20]. PROPX allocations always exist and can be computed efficiently for indivisible chores [123, 114]; nevertheless, they may not exist for goods [123, 20]. Another relaxation between PROP1 and PROPX is proportionality up to the maximin item (PROPm) [25, 26]. It was proven by Baklanov et al. [26] that a PROPm allocation can be computed in polynomial time.

Other relaxations of proportionality include pairwise MMS (PMMS) [57] and groupwise MMS (GMMS) [28]. PMMS requires that the allocation is MMS fair for any reduced instance with any two agents and GMMS requires the MMS fairness for any reduced instance with any group of agents. For PMMS, the best known approximation ratio is 0.781 by Kurokawa [110]. For GMMS, the best known ratio is 4/7 by Amanatidis et al. [11] and Chaudhury et al. [61]. A detailed comparison of these fairness notions was provided by Amanatidis et al. [9].

Efficiency Finding allocations that simultaneously satisfy fairness and efficiency is also an important branch of fair allocation. Computing a fair allocation that maximizes the utilitarian social welfare was proved to be NP-hard by Barman and Krishnamurthy [30]. Thus a lot of effort was devoted to studying the weaker efficiency notion—Pareto optimality (PO). Caragiannis et al. [57] first showed that maximized Nash social welfare implies EF1 and PO. Subsequently, Barman et al. [32] proposed a pseudo-polynomial time algorithm that computes EF1 and PO allocations. Barman and Krishnamurthy [30] designed a polynomial-time algorithm that computes PROP1 and PO allocations. Amanatidis et al. [7] showed for bi-valued utility functions that the maximized Nash social welfare implies EFX and PO, which was then improved by Garg and Murhekar [85] by proposing a polynomial-time algorithm. Garg and Murhekar [85] also proved that EFX and PO allocations may not exist for utility functions with three different values. In contrast, Hosseini et al. [103] designed a polynomial-time algorithm that computes EFX and PO allocations for instances with lexicographic utility functions. For the parallel problem of chores, Aziz et al. [20] proposed an algorithm that computes PROP1 and PO allocations in polynomial time, even for a mixture of goods and chores. However, regarding EF1 and PROPX, very little is known about their compatibility with PO.

1.2.3 More Complicated Settings

In the past few years, more and more attention has been paid to more complicated but realistic settings, e.g., with constraints, partial information, online setting, mixture of goods and chores, with subsidies, weighted agents. These novel settings combine the characteristics of many real-life scenarios and bring theory closer to the needs of real-life applications.

With Constraints In some scenarios, there are constraints on items such that some items cannot be allocated together. For example, when allocating courses to students, it is not desired to allocate too many courses within the same discipline to a student. Considerable effort in the literature was concentrated on the graphical constraint where the items are vertices of an undirected graph and only those that form a connected sub-graph can be allocated together [43, 116, 33, 38, 44, 138, 120, 106]. The budget constraint was also widely studied [144, 84, 126, 65, 29], in which setting the items have sizes and the agents have budgets and the total size of items allocated to one agent cannot exceed the agent's budget. Kyropoulou et al. [113], Biswas and Barman [39], Hummel and Hetland [105] also studied the cardinality constraint where the items are categorized into types and the number of items in each type that can allocated together is limited. Other broadly-studied constraints include matroid constraint [67, 92, 91], geometric constraint [132, 133, 74], separation constraint [76, 75], etc. We refer the readers to the survey by Suksompong [139] for more detailed summary of works on fair allocation with constraints.

Partial Information Sometimes, we need to allocate items when only partial information is given. For example, when a wide heterogeneous population of users are involved, it becomes challenging to elicit and aggregate every user's cardinal values. For goods, it was proved by Amanatidis et al. [8] and Halpern and Shah [96] that the best approximation ratio of MMS is $\Omega(\log n)$ when only ordinal preferences are known. For chores, Aziz et al.

[18] proved constant lower bounds and upper bounds of the approximation ratio. Recently, Hosseini et al. [101, 102] studied an ordinal approximation of MMS for goods and chores, which is more robust to the cardinal version of MMS. Another interesting scenario is when valuations are unknown and we need to quantify the number of queries on the valuations for the algorithm to compute a fair allocation. Oh et al. [124] proved that $\Theta(\log m)$ queries are sufficient to compute EF1 allocations.

Online Setting In some scenarios, fair allocation problems are online, where items or agents or both are coming in an online fashion, and allocation decisions must be made immediately when they come which usually cannot be revoked. Consider allocating organs to patients, the organs need to be allocated immediately when they are donated. Also consider allocating charging slots to cars, the slots need to be allocated as soon as cars come. Different dimensions of the online setting problems were considered in the literature. One dimension is whether the items are divisible [142, 107, 108] or indivisible [1, 4, 118, 2, 3]. Another dimension is which element is online in the problem, i.e., only the items are online [1], only the agents are online [143], or both are online [119]. Different research topics of this setting were also studied. Many algorithms were proposed to deal with the challenges brought by the online nature [1, 5]. Researchers also cared about the properties that could be guaranteed in the online setting, e.g., strategy-proof which means that agents cannot improve their utilities by manipulating [5]. One interesting topic is to quantify the number of adjustments to the allocation to restore desired properties [98]. For more detailed information on online fair allocation, we refer the readers to a comprehensive survey by Aleksandrov and Walsh [6].

Mixture of Goods and Chores Sometimes, the items to be allocated contain both goods and chores. One key characteristic of this setting is that the valuations are not monotone [41, 40]. Aziz et al. [16] designed an algorithm that computes an EF1 allocation for any number of agents. For the special case with only two agents, they designed another algorithm that guarantees EF1 and PO simultaneously. But it still remains an open question where EF1 and PO can be satisfied together for any number of agents. For other fairness notions, Aziz et al. [20] proposed an algorithm that computes PROP1 plus PO allocations. Kulkarni et al. [109] designed an algorithm that guarantees approximately MMS plus PO together.

With Subsidies As the desired fairness notions like envy-freeness and proportionality can rarely be satisfied, one interesting research direction is to compensate agents with subsidies (or money) so that those fairness notions can be restored, whose idea can trace back to the rent division problem in the economics literature [73]. Halpern and Shah [95] quantified the amount of the external subsidies when the marginal value of each item is at most one for each agent. Brustle et al. [52] proved that one unit of subsidies per agent is sufficient to achieve envy-freeness. Caragiannis and Ioannidis [55] improved the results by quantifying the minimum subsidies to compute envy-free allocations. One general extension of this setting is the allocation of a mixture of indivisible and divisible items [34, 35], where the divisible items can be regarded as heterogeneous subsidies.

Weighted Agents Researchers also care about the setting where agents have unequal rights which models some real-life scenarios. For example, people at higher positions should take more responsibilities. For goods, Farhadi et al. [78] gave an approximation ratio of $\Theta(n)$ for weighted MMS. Chakraborty et al. [58] proved that weighted EF1 allocations always exist. Nevertheless, for chores, few works have been done except that Aziz et al. [17] studied the weighted MMS. Other fairness notions including l-out-of-d MMS were also introduced and studied in the weighted setting [22, 24].

1.3 Our Problems and Results

1.3.1 The MinMaxShare

Motivation

As we have seen, the majority of effort on finding MMS fair allocations for indivisible chores is devoted to additive disutility functions. However, very little is known beyond additivity, which motivates our first work:

> *We study the MMS fair allocation of indivisible chores with non-additive disutility functions.*

Main Results

We first show that no algorithm can ensure better than $\Omega(\min\{n, \frac{\log m}{\log \log m}\})$ approximation when the disutility functions are submodular[1], which is a sharp contrast to the allocation of goods. Further, we show that for general subadditive disutility functions, there always exists an allocation that

[1] In this work we use $\log(\cdot)$ to denote $\log_2(\cdot)$.

is $O(\min\{n, \log m\})$-approximate MMS, and thus the approximation ratio is asymptotically tight. Next, we consider the ordinal relaxation 1-out-of-d MMS. It is trivial that 1-out-of-1 MMS is satisfied no matter how the chores are allocated, and somewhat surprisingly, our impossibility result implies that for any $d \geq 2$, there is an instance for which no allocation is 1-out-of-d MMS.

Result 1 For general subadditive and submodular disutility functions, the tight multiplicative approximation ratio of MMS is $\tilde{\Theta}(\min\{n, \log m\})$. Further, for any $d \geq 2$, a 1-out-of-d MMS allocation may not exist.

Result 1 combines Theorems 1, 2 and Corollary 1. The strong impossibility in Result 1 does not rule out the possibility of constant multiplicative or ordinal approximation of MMS fair allocation for all subadditive disutilities. We then turn to study two concrete settings that have shown successful real-world applications. The first setting deals with a job scheduling problem, where a set of jobs need to be processed by the agents. The agents are heterogeneous and thus each job may be of different lengths to different agents. Each agent controls a set of machines with possibly different speeds. Upon receiving a set of jobs, an agent's disutility is determined by the corresponding minimum completion time when processing the jobs using her own machines (i.e., makespan). As will be clear, job scheduling is a more general setting than additive disutilities, which uncovers new research directions for group-wise fairness. Scheduling problems appear in many research areas, including data science, big data, high-performance computing, and cloud computing [97].

The second setting deals with a bin packing problem, where the chores have sizes which can be different to different agents. The agents have bins that can be used to pack the chores allocated to them with the goal of using as few bins as possible. Semiconductor chip design, loading vehicles with weight capacity limits, and filling containers are all examples of the bin packing problem [63].

Result 2 For the job scheduling setting, a 2-approximate MMS allocation can be computed in polynomial time, and a 1-out-of $\lfloor \frac{n}{2} \rfloor$ MMS allocation always exists. For the bin packing setting, a 2-approximate MMS allocation and a 1-out-of-$\lfloor \frac{n}{2} \rfloor$ MMS allocation can be computed in polynomial time.

Result 2 combines Corollaries 2, 3 and Theorems 3, 4.

Besides the study of MMS allocations, we also provide a detailed discussion about two other relaxations of proportionality, i.e., PROP1 and PROPX.

1.3.2 The Hill's Share

Motivation

We also notice that all the aforementioned works on Hill's share focus on the allocation of goods, and the mirror problem of chores is not as well understood as that of goods, which motivates our second work:

> *We apply Hill's approach to the allocation of indivisible chores and prove a set of results parallel to those for indivisible goods.*

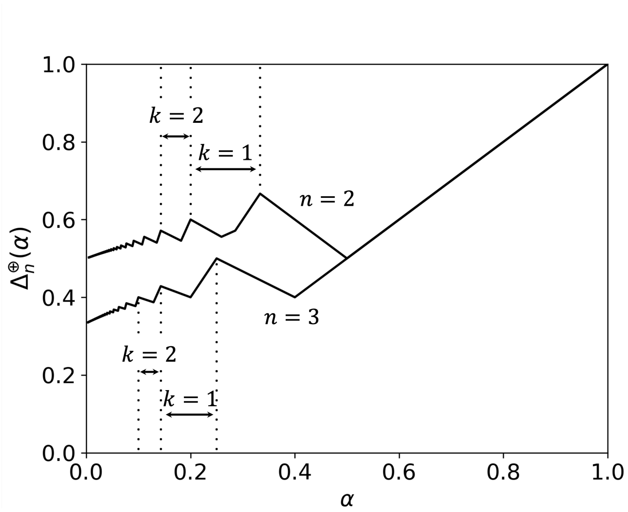


Figure 1.1: Hill's share $\Delta_n^{\oplus}(\alpha)$ when $n = 2$ and 3 and m is not restricted.

Main Results

We first compute the tight characterisation of Hill's share, refined to problems with a given number m of chores, i.e., the exact upper bound $\Delta_n^{\oplus}(\alpha; m)$ of the MinMaxShare in the domain $\mathcal{V}(\alpha; m)$, where $\mathcal{V}(\alpha; m)$ contains the disutility functions over m items with the highest disutility being α. This result is stated in Theorem 6. If m is not restricted, i.e., $\mathcal{V}(\alpha) = \bigcup_m \mathcal{V}(\alpha; m)$ and $\Delta_n^{\oplus}(\alpha) = \max_m \Delta_n^{\oplus}(\alpha; m)$, we illustrate the function $\Delta_n^{\oplus}(\alpha)$ for $n = 2, 3$ in Fig. 1.1. Just like Gourvès et al. [93] observed for the problem of goods, this function is not monotone in α. In passing, we tighten the bounds proposed by Hill [100] and Gourvès et al. [93] for the worst-case MaxMinShare in the two-agent problem of goods; see Remark 1.

Compared to the MinMaxShare, Hill's share $\Delta_n^{\oplus}(\alpha; m)$ is immediately verifiable, whereas deciding whether (a multiple of) the MinMaxShare is met at a given allocation involves solving an NP-hard problem. Moreover, the

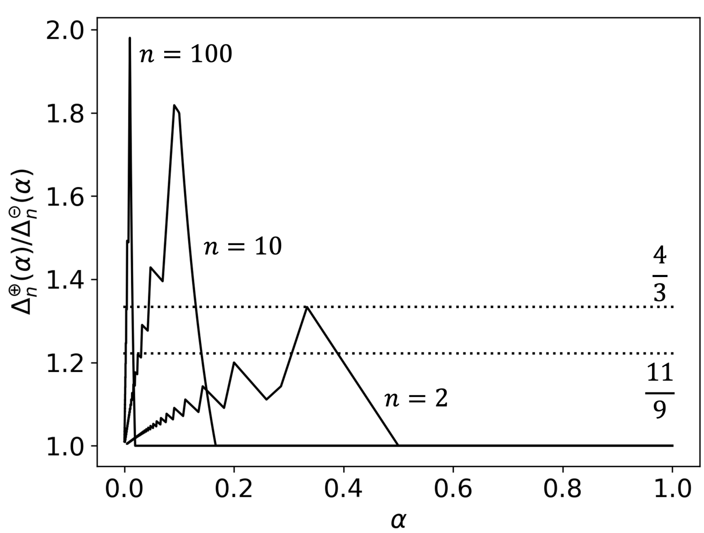


Figure 1.2: The ratio between the upper and lower bounds of the MinMaxShare of disutilities in $\mathcal{V}(\alpha)$. 4/3 and 11/9 are two fractions of the MinMaxShare known to be achievable.

function $\alpha \to \Delta_n^{\oplus}(\alpha; m)$ relating the guaranteed share to the disutility of the worst chore (relative to total disutility) is a transparent hard design constraint of which all participants should be aware. Although $\Delta_n^{\oplus}(\alpha; m)$ seems more conservative than the MinMaxShare of a specific disutility function, we argue that $\Delta_n^{\oplus}(\alpha; m)$ is approximately as effective as MinMaxShare. First, $\Delta_n^{\oplus}(\alpha; m)$ is at most twice the MinMaxShare of every disutility in $\mathcal{V}(\alpha; m)$. We plot the exact ratio of $\Delta_n^{\oplus}(\alpha)$ and the best MinMaxShare of disutilities in $\mathcal{V}(\alpha)$ for every α in Fig. 1.2 when $n = 2, 10$ and 100. As we can see, although the largest ratio may reach 2 (only happens when n is large), for most values of α, the ratio is not far from 1. In particular, $\Delta_n^{\oplus}(\alpha)$ outperforms the fractions of the MinMaxShare known to be implementable ($\frac{4}{3}$ by Barman and Krishnamurthy [31] and $\frac{11}{9}$ by Huang and Lu [104]) for most α

no matter what values n has. Besides the above worst-case comparison, in Section 4.3, we conduct numerical experiments with synthetic and real-world data to illustrate the real distances between Hill's share and MinMaxShare. The experiments show that Hill's share is actually very close to (e.g., within 1.1 fraction of) the MinMaxShare for the majority of the instances.

Finally, we obtain the main result of this work – a counterpart for chores of Hill's guarantee for goods improved by Markakis and Psomas [117]. Letting $V_n(\alpha; m)$ denote the monotonic cover of $\Delta_n^{\oplus}(\alpha; m)$ with respect to α, Theorem 7 shows that the share $V_n(\alpha_i; m)$ is a guarantee over the full domain of additive disutilities with m chores. We also provide an algorithm to implement this guarantee in polynomial time. To the best of our knowledge no other similarly simple guarantee for allocating chores has been identified.

1.4 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, we introduce some necessary preliminaries such as the definitions and notations used in this thesis. In Chapter 3, we present our work about fair allocation of indivisible chore with beyond additive disutilities. Specifically, in Section 3.1, we first design an instance with submodular disutilities where no allocation can be better than n-MMS and 1-out-of-d MMS for any $d \geq 2$. We then propose an algorithm that computes a $\min\{n, \lceil \log m \rceil\}$-MMS allocation for any instance with subadditive disutilities; in Section 3.2, we first introduce the job scheduling model and then elaborate on the algorithms that compute a 1-out-of-$\lfloor \frac{n}{2} \rfloor$ MMS for any job scheduling instance. We also show how to modify the algorithms to compute a 2-MMS allocation in polynomial time; in Section 3.3, we first introduce the bin packing model and then present the polynomial-time algorithms that compute a 1-out-of-$\lfloor \frac{n}{2} \rfloor$ MMS for any bin packing instance. Besides, we show that a slight modification to the algorithms gives us a 2-MMS allocation. Moreover, we show the multiplicative ratio is actually tight by presenting an instance where no allocation is better than 2-MMS. in Section 3.4, we provide a detail discussion on two other relaxations of proportionality, i.e., PROP1 and PROPX. In Chaper 4, we present our work about Hill's worst-case guarantee for indivisible chores. Specifically, in Section 4.1, we compute the tight characterisation of Hill's share; in Section 4.2, we show that the monotonic cover of Hill's share is a guarantee over the full domain of additive disutilities and design an algorithm that implements the guarantee in polynomial time; in Section 4.3, we conduct various experiments to demonstrate that Hill's share can serve as a good alternative of the MinMaxShare. Finally, in Chapter 5, we make a conclusion and provide many promising future directions that extend the works in this thesis.

CHAPTER 2

PRELIMINARIES

In this chapter, we introduce the necessary preliminaries such as some notations and definitions, which will be used when we present the details of our works in the following chapters.

For any integer $k \geq 1$, let $[k] = \{1, \ldots, k\}$. In a fair allocation instance $I = (N, M, \{v_i\}_{i \in N})$, there are n agents denoted by $N = [n]$ and m chores denoted by $M = \{e_1, \ldots, e_m\}$. Each agent i has a disutility function over the chores, $v_i : 2^M \to \mathbb{R}^+ \cup \{0\}$. For simplicity, we abuse $v_i(\cdot)$ to denote a disutility function and write $v(e)$ to represent $v(\{e\})$ for each $e \in M$. The disutility functions satisfy $v_i(\emptyset) = 0$ and $v_i(S_1) \leq v_i(S_2)$ for any $S_1 \subseteq S_2 \subseteq M$. A disutility function v_i is subadditive if for any $S_1, S_2 \subseteq M$,

$$v_i(S_1 \cup S_2) \leq v_i(S_1) + v_i(S_2).$$

It is submodular if for any $S_1 \subseteq S_2 \subseteq M$ and $e \in M \setminus S_2$,

$$v_i(S_2 \cup \{e\}) - v_i(S_2) \leq v_i(S_1 \cup \{e\}) - v_i(S_1).$$

It is additive if for any $S \subseteq M$,

$$v_i(S) = \sum_{e \in S} v_i(e).$$

It is widely known that any additive function is also submodular, and any submodular function is also subadditive.

An allocation $\mathbf{A} = (A_1, \ldots, A_n)$ is an n-partition of the chores where A_i contains the chores allocated to agent i such that $A_i \cap A_j = \emptyset$ for any $i \neq j$ and $\bigcup_{i \in N} A_i = M$. For any set S and integer d, let $\Pi_d(S)$ be the set of all d-partitions of S. The MinMaxShare (MMS) of agent i is

$$\mathsf{MMS}_i^n(I) = \min_{(X_1,\ldots,X_n) \in \Pi_n(M)} \max_{j \in N} v_i(X_j).$$

We may neglect n and I in $\mathsf{MMS}_i^n(I)$ when there is no ambiguity. Note that the computation of MMS_i is NP-hard even when the disutilities are additive, which can be verified by a reduction from the Partition problem. Given an n-partition of M, $\mathbf{X} = (X_1, \ldots, X_n)$, if $v_i(X_j) \leq \mathsf{MMS}_i$ for any $j \in N$, then $\mathbf{X}$ is called an *MMS-defining partition* for agent i. Note that the original definition of MMS_i for chores is defined with non-positive values, where the minimum valued bundle is maximized. In this thesis, to simplify the notions, we choose to use non-negative numbers (representing disutiltities).

Definition 1 **(α-MMS)** *An allocation* $\mathbf{A} = (A_1, \ldots, A_n)$ *is* α*-approximate MinMaxShare (α-MMS) fair if* $v_i(A_i) \le \alpha \cdot \mathsf{MMS}_i$ *for all* $i \in N$. *The allocation is MMS fair if* $\alpha = 1$.

Given the definition of MMS, for any agent i with subadditive disutility $v_i(\cdot)$, we have the following simple bounds for MMS_i,

$$\mathsf{MMS}_i \ge \max\big\{ \max_{e \in M} v_i(e), \frac{1}{n} \cdot v_i(M) \big\}. \tag{2.1}$$

Following recent works [23, 102, 101], we also consider the ordinal approximation of MMS, namely, 1-out-of-d MMS fairness. Intuitively, MMS fairness can be regarded as 1-out-of-n MMS (i.e., partitioning the chores into n bundles but receiving the largest bundle). Since 1-out-of-n MMS allocations may not exist, we can instead find a maximum integer $d \le n$ such that an 1-out-of-d MMS allocation is guaranteed to exist. Formally, an allocation $\mathbf{A}$ is 1*-out-of-d MMS* fair if for every agent $i \in N$, $v_i(A_i) \le \mathsf{MMS}_i^d$. More generally, given any $\alpha \ge 1$, we have the bi-factor approximation, α-approximate 1-out-of-d MMS, if $v_i(A_i) \le \alpha \cdot \mathsf{MMS}_i^d$ for every $i \in N$. By the definition, we have the following simple observation.

Observation 1 *Given* $1 \le d \le n$, *any 1-out-of-d MMS allocation is* $\lceil \frac{n}{d} \rceil$*-MMS fair.*

Proof. To prove the observation, it suffices to show $\mathsf{MMS}_i^d \le \lceil \frac{n}{d} \rceil \cdot \mathsf{MMS}_i^n$ for any agent $i \in N$. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be an MMS-defining partition for agent i, which satisfies $v_i(X_j) \le \mathsf{MMS}_i^n$ for any $j \in [n]$. Consider a d-partition $\mathbf{X}' = (X'_1, \ldots, X'_d)$ built by evenly distributing the n bundles in $\mathbf{X}$ to the d

bundles in $\mathbf{X}'$; that is, the number of bundles distributed to the bundles in $\mathbf{X}'$ differs by at most one. Clearly, $\mathbf{X}'$ satisfies $v_i(X'_j) \leq \lceil \frac{n}{d} \rceil \cdot \mathsf{MMS}_i^n$ for any $j \in [d]$. By the definition of MMS, it follows that

$$\mathsf{MMS}_i^d \leq \max_{j \in [d]} v_i(X'_j) \leq \lceil \frac{n}{d} \rceil \cdot \mathsf{MMS}_i^n,$$

thus completing the proof. ■

We let $\mathcal{A}dd(M)$ be the domain made of the nonnegative additive disutility functions v on chore set M, normalised without loss of generality, as follows

$$v(S) = \sum_{e \in S} v(e) \text{ for all } S \subseteq M \text{ and } v(M) = 1.$$

For any $\alpha \in [0, 1]$, the subdomain $\mathcal{V}(\alpha; m) \subseteq \mathcal{A}dd(M)$ is defined by the property $\max_{e \in M} v(e) = \alpha$ and $\mathcal{U}(\alpha; m)$ by $v(e) \leq \alpha$ for all $e \in M$. According to the definitions, $\mathcal{V}(\alpha; m) \subseteq \mathcal{U}(\alpha; m)$ for any valid pair of α and m. Note that, since the functions are all normalised, $\mathcal{V}(\alpha; m)$ is only well defined if $\alpha \times m \geq 1$, equivalently for $m \geq m_* = \lceil \frac{1}{\alpha} \rceil$ (the upper integer part of $\frac{1}{\alpha}$).

We next define the upper and lower bounds of MinMaxShare among all disutilities in $\mathcal{V}(\alpha; m)$,

$$\Delta_n^{\oplus}(\alpha; m) = \max_{v \in \mathcal{V}(\alpha; m)} \mathsf{MMS}_n(v); \text{ and}$$

$$\Delta_n^{\ominus}(\alpha; m) = \min_{v \in \mathcal{V}(\alpha; m)} \mathsf{MMS}_n(v).$$

The upper bound $\Delta_n^{\oplus}(\alpha; m)$ (i.e., the worst-case MinMaxShare) is called *Hill's share*, and we use these terms interchangeably in this thesis.

It is not difficult to obtain the below formula of $\Delta_n^{\ominus}(\alpha; m)$.

Lemma 1 *Given $0 < \alpha < 1$, $n \geq 2$, and $m \geq \lceil \frac{1}{\alpha} \rceil$, $\Delta_n^{\ominus}(\alpha; m)$ is as follows:*

$$\Delta_n^{\ominus}(\alpha; m) = \begin{cases} \alpha, & \text{if } \alpha > \frac{1}{n}, \\ \frac{1}{n}, & \text{if } \alpha = \frac{1}{kn}, \text{ or } \frac{1}{(k+1)n} < \alpha < \frac{1}{kn} \text{ and } m \geq kn + n \\ k\alpha + \frac{1-kn\alpha}{m-kn}, & \text{if } \frac{1}{(k+1)n} < \alpha < \frac{1}{kn} \text{ and } m \leq kn + n - 1 \end{cases}$$

for some integer $k \geq 1$.

Proof. For each case, we show that $\mathsf{MMS}_n(v) \geq \Delta_n^{\ominus}(\alpha; m)$ for any $v \in \mathcal{V}(\alpha; m)$, and design a disutility function such that the MinMaxShare is exactly $\Delta_n^{\ominus}(\alpha; m)$. By the definition of $\mathcal{V}(\alpha; m)$, there exists a chore with disutility α, thus $\mathsf{MMS}_n(v) \geq \alpha$ for any $v \in \mathcal{V}(\alpha; m)$. Moreover, when $\alpha > 1/n$, there exists a disutility function such that the MinMaxShare is exactly α. Specifically, v_1 contains $\lceil \frac{1}{\alpha} \rceil$ chores, $\lfloor \frac{1}{\alpha} \rfloor$ with disutility α and one with disutility $(1 - \lfloor \frac{1}{\alpha} \rfloor \cdot \alpha) < \alpha$ (if 1 is indivisible by α). $\mathsf{MMS}_n(v_1) = \alpha$ follows from the fact that v_1 contains at most n chores.

By the definition of MinMaxShare, $\mathsf{MMS}_n(v) \geq \frac{1}{n}$, where the equality is achieved when the total disutility of M can be evenly distributed among the n-partition. When $1/n$ is divisible by α (i.e., $\alpha = \frac{1}{kn}$ for some positive integer k), or $1/n$ is not divisible by α (i.e., $\frac{1}{(k+1)n} < \alpha < \frac{1}{kn}$) and the number of chores m is at least $kn + n$, there exists an disutility function such that the MinMaxShare is exactly $1/n$. For the former, the disutility function v_2 contains $1/\alpha = kn$ chores with disutility α. Clearly, each bundle in the best n-partition contains k chores with disutility α and $\mathsf{MMS}_n(v_2) = 1/n$. For the latter, intuitively, the total disutility of M can also be evenly distributed by letting each bundle contain $\lfloor \frac{1}{n\alpha} \rfloor = k$ chores with disutility α and one chore with disutility $\frac{1}{n} - k\alpha < \alpha$. In total, $kn + n \leq m$ chores are needed. Hence, the disutility function v_3 contains kn chores with disutility α, n chores with disutility $\frac{1}{n} - k\alpha$ and $m - kn - n$ chores with disutility 0, and $\mathsf{MMS}_n(v_3) = 1/n$.

However, when $1/n$ is indivisible by α but the number of chores m is limited to $kn + n - 1$, $1/n$ cannot be achieved since some bundles in any n-partition contain no more than k chores, and the disutilities of these bundles are at most $k\alpha < 1/n$. For this case, we show that $\mathsf{MMS}_n(v) \geq k\alpha + \frac{1-kn\alpha}{m-kn}$ for any $v \in \mathcal{V}(\alpha; m)$. Letting x be the number of bundles in the n-partition that contain no more than k chores, it follows that $x \geq kn + n - m$. Since the disutility of each of these bundles is at most $k\alpha$, the average disutility of the other bundles is at least

$$\frac{1 - k\alpha x}{n - x} \geq \frac{1 - (kn + n - m) \cdot k\alpha}{m - kn} = k\alpha + \frac{1 - kn\alpha}{m - kn} > k\alpha$$

where the leftmost-hand side is an increasing function of x since $k\alpha < 1/n$, and the last inequality is because $m \geq \lceil \frac{1}{\alpha} \rceil > kn$. Therefore, the largest disutility of any n-partition is at least $k\alpha + \frac{1-kn\alpha}{m-kn}$; that is, $\mathsf{MMS}_n(v) \geq k\alpha + \frac{1-kn\alpha}{m-kn}$ for any $v \in \mathcal{V}(\alpha; m)$. Let v_4 contain kn chores with disutility α and $m - kn$ chores with disutility $\frac{1-kn\alpha}{m-kn} < \alpha$. Clearly, the worst bundle in the best n-partition contains k chores with disutility α and one chore with disutility $\frac{1-kn\alpha}{m-kn}$, thus $\mathsf{MMS}_n(v_4) = k\alpha + \frac{1-kn\alpha}{m-kn}$. ■

Computing $\Delta_n^{\oplus}(\alpha; m)$ is non-trivial, as shown in Section 4.1, but the following lemma presents two simple properties.

Lemma 2 *(1) $\Delta_n^{\oplus}(\alpha; m)$ is weakly decreasing in n; (2) $\Delta_n^{\oplus}(\alpha; m)$ is weakly* *increasing in m from $\lceil \frac{1}{\alpha} \rceil$ to $\lceil \frac{2}{\alpha} \rceil - 1$ and constant thereafter.*

Proof. That $\Delta_n^{\oplus}(\alpha; m)$ decreases in n is clear by comparing the MinMaxShares of an arbitrary n-partition and the $(n+1)$-partition obtained by adding one empty share. The monotonicity in m (i.e., $\Delta_n^{\oplus}(\alpha; m) \leq \Delta_n^{\oplus}(\alpha; m+1)$) follows that every disutility in $\mathcal{V}(\alpha; m)$ can be transformed to one in $\mathcal{V}(\alpha; m+1)$ by adding a chore with disutility 0, without changing the MinMaxShare.

We then show when $m \geq \lceil \frac{2}{\alpha} \rceil - 1$, $\Delta_n^{\oplus}(\alpha; m) \geq \Delta_n^{\oplus}(\alpha; m+1)$, thus $\Delta_n^{\oplus}(\alpha; m)$ remains constant. To achieve this, we first claim that when $m \geq \lceil \frac{2}{\alpha} \rceil - 1$, for any $v \in \mathcal{V}(\alpha; m+1)$ and any allocation $(A_1, \ldots, A_n)$, there exists one bundle such that the total disutility of two of its chores is no more than α. Otherwise, for any bundle A_k, the total disutility of any two chores is larger than α, which means that $v(A_k) > \frac{|A_k|}{2} \cdot \alpha$. Upon summing up the lower bounds over all bundles, $1 = \sum_{k \in N} v(A_k) > \frac{\alpha}{2} \cdot \lceil \frac{2}{\alpha} \rceil \geq 1$, a contradiction.

Now we pick any disutility $v \in \mathcal{V}(\alpha; m+1)$, and let $(A_1, \ldots, A_n)$ be the allocation that gives the MinMaxShare of v. By the claim, there exists a bundle (w.l.o.g., A_1) such that two chores $e_1, e_2 \in A_1$ satisfy $v(e_1)+v(e_2) \leq \alpha$. We derive a disutility $v' \in \mathcal{V}(\alpha; m)$ by merging e_1 and e_2 into one chore e, and show that $\mathsf{MMS}_n(v) = \mathsf{MMS}_n(v')$. On one hand, letting $A_1' = A_1 \setminus \{e_1, e_2\} \cup \{e\}$, it follows that $\mathsf{MMS}_n(v) \geq \mathsf{MMS}_n(v')$ since $(A_1', \ldots, A_n)$ is an allocation regarding v' with the largest disutility being $\mathsf{MMS}_n(v)$. On the other hand, by decomposing e into e_1 and e_2, we can convert any allocation regarding v' to an allocation regarding v without changing the largest disutility, thus $\mathsf{MMS}_n(v) \leq \mathsf{MMS}_n(v')$.

Therefore, when $m \geq \lceil \frac{2}{\alpha} \rceil - 1$, every disutility in $\mathcal{V}(\alpha; m+1)$ can be transformed to one in $\mathcal{V}(\alpha; m)$ without changing the MinMaxShare, which gives $\Delta_n^{\oplus}(\alpha; m) \geq \Delta_n^{\oplus}(\alpha; m+1)$. Combining with the monotonicity in m, $\Delta_n^{\oplus}(\alpha; m)$ remains constant when $m \geq \lceil \frac{2}{\alpha} \rceil - 1$. ■

By the second property in Lemma 2, and also following [100, 117, 93], we also consider the case when m is not restricted, or equivalently, $m = \infty$. Let $\mathcal{V}(\alpha) = \bigcup_m \mathcal{V}(\alpha, m)$ and $\mathcal{U}(\alpha) = \bigcup_m \mathcal{U}(\alpha; m)$. Accordingly, we have

$$\Delta_n^{\oplus}(\alpha) = \max_{v \in \mathcal{V}(\alpha)} \mathsf{MMS}_n(v); \text{ and}$$

$$\Delta_n^{\ominus}(\alpha) = \min_{v \in \mathcal{V}(\alpha)} \mathsf{MMS}_n(v).$$

By Lemma 1, $\Delta_n^{\ominus}(\alpha) = \max\{\alpha, 1/n\}$.

Hill's share $\Delta_n^{\oplus}(\alpha; m)$ (and $\Delta_n^{\oplus}(\alpha)$) behave much like the MinMaxShare in the following senses. First, for any $v \in \mathcal{V}(\alpha; m)$ there is an allocation $(A_1, \ldots, A_n)$ such that $v(A_i) \leq \Delta_n^{\oplus}(\alpha; m)$ for all i. This follows from the definition of the MinMaxShare plus that $\Delta_n^{\oplus}(\alpha; m)$ is an upper bound of the MinMaxShare. Second, the *max* in the definition of $\Delta_n^{\oplus}(\alpha; m)$ is achieved by some $v^* \in \mathcal{V}(\alpha; m)$; that is, $\Delta_n^{\oplus}(\alpha; m) = \mathsf{MMS}_n(v^*)$. This is because $\mathcal{V}(\alpha; m)$ is a compact set and all functions are continuous. Then we know that for any allocation $(B_1, \ldots, B_n)$ there is some i such that $v^*(B_i) \geq \Delta_n^{\oplus}(\alpha; m)$. Note that these two facts have nothing to do with what the function $\Delta_n^{\oplus}(\alpha; m)$ actually looks like and they can be easily adapted to $\Delta_n^{\oplus}(\alpha)$.

CHAPTER 3

MMS ALLOCATION OF INDIVISIBLE CHORES: BEYOND ADDITIVE DISUTILITY FUNCTIONS

In this chapter, we elaborate on our work about MMS allocation of indivisible chores when the disutility functions are beyond additivity. We first focus on the setting with general subadditive disutilities. A lower-bound instance and an algorithm are designed to show the tightness of our approximation ratio of MMS. We then turn to the job scheduling and the bin packing settings. Algorithms that compute constant approximate MMS fair allocations are designed for these two specific settings. We end this chapter by providing a detailed discussion about two other relaxations of proportionality.

3.1 Subadditve Disutilities

By Inequality 2.1, if the disutilities are subadditive, allocating all chores to a single agent ensures an approximation of n, which is somehow the most unfair algorithm. Surprisingly, such an unfair algorithm achieves the optimal approximation ratio of MMS even if the disutilities are submodular.

Theorem 1 *For any $n \geq 2$, there is an instance with submodular disutilities for which no allocation is better than n-MMS or $\frac{\log m}{\log \log m}$-MMS.*

Proof. Since

$$n = \frac{\log m}{\log \log m - \log \log \log m + o(1)} \geq \frac{\log m}{\log \log m},$$

in the following, it suffices to show no allocation can be better than n-MMS. For any fixed $n \geq 2$, we construct the following instance with n agents and $m = n^n$ chores. Let each chore correspond to a point in an n-dimensional coordinate system, i.e.,

$$M = \{(x_1, x_2, \ldots, x_n) \mid x_i \in [n] \text{ for all } i \in [n]\}.$$

For each agent $i \in N$, we define n covering planes $\{C_{il}\}_{l \in [n]}$ and for each $l \in [n]$,

$$C_{il} = \{(x_1, x_2, \ldots, x_n) \mid x_i = l \text{ and } x_j \in [n] \text{ for all } j \in [n] \setminus \{i\}\}.$$

Note that $\{C_{il}\}_{l \in [n]}$ forms an exact cover of the points in M, i.e., $\bigcup_l C_{il} = M$

and $C_{il} \cap C_{iz} = \emptyset$ for all $l \neq z$. For any set of chores $S \subseteq M$, $v_i(S)$ equals the minimum number of planes in $\{C_{il}\}_{l\in[n]}$ that can cover S. Therefore, $v_i(S) \in [n]$ for all S. We first show $v_i(\cdot)$ is submodular for every i. For any $S \subseteq T \subseteq M$ and any $e \in M \setminus T$, if e is not in the same covering plane with any point in T, e is not in the same covering plane with any point in S, either. Thus, $v_i(T \cup \{e\}) - v_i(T) = 1$ implies $v_i(S \cup \{e\}) - v_i(S) = 1$, and accordingly,

$$v_i(T \cup \{e\}) - v_i(T) \leq v_i(S \cup \{e\}) - v_i(S).$$

Since $\{C_{il}\}_{l\in[n]}$ is an exact cover of M, $\mathsf{MMS}_i = 1$ for every i, where the MMS-defining partition is simply $\{C_{il}\}_{l\in[n]}$. Then to prove the theorem, it suffices to show that for any allocation of M, there is at least one agent whose disutility is n. For the sake of contradiction, we assume there is an allocation $\mathbf{X} = (X_1, \ldots, X_n)$ where every agent has disutility at most $n-1$. This means that for every $i \in [n]$, there exists a plane C_{il_i} such that $X_i \cap C_{il_i} = \emptyset$. Consider the point $\mathbf{b} = (l_1, \ldots, l_n)$, it is clear that $\mathbf{b} \in C_{il_i}$ and thus $\mathbf{b} \notin X_i$ for all i. Hence $\mathbf{b}$ is not allocated to any agent, which is a contradiction to $\mathbf{X}$ being an allocation. ■

To facilitate the understanding of Theorem 1, we provide an instance with 3 agents and 27 chores where no allocation is better than 3-MMS. The instance is illustrated in Figure 3.1, where each agent has three covering planes. Take agent 1 for example, her three covering planes contain the chores whose x coordinates are 1, 2, 3, respectively. If there exists an allocation that is better than 3-MMS, then each agent is allocated chores from at most 2 of her covering planes. Without loss of generality, we assume that agent 1 (or

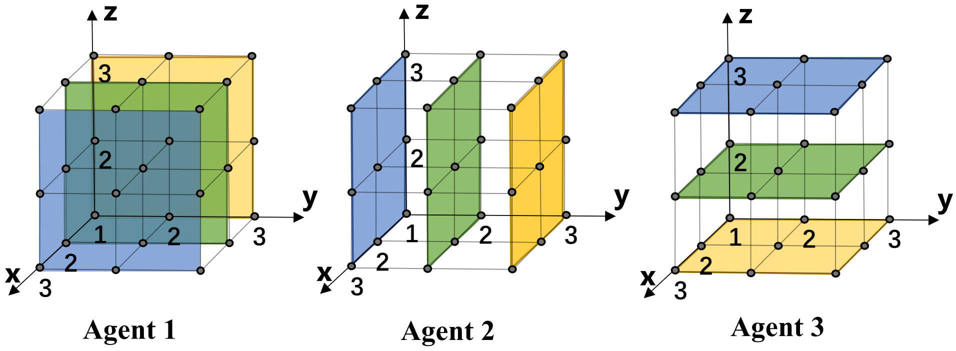


Figure 3.1: An instance with 3 agents and 27 chores

agent 2 and 3 respectively) is not allocated any chore whose x (or y and z respectively) coordinate is 1. Then, the chore $(1, 1, 1)$ is not allocated to any agent, a contradiction.

The hard instance in Theorem 1 also implies the following lower bound for 1-out-of-d MMS.

Corollary 1 *For any $2 \leq d \leq n$, there is an instance with submodular disutility functions for which no allocation is 1-out-of-d MMS.*

Proof. We consider the same instance that was designed in Theorem 1. In this instance, we have proved that no matter how the chores are allocated among the agents, there is at least one agent, say i, whose disutility is n. Moreover, by the design of the disutility function, for any integer d, it can be observed that $\mathsf{MMS}_i^d = \lceil \frac{n}{d} \rceil$. Note that $\lceil \frac{n}{d} \rceil$ is always smaller than n for all $d \geq 2$ and thus the allocation is not 1-out-of-d MMS to i. ■

Theorem 2 *For any instance with subadditive disutility functions, there always exists a $\min\{n, \lceil \log m \rceil\}$-MMS allocation.*

Algorithm 1 Algorithm for subadditive disutility functions

Input: A subadditive instance $(N, M, \{v_i\}_{i \in N})$.

Output: An allocation $\mathbf{A} = (A_1, \dots, A_n)$ such that $v_i(A_i) \leq \lceil \log m \rceil \cdot \mathsf{MMS}_i$ for all $i \in N$.

Initialize $A_i \leftarrow \emptyset$ for every $i \in N$.
if $\log m \geq n$ **then**
  $A_1 \leftarrow M$.
else
  $i \leftarrow 1$ and $M_0 \leftarrow M$.
end if
while $M_{i-1} \neq \emptyset$ and $i \leq n$ **do**
  Let $(D^i_1, \dots, D^i_n)$ be one of i's MMS-defining partitions over M.
  Let $R^i_j = D^i_j \cap M_{i-1}$ for all $j \in [n]$. Re-index the bundles such that $|R^i_1| \geq \cdots \geq |R^i_n|$.
  $A_i \leftarrow \bigcup_{j \in [\lceil \log m \rceil]} R^i_j$ and $M_i \leftarrow M_{i-1} \setminus A_i$.
  $i \leftarrow i + 1$.
end while

Proof. We describe the algorithm that computes a $\min\{n, \lceil \log m \rceil\}$-MMS allocation in Algorithm 1. First, if $\log m \geq n$, then we are safe to arbitrarily allocate the chores to the agents, which ensures n-approximation.

The tricky case is when $\log m < n$, where we cannot allocate too many chores to a single agent. For this case, we first look at agent 1's MMS-defining partition $\mathbf{D}^1 = (D^1_1, \dots, D^1_n)$, where $c_1(D^1_j) \leq \mathsf{MMS}_1$ for all $j \in [n]$ and assume they are ordered by sizes, i.e., $|D^1_1| \geq \cdots \geq |D^1_n|$. In order to ensure agent 1's disutility to be no more than $O(\log m)$ times her MMS, we ask her to take away $\lceil \log m \rceil$ largest bundles (in terms of number of chores) in $\mathbf{D}^1$, i.e., $A_1 = \bigcup_{j \in [\lceil \log m \rceil]} D^1_j$. Since the disutility function is subadditive,

$$v_1(A_1) \leq \sum_{j \in [\lceil \log m \rceil]} v_1(D^1_j) \leq O(\log m) \cdot \mathsf{MMS}_1.$$

Moreover, since on average each bundle in $\mathbf{D}^1$ contains $\frac{m}{n}$ chores and A_1

contains the bundles with largest number of chores, $|A_1| \geq (\log m) \cdot \frac{m}{n} = \frac{\log m}{n} \cdot m$. That is, at least $\frac{\log m}{n}$ fraction of the chores are taken away by agent 1. Let $M_1 = M \setminus A_1$ be the set of remaining chores, and we have

$$|M_1| \leq \left(1 - \frac{\log m}{n}\right) \cdot m.$$

We next ask agent 2 to take away chores in a similar way to agent 1. Let $\mathbf{D}^2 = (D^2_1, \ldots, D^2_n)$ be one of agent 2's MMS-defining partitions, and $\mathbf{R}^2 = (R^2_1, \ldots, R^2_n)$ be the remaining chores in these bundles, i.e., $R^2_j = D^2_j \cap M_1$. Again, we assume $R^2_1, \ldots, R^2_n$ are ordered by sizes, i.e., $|R^2_1| \geq \cdots \geq |R^2_n|$. Letting $A_2 = \bigcup_{j \in [\lceil \log m \rceil]} R^2_j$ and $M_2 = M_1 \setminus A_2$, we have $v_2(A_2) \leq O(\log m) \cdot \mathsf{MMS}_2$. Moreover, since on average each bundle in $\mathbf{R}^2$ contains $\frac{|M_1|}{n}$ chores and A_2 contains the bundles with largest number of chores,

$$|A_2| \geq (\log m) \cdot \frac{|M_1|}{n} = \frac{\log m}{n} \cdot |M_1|,$$

$$|M_2| \leq \left(1 - \frac{\log m}{n}\right) \cdot |M_1| \leq \left(1 - \frac{\log m}{n}\right)^2 \cdot m. \tag{3.1}$$

We continue with the above procedure for agents $i = 3, \ldots, n$ with the formal description shown in Algorithm 1. It is straightforward that every agent i who gets a bundle A_i has disutility at most $O(\log m) \cdot \mathsf{MMS}_i$. Further, by induction, Equation 3.1 holds for all agents $i \leq n$, i.e.,

$$|M_i| \leq \left(1 - \frac{\log m}{n}\right) \cdot |M_{i-1}| \leq \left(1 - \frac{\log m}{n}\right)^i \cdot m.$$

To show the validity of the Algorithm, it remains to show that the algorithm can allocate all chores, i.e., $M_n = \emptyset$. This can be seen from the following inequalities,

$$
\begin{aligned}
|M_n| &\leq \left(1 - \frac{\log m}{n}\right)^n \cdot m = \left(1 - \frac{\log m}{n}\right)^{\frac{n}{\log m} \cdot \log m} \cdot m \\
&< \left(\frac{1}{e}\right)^{\log m} \cdot m < \frac{1}{m} \cdot m = 1.
\end{aligned}
$$

which means that M_n must be empty, thus completing the proof of the theorem. ■

We remark that Theorem 1 does not imply a polynomial-time algorithm and we leave this as an open problem.

Theorem 1 does not rule out the possibility of beating the approximation ratio for specific combinatorial problems with subadditive disutilities. In the next two sections, we turn to studying two settings, for which we are able to beat the lower bounds in Theorem 1 and Corollary 1 by designing algorithms that can guarantee constant multiplicative and ordinal approximations of MMS.

Note that in the following sections, we mostly consider the ordinal approximation of MMS. By Observation 1, the ordinal approximation gives a result of the multiplicative approximation, which we will improve by slightly modifying the designed algorithms.

3.2 Job Scheduling Problem

3.2.1 Model

The first specific setting encodes the job scheduling problem where the chores are jobs that need to be processed by the agents. Each chore $e_j \in M$ has a size $s_{i,j} \geq 0$ to each agent $i \in N$, and for a set of chores $S \subseteq M$, $s_i(S) = \sum_{e_j \in S} s_{i,j}$. Each agent $i \in N$ exclusively controls a set of k_i machines $P_i = [k_i]$ with possibly different speed $\rho_{i,j}$ for $j \in P_i$. Without loss of generality, we assume $\rho_{i,1} \geq \cdots \geq \rho_{i,k_i}$. Let $d = \lfloor \frac{n}{2} \rfloor$. Upon receiving a set of chores $S \subseteq M$, agent i's disutility $v_i(S)$ is the minimum completion time of processing S using her own machines P_i (i.e., *the makespan of* P_i). Formally,

$$v_i(S) = \min_{(T_1,\ldots,T_{k_i}) \in \Pi_{k_i}(S)} \max_{l \in [k_i]} \frac{\sum_{e_t \in T_l} s_{i,t}}{\rho_{i,l}}.$$

Note that the computation of $v_i(S)$ is NP-hard if $k_i \geq 2$. Moreover, for any two sets S_1 and S_2, $v_i(S_1 \cup S_2) \leq v_i(S_1) + v_i(S_2)$ since the makespan of scheduling $S_1 \cup S_2$ is no larger than the sum of the makespans of scheduling S_1 and S_2 separately and thus $v_i(\cdot)$ is subadditive.

Regarding the value of MMS_i^d, intuitively, it is obtained by partitioning the chores into $d \cdot k_i$ bundles, and allocating them to k_i different types of machines (with possibly different speeds) where each type has d identical machines so that the makespan is minimized.[1] Note that when each agent

[1] An alternative way to explain the scheduling model is to view each agent i as a group of k_i small agents and MMS_i^d as the *collective maximin share* for these k_i small agents. We believe this notion of collective maximin share is of independent interest as a groupwise fairness notion. We remark that this notion is different with the groupwise (and pairwise) maximin share defined in [28] and [57], where the max-min value is defined for each single agent. In our definition, however, a set of agents share the same value for the chores

controls a single machine, i.e., $k_i = 1$ for all i, the problem degenerates to the additive disutility case, and thus the job scheduling setting strictly generalizes the additive setting. For each agent i and each machine $j \in P_i$, let $c_{i,j} = \rho_{i,j} \cdot \mathsf{MMS}_i^d$ denote j's capacity, which means that if the total size of a set of chores to i does not exceed $c_{i,j}$, the time it takes for j to process the chores does not exceed MMS_i^d.

3.2.2 The IDO Reduction

For an instance $I = (N, M, \{v_i\}_{i \in N}, \{s_i\}_{i \in N})$ of the job scheduling setting or the bin packing setting, we can construct the IDO instance $I' = (N, M, \{v'_i\}_{i \in N}, \{s'_i\}_{i \in N})$ where $s_{i,1} \geq \cdots \geq s_{i,m}$ for all i. Observe that for any $i \in N$, there exists a permutation $\sigma_i : M \to M$ such that for any $e_j, e'_j \in M$, $e_j \leq e'_j$ implies $s_{i,\sigma_{i,j}} \geq s_{i,\sigma_{i,j'}}$. These permutations are used to define the sizes of the chores in I' as follows: for each agent $i \in N$, $s'_{i,j} = s_{i,\sigma_{i,j}}$ for every $e_j \in M$. In short, for each agent $i \in N$ and each chore $e_j \in M$, the size of e_j to i in I' is the j-th largest size of the chores to i in I. Note that the above construction runs in polynomial time.

We then apply a widely-used reduction [46, 104] to restrict our attention on IDO instances. Specifically, it means that any algorithm that ensures α-approximate 1-out-of-d MMS allocations for IDO instances can be converted to compute α-approximate 1-out-of-d MMS allocations for general instances. The reduction may not work for all subadditive valuations, but we show in the following lemma that it does work for the job scheduling setting and the bin packing setting.

assigned to them.

Algorithm 2 IDO reduction for the job scheduling setting and the bin packing setting

Input: A general instance I, the IDO instance I' and an allocation $\mathbf{A}' = (A'_1, ..., A'_n)$ for the IDO instance such that $v'_i(A'_i) \leq \alpha \cdot \mathsf{MMS}^d_i(I')$ for all $i \in N$.

Output: An allocation $\mathbf{A} = (A_1, ..., A_n)$ such that $v_i(A_i) \leq \alpha \cdot \mathsf{MMS}^d_i(I)$ for all $i \in N$.

1: For all $i \in N$ and $g \in A'_i$ set $p_g := i$.
2: Initialize $A_i \leftarrow \emptyset$ for all $i \in N$, and $R \leftarrow M$.
3: **for** $g = m$ to 1 **do**
4:     Pick $k_g \in \arg\min_{g' \in R}\{s_{p_g,g'}\}$.
5:     $A_{p_g} \leftarrow A_{p_g} \cup \{k_g\}$, $R \leftarrow R \setminus \{k_g\}$.
6: **end for**

Lemma 3 *For the job scheduling setting or the bin packing setting, if there exists an allocation* $\mathbf{A}' = (A'_1, \ldots, A'_n)$ *in the IDO instance* I' *such that* $v'_i(A'_i) \leq \alpha \cdot \mathsf{MMS}^d_i(I')$ *for all* $i \in N$*, then there exists an allocation* $\mathbf{A} = (A_1, \ldots, A_n)$ *in the original instance* I *such that* $v_i(A_i) \leq \alpha \cdot \mathsf{MMS}^d_i(I)$ *for all* $i \in N$*. Furthermore,* $\mathbf{A}$ *can be constructed in polynomial time.*

Proof. We show that given any I, I' and $\mathbf{A}'$, Algorithm 2 computes the desired allocation $\mathbf{A}$ in polynomial time. In the algorithm, we look over the chores in descending order of their indices (i.e., in increasing order of their sizes in I'). For each chore, we let the agent who receives it in I' pick her smallest remaining chore in I.

Clearly, Algorithm 2 runs in polynomial time. We next show that $v_i(A_i) \leq v'_i(A'_i)$ for all $i \in N$. Consider the g-th iteration of the for-loop (Steps 3 to 6), where we suppose that agent i picks chore k_g; that is, $g \in A'_i$, $k_g \in A_i$ and k_g is the smallest remaining chore to i. Since a chore is removed from the set R after it is allocated, exactly $g-1$ chores have been allocated before k_g is allocated. Therefore, k_g is among the top g smallest chores to agent

i. Recall that g is the chore with the exactly g-th smallest size to i, hence $s_{i,k_g} \leq s'_{i,g}$. The same reasoning can be applied to other chores in A'_i and A_i, and to other agents. It follows that for any $i \in N$, any $g \in A'_i$ and the corresponding $k_g \in A_i$, $s_{i,k_g} \leq s'_{i,g}$. For the job scheduling setting or the bin packing setting, this implies $v_i(A_i) \leq v'_i(A'_i)$. Since the maximin share depends on the sizes of the chores but not on the order, the maximin share of agent i in I' is the same as that in I, i.e., $\mathsf{MMS}^d_i(I') = \mathsf{MMS}^d_i(I)$. Hence, the condition that $v'_i(A'_i) \leq \alpha \cdot \mathsf{MMS}^d_i(I')$ gives $v_i(A_i) \leq \alpha \cdot \mathsf{MMS}^d_i(I)$, which completes the proof. ■

Therefore, in the job scheduling setting and the bin packing setting, we only consider IDO instances.

3.2.3 Algorithm

Next, we elaborate on the algorithm that proves Theorem 3.

Theorem 3 *A 1-out-of-$\lfloor \frac{n}{2} \rfloor$ MMS allocation always exists for any job scheduling instance.*

In a nutshell, our algorithm consists of three parts: we first partition all chores into d bundles. For any of the bundles and any agent, in the second part, we present an imaginary assignment of the chores in the bundle to the agent's machines. These imaginary assignments are used in the third part to guide the allocation of the chores in each of the d bundles to two agents, such that each agent receives disutility no more than her 1-out-of-d MMS.

Part 1: partitioning the chores into d bundles

We adopt the classic round-robin algorithm to partition the chores into d bundles $\mathbf{B} = (B_1, \dots, B_d)$. Specifically, we allocate the chores in descending order of their sizes to the bundles by turns, from the first bundle to the last one. Each time, we allocate one chore to one bundle, and when every bundle receives a chore, we start over from the first bundle and so on. For any set of chores S, let $S[l]$ be the l-th largest chore, then the algorithm is formally presented in Algorithm 3.

Algorithm 3 Round-robin algorithm

Input: An IDO job scheduling instance $(N, M, \{v_i\}_{i\in N}, \{s_i\}_{i\in N})$.
Output: A d-partition of M: $\mathbf{B} = (B_1, \dots, B_d)$.

Initialize $B_j \leftarrow \emptyset$ for every $j \in [d]$, and $r \leftarrow 1$.
while $r \leq m$ **do**
  for $j = 1$ to d **do**
    if $r \leq m$ **then**
      $B_j \leftarrow B_j \cup \{M[r]\}$.
      $r \leftarrow r + 1$.
    end if
  end for
end while

By the characteristic of the round-robin algorithm, we have the following observation.

Observation 2 *For each bundle $B_j \in \mathbf{B}$ and each chore $e_k \in B_j \setminus \{B_j[1]\}$ (if exists), there are at least $d-1$ other chores in M with at least the same size as e_k; that is, chores $e_{k-1}, e_{k-2}, \dots, e_{k-d+1}$.*

Part 2: imaginary assignment

Consider any bundle $B_j \in \mathbf{B}$ computed in the first part and any agent $i \in N$, we imaginatively assign the chores in $B_j \setminus B_j[1]$ to i's machines as follows. We greedily assign the chores with larger sizes to i's machines with faster speeds (in other words, larger capacities), as long as the total workload on one machine does not exceed the machine's capacity. The first time when the workload exceeds the capacity, we move to the next machine and so on.

Algorithm 4 Imaginary assignment

Input: A bundle $B_j \in \mathbf{B}$ computed in the first part and an agent $i \in N$.
Output: Sets of internal chores $\{C^I_{i,1}, \ldots, C^I_{i,k_i}\}$ and external chores $\{t_{i,0}, \ldots, t_{i,k_i}\}$.

Initialize $C^I_{i,l} \leftarrow \emptyset$, $t_{i,l} \leftarrow$ null for every $l \in [k_i]$, and $r \leftarrow 1$.
while $r \leq |B_j|$ **do**
  for $l = 1$ to k_i **do**
    $t_{i,l-1} \leftarrow B_j[r]$, $r \leftarrow r + 1$.
    while $r \leq |B_j|$ and $s_i(C^I_{i,l} \cup \{B_j[r]\}) \leq c_{i,l}$ **do**
      $C^I_{i,l} \leftarrow C^I_{i,l} \cup \{B_j[r]\}$, $r \leftarrow r + 1$.
    end while
  end for
end while

The algorithm is formally presented in Algorithm 4 and illustrated in Figure 3.2, where for each $l \in [k_i]$, $C^I_{i,l}$ contains the chores assigned to machine l that do not make the total workload exceed l's capacity, and $t_{i,l}$ is the last chore assigned to l that makes the total workload exceed the capacity. Note that $C^I_{i,l}$ may be empty and $t_{i,l}$ may be null. For simplicity, let $t_{i,0} = B_j[1]$; that is, $B_j[1]$ is assigned to an imaginary machine 0. The chores in $\bigcup_{l \in [k_i]} C^I_{i,l}$ are called *internal* chores (as shown by the *dark* boxes in Figure 3.2), and $\{t_{i,0}, \ldots, t_{i,k_i}\}$ are called *external* chores (as shown by the *light* boxes).

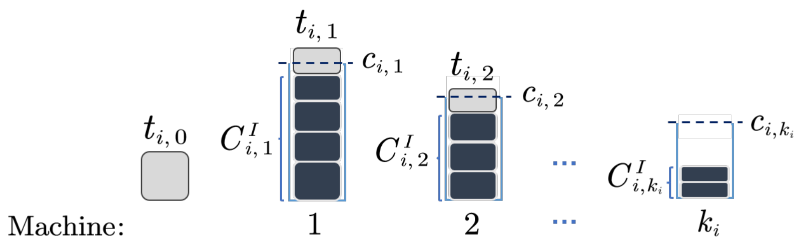


Figure 3.2: The imaginary assignment of B_j to agent i

For any bundle $B_j \in \mathbf{B}$ and any agent $i \in N$, the imaginary assignment has the following properties.

- **Property 1**: all chores in $B_j \setminus \{B_j[1]\}$ can be assigned to agent i's machines. Besides, the last machine k_i does not have an external chore; that is, t_{i,k_i} is null.

- **Property 2**: for any $1 \leq l \leq k_i$, the total size of the internal chores $C^I_{i,l}$ does not exceed the capacity of machine l:

$$s_i(C^I_{i,l}) \leq c_{i,l}.$$

- **Property 3**: for any $1 \leq l \leq k_i$, the external chore $t_{i,l-1}$ (if not null) has size no greater than the capacity of machine l:

$$s_i(t_{i,l-1}) \leq c_{i,l}.$$

Proof. The first property holds since otherwise, $s_i(B_j \setminus \{B_j[1]\}) > \sum_{l \in [k_i]} c_{i,l}$.

By Observation 2, it follows that

$$s_i(M) > d \cdot s_i(B_j \setminus \{B_j[1]\}) > d \cdot \sum_{l \in [k_i]} c_{i,l}.$$

However, since all chores can be assigned to i's machines in her MMS-defining partition, we have $s_i(M) \leq d \cdot \sum_{l \in [k_i]} c_{i,l}$, a contradiction.

The second property directly follows the algorithm. For the third property, $s_i(t_{i,0}) \leq c_{i,1}$ follows the facts that $t_{i,0}$ is assigned to some machine in i's MMS-defining partition and $c_{i,1}$ is the largest capacity of the machines. We then consider $l \in [k_i - 1]$ and show $s_i(t_{i,l}) \leq c_{i,l+1}$ (if $t_{i,l}$ is not null). The same reasoning can be applied to any other $l' \in [k_i - 1]$. Let $S_1 = \bigcup_{p \in [l]} (C^I_{i,p} \cup \{t_{i,p}\})$. From the algorithm, we know that $s_i(S_1) > \sum_{p \in [l]} c_{i,p}$ and $t_{i,l}$ is the smallest chore in S_1. By Observation 2, there exist another $d-1$ subsets of chores $\{S_2, \ldots, S_d\}$ such that $s_i(S_k) \geq s_i(S_1)$ for any $k \in [2, d]$ and $t_{i,l}$ is also the smallest chore in $\bigcup_{k \in [d]} S_k$. Hence, $\sum_{k \in [d]} s_i(S_k) > d \cdot \sum_{p \in [l]} c_{i,p}$. This implies that in i's MMS-defining partition, at least one chore in $\bigcup_{k \in [d]} S_k$ is assigned to machine $p \geq l + 1$. Combining with the fact that $t_{i,l}$ is the smallest chore in $\bigcup_{k \in [d]} S_k$, we have $s_i(t_{i,l}) \leq c_{i,l+1}$. ■

By these properties, for any agent $i \in N$ and any of her machine $l \in P_i$, we can assign either the internal chores $C^I_{i,l}$ or the external chore $t_{i,l-1}$ to l, such that the completion time of l does not exceed MMS^d_i. This intuition guides the allocation of the chores in each bundle in $\mathbf{B}$ to two agents in the following part.

Part 3: allocating the chores to the agents

For any bundle $B_j \in B$, we arbitrarily choose two agents $i_1, i_2 \in N$ and allocate them the chores in B_j in the following way. Recall that in the imaginary assignment of B_j to each agent $i \in \{i_1, i_2\}$, the chores in B_j are divided into internal chores $\bigcup_{l\in[k_i]} C^I_{i,l}$ and external chores $\{t_{i,0}, \ldots, t_{i,k_i}\}$. Let $E = \{e^*_1, \ldots, e^*_{|E|}\}$ contain all external chores shared by i_1 and i_2. We allocate the chores in B_j to agents i_1 and i_2 in $|E|$ rounds. In each round $q \in [|E|]$, we first find the machines of i_1 and i_2 to which the shared external chores e^*_q and e^*_{q+1} are assigned (denoted by l_1, l_2, l'_1 and l'_2, respectively. If e^*_{q+1} does not exist, simply let $l'_1 = k_{i_1}$ and $l'_2 = k_{i_2}$). We then find the agent $i_k \in \{i_1, i_2\}$ whose machine $l_k + 1$ has more internal chores. We allocate i_k her internal chores from machine $l_k + 1$ to machine l'_k, and allocate the other agent i_k's external chores from machine l_k to machine $l'_k - 1$. The algorithm is formally presented in Algorithm 5.

Since $2 \cdot d = 2 \cdot \lfloor \frac{n}{2} \rfloor \leq n$, no more than n agents are needed to allocate all chores. Thus to prove Theorem 3, it remains to show that for each agent who is allocated chores in Algorithm 5, the disutility of the chores allocated to her is no more than her 1-out-of-d MMS.

Proof of Theorem 3. Consider the first round of the process when the chores in any bundle $B_j \in \mathbf{B}$ are allocated to two arbitrarily chosen agents $i_1, i_2 \in N$. Without loss of generality, assume that the first machine of i_1 contains more internal chores than that of i_2, i.e., $C^I_{i_1,1} \geq C^I_{i_2,1}$. From the algorithm, we know that the chores i_1 takes are $\bigcup_{l=1}^{l'_1} C^I_{i_1,l}$. By the second property of the imaginary assignment, these chores can be assigned to the

Algorithm 5 Allocating the chores to the agents

Input: A d-partition $\mathbf{B} = (B_1, \ldots, B_d)$ returned by Algorithm 3.
Output: An allocation $\mathbf{A} = (A_1, \ldots, A_n)$ such that $v_i(A_i) \leq \mathsf{MMS}_i^d$ for all $i \in N$.

Initialize $A_i \leftarrow \emptyset$ for every $i \in N$.
for $j = 1$ to d **do**
  Arbitrarily choose 2 agents i_1 and i_2, $N \leftarrow N \setminus \{i_1, i_2\}$.
  $\{C^I_{i_1,1}, \ldots, C^I_{i_1,k_{i_1}}\}, \{t_{i_1,0}, \ldots, t_{i_1,k_{i_1}}\} \leftarrow$ *Algorithm* 4(B_j, i_1).
  $\{C^I_{i_2,1}, \ldots, C^I_{i_2,k_{i_2}}\}, \{t_{i_2,0}, \ldots, t_{i_2,k_{i_2}}\} \leftarrow$ *Algorithm* 4(B_j, i_2).
  $E \leftarrow \{t_{i_1,0}, \ldots, t_{i_1,k_{i_1}}\} \cap \{t_{i_2,0}, \ldots, t_{i_2,k_{i_2}}\}$. Re-label $E \leftarrow \{e^*_1, \ldots, e^*_{|E|}\}$.
  for $q = 1$ to $|E|$ **do**
    Find $l_1 \in [0, k_{i_1}]$ and $l_2 \in [0, k_{i_2}]$ such that $e^*_q = t_{i_1,l_1} = t_{i_2,l_2}$.
    if $q < |E|$ **then**
      Find l'_1 and l'_2 such that $e^*_{q+1} = t_{i_1,l'_1} = t_{i_2,l'_2}$.
    else
      $l'_1 = k_{i_1}$ and $l'_2 = k_{i_2}$.
    end if
    if $|C^I_{i_1,l_1+1}| \geq |C^I_{i_2,l_2+1}|$ **then**
      $A_{i_1} \leftarrow \bigcup_{l=l_1+1}^{l'_1} C^I_{i_1,l}$, $A_{i_2} \leftarrow \bigcup_{l=l_1}^{l'_1-1} t_{i_1,l}$.
    else
      $A_{i_2} \leftarrow \bigcup_{l=l_2+1}^{l'_2} C^I_{i_2,l}$, $A_{i_1} \leftarrow \bigcup_{l=l_2}^{l'_2-1} t_{i_2,l}$.
    end if
  end for
end for

first l'_1 machines of i_1 such that the completion time of each machine does not exceed $\mathsf{MMS}_{i_1}^d$. Besides, we know that the chores i_2 takes are $\bigcup_{l=0}^{l'_1-1} t_{i_1,l}$, which are e_0^* and a subset of $\bigcup_{l=2}^{l'_2} C_{i_2,l}^I$. By the second and third properties of the imaginary assignment, these chores can be assigned to the first l'_2 machines of i_2 such that each completion time does not exceed $\mathsf{MMS}_{i_2}^d$. The same reasoning can be applied to all the following rounds. By induction, it follows that both i_1 and i_2 can process the chores allocated to them such that the maximum completion time of their machines does not exceed their 1-out-of-d MMS, which completes the proof. ■

For the multiplicative relaxation of MMS, by Theorem 3 and Observation 1, a $\lceil n/\lfloor \frac{n}{2} \rfloor \rceil$-MMS allocation always exists for any job scheduling instance. We next show that after a slight modification, Algorithm 3 computes a 2-MMS allocation in polynomial time, which is better than $\lceil n/\lfloor \frac{n}{2} \rfloor \rceil$-MMS.

Corollary 2 *A 2-MMS allocation can be computed in polynomial time for any job scheduling instance.*

Proof. We show that by replacing the value of d with n, Algorithm 3 computes a 2-MMS allocation. Particularly, in the new version of Algorithm 3, we partition the chores in M into n bundles in a round-robin fashion and allocate each of the n bundles to one agent in N. By the properties of the imaginary assignment, for each agent, the makespan of processing either the internal chores or the external chores in her bundle using her machines does not exceed MMS_i^n. This implies that for each agent the total disutility of her bundle does not exceed $2\mathsf{MMS}_i^d$, which completing the proof. ■

3.3 Bin Packing Problem

3.3.1 Model

The second setting encodes the bin packing problem where the chores have sizes and need to be packed into bins by the agents. The chores may be of different sizes to different agents. Specifically, each chore $e_j \in M$ has size $s_{i,j} \geq 0$ to each agent $i \in N$. As in the job scheduling setting, it suffices to consider IDO bin packing instances where $s_{i,1} \geq \cdots \geq s_{i,m}$ for all $i \in N$. For a set of chores S, $s_i(S) = \sum_{e_j \in S} s_{i,j}$. Each agent $i \in N$ has unlimited number of bins with the same capacity c_i. Without loss of generality, we assume $c_1 \geq \cdots \geq c_n$ and $c_i \geq \max_{e_j \in M} s_{i,j}$ for all $i \in N$.

Upon receiving a set of chores $S \subseteq M$, agent i's disutility $v_i(S)$ [2] is determined by the minimum number of bins (with capacity c_i) that can pack all chores in S. The calculation of $v_i(S)$ involves solving a classic bin packing problem which is NP-hard. For any two sets S_1 and S_2, $v_i(S_1 \cup S_2) \leq v_i(S_1) + v_i(S_2)$ since the optimal packing of $S_1 \cup S_2$ is no worse than the union of packing S_1 and S_2 separately and thus $v_i(\cdot)$ is subadditive. Let $d = \lfloor \frac{n}{2} \rfloor$. Accordingly, MMS_i^d is essentially the minimum number k_i such that the chores can be partitioned into d bundles and the chores in each bundle can be packed into no more than k_i bins. The definition of MMS_i^d gives a simple observation: $\mathsf{MMS}_i^d \cdot c_i \geq \frac{s_i(M)}{d}$ for all $i \in N$.

We say a chore e_j is *large* for an agent i if the size of e_j to i exceeds half of the capacity of i's bins, i.e., $s_{i,j} > c_i/2$; otherwise, we say e_j is *small* for

[2] Note that although the value of $v_i(S)$ also depends on c_i, to simplify the notations, we let the subscript i absorb c_i and neglect an extra parameter in $v_i(\cdot)$.

i. Let H_i denote the set of i's large chores in M, and L_i denote the set of i's small chores:

$$\begin{aligned} H_i &= \{e_j \in M \mid s_{i,j} > c_i/2\}, \\ L_i &= \{e_j \in M \mid s_{i,j} \le c_i/2\}. \end{aligned} \tag{3.2}$$

Since two large chores cannot be put together into the same bin, the number of chores that are large for i is at most $\mathsf{MMS}_i^d \cdot d$; that is, $|H_i| \le \mathsf{MMS}_i^d \cdot d$. We say a set of chores are *acceptable* for an agent i if their total size does not exceed c_i. If the total size exceeds c_i, we still say these chores are *passable* for i, as long as some of them are small for i and removing one of the small chores makes the remaining chores acceptable for i.

3.3.2 Algorithm

Next, we introduce the algorithm that proves Theorem 4.

Theorem 4 *A 1-out-of-$\lfloor \frac{n}{2} \rfloor$ allocation can be computed in polynomial time for any bin packing instance.*

Actually, we can apply the similar idea (round-robin) in Subsection 3.2.3 to the bin packing setting, however, there will be an extra additive loss. To avoid this loss, in the following, we introduce a new approach based on the bag filling algorithm. In a nutshell, our algorithm contains two parts: in the first part (the main part), we adopt the bag filling algorithm to partition the chores into d bundles, and at the same time select at most two agents for each bundle. In the second part, for each of the d bundles, we present an imaginary assignment of the chores in the bundle to the bins of each of its selected agents. We then use these imaginary assignments to allocate the chores in the bundles to their selected agents, such that each agent receives disutility at most her 1-out-of-d MMS.

Part 1: modified bag filling algorithm

The algorithm (see Algorithm 6) runs in d rounds of bag initialization (Steps 5 through 8) and bag filling (Steps 12 through 18). Specifically, we call the agents for whom the bag is not large enough and some remaining chores are small *candidate agents*. Note that the set of candidate agents shrinks when chores are added into the bag during the algorithm. In the bag initialization procedure of round $j \in [d]$, we put e_j and the chores every d chores after e_j (i.e., $e_{j+d}, e_{j+2d}, \ldots$) into the bag, as long as the chores have not been allocated and are large for at least one agent. Observe that since we consider IDO instances, the agents for whom the lastly added chore is large think all chores in the initialized bag are large. We select one such agent. If there is at most one candidate agent after the bag initialization procedure, the round ends and the candidate agent (if exists and has not been selected) is added as another selected agent. Otherwise, we enter the bag filling procedure.

In the bag filling procedure, as long as there exist at least two candidate agents, we replace the selected agents with two of the candidate agents and put the smallest remaining chore into the bag. Note that the smallest chore is small for all candidate agents (including the two selected). If there is at most one candidate agent after the smallest chore is put into the bag, the round ends and the only candidate agent (if exists and has not been selected) replaces one of the selected agents. The way we establish the bag and select the agents makes the following two properties satisfied for every round $j \in [d]$.

Algorithm 6 Modified bag filling algorithm.

Input: An IDO bin packing instance $(N, M, \{v_i\}_{i\in N}, \{s_i\}_{i\in N})$.
Output: A d-partition of M: $\mathbf{B} = \{B_1, \ldots, B_d\}$, and sets of selected agents $\mathbf{G} = \{G_1, \ldots, G_d\}$.

Initialize $B_j \leftarrow \emptyset$, $G_j \leftarrow \emptyset$ for each $j \in [d]$, and $R \leftarrow M$.
$N(B_j) = \{i \in N : s_i(B_j) \leq \frac{s_i(M)}{d}$ and $L_i \cap R \neq \emptyset\}$. // candidate agents which change with B_j and R
for $j = 1$ to d **do**
    $t \leftarrow j$.
    while $e_t \in R$ and there exists an agent $i \in N$ for whom e_t is large **do**
        $B_j \leftarrow B_j \cup \{e_t\}$, $R \leftarrow R \setminus \{e_t\}$, $t \leftarrow t + d$.
        $G_j \leftarrow \{i\}$.
    end while
    if $|N(B_j)| = 1$ and $N(B_j) \nsubseteq G_j$ **then**
        Pick $i \in N(B_j)$, $G_j \leftarrow G_j \cup \{i\}$.
    end if
    while $|N(B_j)| \geq 2$ **do**
        $i_1, i_2 \in N(B_j)$, $G_j \leftarrow \{i_1, i_2\}$.
        Pick the smallest chore $e \in R$, $B_j \leftarrow B_j \cup \{e\}$, $R \leftarrow R \setminus \{e\}$.
        if $|N(B_j)| = 1$ and $N(B_j) \nsubseteq G_j$ **then**
            Pick $i \in N(B_j)$ and replace one arbitrary agent in G_j with agent i.
        end if
    end while
    $N \leftarrow N \setminus G_j$.
end for

- **Property 1**: for each selected agent $i \in G_j$, there are at most MMS_i^d chores in B_j that are large for i. Besides, letting e_j^* be the chore lastly added to B_j, if e_j^* is small for i, we have $s_i(B_j \setminus \{e_j^*\}) \leq \frac{s_i(M)}{d}$.

- **Property 2**: for each remaining agent i' (i.e., $i' \notin \bigcup_{l\in[j]} G_l$), either $s_{i'}(B_j) > \frac{s_{i'}(M)}{d}$ or no remaining chore is small for i'. Besides, all chores in $\{e_j, e_{j+d}, \ldots\}$ that are large for i' have been allocated.

Proof. For the first property, observe that large chores are added into the bag only in the bag initialization procedure, where one out of every d chores is picked. Since there are at most $d \cdot \mathsf{MMS}_i^d$ large chores for every agent i, the bag contains at most MMS_i^d chores that are large for i. There are two cases where e_j^* is small for an agent $i \in G_j$. First, i is the only candidate agent after e_j^* is added, for which case, we have $s_i(B_j) \leq \frac{s_i(M)}{d}$. Second, i is one of the two selected candidate agents before e_j^* is added, for which case, we have $s_i(B_j \setminus \{e_j^*\}) \leq \frac{s_i(M)}{d}$. In both cases, $s_i(B_j \setminus \{e_j^*\}) \leq \frac{s_i(M)}{d}$ holds.

The second property is quite direct by the algorithm, since there remains no candidate agent outside G_j at the end of round j, and all remaining large chores in $\{e_j, e_{j+d}, \ldots\}$ are allocated in the bag initialization procedure of round j. ■

Property 1 ensures that the chores in every bundle $B_j \in \mathbf{B}$ can be allocated to the agents in G_j, such that each agent $i \in G_j$ can use at most MMS_i^d bins to pack all chores allocated to her, which will be shown in the following part. Property 2 ensures the following claim.

Claim 1 *all chores can be allocated in Algorithm 6.*

Proof. Observe that when the last round begins, there are at least $n - (d-1) \cdot 2 \geq 2$ remaining agents. If all remaining chores are large for some remaining agent, all of them are added into the bag during the bag initialization procedure of the last round and no chore remains unallocated. Now consider the case where some remaining chore is small for any remaining agent. By Property 2, for any $j \in [d-1]$ and any remaining agent i', we have $s_{i'}(B_j) > \frac{s_{i'}(M)}{d}$. This gives that the total size of all remaining chores to any remaining agent i' is smaller than $\frac{s_{i'}(M)}{d}$. Besides, after the bag initialization procedure of the last round, no large chore remains and every remaining chore is small for any remaining agent. Combining these two facts, we know that there are always at least 2 candidate agents and thus all small chores can also be allocated. ■

Part 2: Allocating the chores to the agents

Next, we allocate the chores in each bundle $B_j \in \mathbf{B}$ to the selected agents in G_j. Let i be any agent in G_j and $B'_j = B_j \setminus \{e^*_j\}$ where e^*_j is the chore lastly added to B_j. We first imaginatively assign the chores in B'_j to i's bins. As illustrated by Figure 3.3, we first put i's large chores in B'_j into individual empty bins. We then greedily put the remaining small chores in B'_j into the bins in decreasing order of their sizes, as long as the total size of the assigned chores does not exceed the bin's capacity. The first time when the total size exceeds the capacity, we move to the next bin and so on (if all bins with large chores are filled, we move to an empty bin). We call the chore lastly added to each bin that makes the total size exceed the capacity an *extra chore*. Denote by $J_i(B'_j)$ the set of extra chores and by $W_i(B'_j) = B'_j \setminus J_i(B'_j)$ the other chores in B'_j.

If all chores in B_j are large for some agent $i \in G_j$, we allocate all of them to i. Otherwise, by the algorithm, we know that there are two agents in G_j and the last chore e^*_j is small for both of them. Letting i_1 be the agent who has more large chores in B_j and i_2 be the other agent, we allocate i_1 the chores in $W_{i_1}(B'_j)$ and allocate i_2 the chores in $J_{i_1}(B'_j) \cup \{e^*_j\}$.

Clearly, the algorithms in the first part and the second part run in poly-

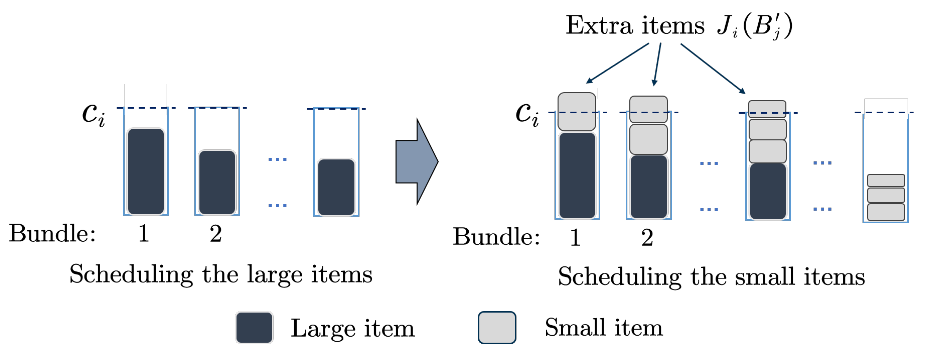


Figure 3.3: Imaginary assignment of B'_j to agent i

nomial time. Thus to prove Theorem 4, it remains to show that any agent $i \in N$ can pack all chores allocated to her with no more than MMS_i^d bins.

Proof of Theorem 4. Consider any round $j \in [d]$. If all chores in B_j are large for some agent $i \in G_j$, by Property 1, we know that there are at most MMS_i^d chores in B_j. Thus i can pack all chores in B_j using no more than MMS_i^d bins.

For the other case, recall that the agent $i_1 \in G_j$ who has more large chores in B_j receives the chores in $W_{i_1}(B'_j)$, and the other agent i_2 receives the chores in $J_{i_1}(B'_j) \cup \{e^*_j\}$. We first discuss agent i_1. By Property 1, we know that for each agent $i \in \{i_1, i_2\}$, there are at most MMS_i^d large chores in B'_j and $s_i(B'_j) \leq \frac{s_i(M)}{d}$. These two facts imply that in the imaginative assignment of B'_j to i_1, no more than $\mathsf{MMS}_{i_1}^d$ bins are used. Since otherwise, $s_{i_1}(B'_j) > \mathsf{MMS}_{i_1}^d \cdot c_{i_1} \geq \frac{s_{i_1}(M)}{d}$, a contradiction. Therefore, i_1 can pack all chores in $W_{i_1}(B'_j)$ using no more than $\mathsf{MMS}_{i_1}^d$ bins.

Next we discuss agent i_2. Observe that in the imaginary assignment of B'_j to i_1, for each extra chore in $J_{i_1}(B'_j)$, there exists another chore in the same bin with a larger size. Therefore, we have $s_{i_2}(J_{i_1}(B'_j)) \leq \frac{s_{i_2}(B'_j)}{2} \leq \frac{s_{i_2}(M)}{2d}$.

Combining with the fact that there are at most $\mathsf{MMS}^d_{i_2}$ large chores in B'_j for i_2, we know that i_2 can use no more than $\mathsf{MMS}^d_{i_2}$ bins to pack all chores in $J_{i_1}(B'_j)$ and there exists one bin with at least half the capacity not occupied. Since otherwise, $s_{i_2}(J_{i_1}(B'_j)) > \mathsf{MMS}^d_{i_2} \cdot \frac{c_{i_2}}{2} \geq \frac{s_{i_2}(M)}{2d}$, a contradiction. Recall that the last chore e^*_j is small for i_2, it can be put into the bin that has enough unoccupied capacity. Therefore, i_2 can also pack all chores in $J_{i_1}(B'_j) \cup \{e^*_j\}$ using at most $\mathsf{MMS}^d_{i_2}$ bins, which completes the proof. ■

For the multiplicative relaxation of MMS, by Theorem 4 and Observation 1, we can compute a $\lceil n/\lfloor \frac{n}{2} \rfloor \rceil$-MMS allocation in polynomial time. As the job scheduling setting, we show that slightly modifying the algorithm gives a 2-MMS allocation.

Corollary 3 *A 2-MMS allocation can be computed in polynomial time for any bin packing instance.*

Proof. To compute a 2-MMS allocation, we replace the value of d with n in Algorithm 6 and select only one agent in each round who receives the bag in that round. The modified algorithm is formally presented in Algorithm 7.

Following the same reasonings in Parts 1 and 2, it is not hard to see that all chores can be allocated in Algorithm 7 and for any $i \in N$, there are at most MMS_i large chores in A_i. Besides, if the last chore e^*_i is small for i, we have $s_i(A_i \setminus \{e^*_i\}) \leq \frac{s_i(M)}{n}$. Again, in the imaginary assignment of $A_i \setminus \{e^*_i\}$ to i, no more than MMS_i bins are used and at least one of them does not have an extra chore. Therefore, agent i can use MMS_i bins to pack all chores in $W_i(A_i \setminus \{e^*_i\})$ and another MMS_i bins to pack all chores in $J_i(A_i \setminus \{e^*_i\}) \cup \{e^*_i\}$, which completes the proof. ■

Algorithm 7 Computing 2-MMS allocations for bin packing setting

Input: An IDO bin packing instance $(N, M, \{v_i\}_{i\in N}, \{s_i\}_{i\in N})$.
Output: An allocation $\mathbf{A} = (A_1, \ldots, A_n)$ such that $v_i(A_i) \leq 2 \cdot \mathsf{MMS}_i$ for $i \in N$.

```
Initialize R ← M.
for j = 1 to n do
    Initialize B ← ∅, t ← j, k ← an arbitrary agent in N.
    while e_t ∈ R and there exists an agent i_1 for whom e_t is large do
        B ← B ∪ {e_t}, R ← R \ {e_t}, t ← t + n.
        k ← i_1.
    end while
    while there exists an agent i_2 that satisfies s_{i_2}(B) ≤ s_{i_2}(M)/n and L_{i_2} ∩ R ≠ ∅ do
        k ← i_2.
        e ← the smallest chore in R, B ← B ∪ {e}, R ← R \ {e}.
    end while
    A_k ← B, N ← N \ {k}.
end for
```

We next show that the above multiplicative ratio is actually tight in the sense that there exists an instance where no allocation is better than 2-MMS. We first recall the impossibility instance given by Feige et al. [79]. In this instance there are three agents and nine chores as arranged in a three by three matrix. The three agents' disutilities are shown in the matrices V_1, V_2 and V_3.

$$V_1 = \begin{pmatrix} 6 & 15 & 22 \\ 26 & 10 & 7 \\ 12 & 19 & 12 \end{pmatrix} \qquad V_2 = \begin{pmatrix} 6 & 15 & 23 \\ 26 & 10 & 8 \\ 11 & 18 & 12 \end{pmatrix}$$

$$V_3 = \begin{pmatrix} 6 & 16 & 22 \\ 27 & 10 & 7 \\ 11 & 18 & 12 \end{pmatrix}$$

Feige et al. [79] proved that for this instance the MMS value of every agent is 43, however, in any allocation, at least one of the three agents gets disutility no smaller than 44.

We can adapt this instance to the bin packing setting and obtain a lower bound of 2. In particular, we also have three agents and nine chores. The numbers in matrices V_1, V_2 and V_3 are the sizes of the chores to agents $1, 2$ and 3, respectively. Let the capacities of the bins of each agent be $c_i = 43$ for all $i \in \{1, 2, 3\}$. Accordingly, we have $\mathsf{MMS}_i = 1$. Since in any allocation, there is at least one agent who gets chores with total size no smaller than 44, for this agent, she has to use 2 bins to pack the assigned chores, which means that no allocation can be better than 2-MMS.

At last, we show that Algorithm 7 actually computes an allocation where every agent i can use at most $\frac{3}{2}\mathsf{MMS}_i + 1$ bins to pack all chores allocated to her. Recall that in the proof of Corollary 3, it has been shown that each agent $i \in N$ can use MMS_i bins to pack all chores in $W_i(A_i \setminus \{e_i^*\})$ and another MMS_i bins to pack all chores in $J_i(A_i \setminus \{e_i^*\}) \cup \{e_i^*\}$. Actually, since all chores in $J_i(A_i \setminus \{e_i^*\}) \cup \{e_i^*\}$ are small for i and at least two small chores can be put into the same bin, i only needs $\lceil \frac{\mathsf{MMS}_i}{2} \rceil$ bins to pack all chores in $J_i(A_i \setminus \{e_i^*\}) \cup \{e_i^*\}$. Therefore, each agent i can use no more than $\frac{3}{2}\mathsf{MMS}_i + 1$ bins to pack all chores allocated to her.

3.4 PROP1 and PROPX

We now discuss other relaxations for proportionality, "proportional up to one item" and "proportional up to any item", which are also widely studied for additive disutilities.

Definition 2 (α-PROP1 and α-PROPX) *An allocation* $\mathbf{X} = (X_1, \ldots, X_n)$ *is* α*-approximate proportional up to one item (*α*-PROP1) if* $v_i(X_i \backslash \{g\}) \leq \alpha \cdot \frac{v_i(M)}{n}$ *for all agents* $i \in N$ *and some chore* $g \in X_i$*. It is* α*-approximate proportional up to any item (*α*-PROPX) if* $v_i(X_i \backslash \{g\}) \leq \alpha \cdot \frac{v_i(M)}{n}$ *for all agents* $i \in N$ *and any chore* $g \in X_i$*. The allocation is PROP1 or PROPX if* $\alpha = 1$*.*

It is easy to see that a PROPX allocation is also PROP1. Although exact PROPX or PROP1 allocation is guaranteed to exist for chores with additive disutilities, when the disutilities are subadditive, no algorithm can be better than n-PROP1 or n-PROPX. Consider an instance with n agents and $n+1$ chores. The disutility function is $v_i(S) = 1$ for all agents $i \in N$ and any non-empty subset $S \subseteq M$. Clearly, the disutility function is subadditive since $v_i(S) + v_i(T) \geq v_i(S \cup T)$ for any $S, T \subseteq M$. By the pigeonhole principle, at least one agent i receives two or more chores in any allocation of M. After removing any chore $g \in X_i$, X_i is still not empty. That is, $v_i(X_i \backslash \{g\}) = 1 = n \cdot \frac{v_i(M)}{n}$ for any $g \in X_i$. This example can be easily extended to the bin packing setting and the job scheduling setting, thus we have the following theorem.

Theorem 5 *For the bin packing setting and the job scheduling setting, no algorithm performs better than n-PROP1 or n-PROPX.*

Proof. For the bin packing setting, consider an instance with n agents and $n+1$ chores. The capacity of each agent's bins is 1, i.e, $c_i = 1$ for any $i \in N$.

Each chore is very tiny so that every agent can pack all chores in just one bin, e.g., $s_{i,j} = \frac{1}{n+1}$ for any $i \in N$ and $j \in M$. Therefore, we have $v_i(M) = 1$ and $\mathsf{PROP}_i = \frac{1}{n}$ for each agent $i \in N$. By the pigeonhole principle, at least one agent i receives two or more chores in any allocation of M. After removing any chore $g \in X_i$, agent i stills needs one bin to pack the remaining chores. Hence, we have $v_i(X_i \setminus \{g\}) = 1 = n \cdot \mathsf{PROP}_i$ for any $g \in X_i$.

For the job scheduling setting, consider an instance with $2n$ agents and $2n+1$ chores where each agent possesses $2n$ machines with the same speed of 1, and the size of each chore is 1 for every agent. It can be easily seen that for every agent i, the maximum completion time of her machines is minimized when assigning two chores to one machine and one chore to each of the remaining $2n-1$ machines. Therefore, we have $v_i(M) = 2$ and $\mathsf{PROP}_i = \frac{2}{2n} = \frac{1}{n}$ for any $i \in N$. Similarly, by the pigeonhole principle, at least one agent i receives two or more chores in any allocation of M. This implies that $v_i(X_i \setminus \{g\}) = 1 = n \cdot \mathsf{PROP}_i$, thus completing the proof. ■

CHAPTER 4

ON HILL'S WORST-CASE GUARANTEE FOR INDIVISIBLE CHORES

In this chapter, we present our work about Hill's share and guarantee for indivisible chores. We first characterise Hill's share, i.e., the exact upper bound of the MinMaxShare values, $\Delta_n^{\oplus}(\alpha; m)$ and $\Delta_n^{\oplus}(\alpha)$. Thorough case-by-case proofs are provided to show the tightness and closedness of our result. We next show that the monotonic cover of Hill's share is the best guarantee that can be achieved in Hill's model for all allocation instances. A polynomial-time algorithm is designed to give each agent a disutility at most the monotonic cover of Hill's share. We also conduct numerical experiments to show Hill's share is very close to the MinMaxShare for the majority of instances.

4.1 Characterising Hill's Share

For any integers $n \geq 2$ and $k \geq 0$, define the following real intervals:

$$D(n,k) = \left(\frac{1}{kn+n+1}, \frac{k+2}{n(k+1)^2+k+2}\right]$$
$$I(n,k) = \left(\frac{k+2}{n(k+1)^2+k+2}, \frac{1}{kn+1}\right].$$

It is not hard to check that all the intervals are well-defined, non-overlapping, and $\bigcup_{k\geq 0}(D(n,k) \cup I(n,k)) = (0,1]$.

Our first main theorem gives the tight characterisation of Hill's share.

Theorem 6 *For any* $0 < \alpha < 1$*,* $n \geq 2$*, and* $m \geq \lceil \frac{1}{\alpha} \rceil$*,*

$$\Delta_n^{\oplus}(\alpha; m) = \begin{cases} \frac{k+2}{k+1} \cdot \frac{1-\alpha}{n}, & \text{if } \alpha \in D(n,k) \text{ and } m \geq kn+n+1, \\ (k+1)\alpha, & \text{if } \alpha \in D(n,k) \text{ and } m \leq kn+n, \\ (k+1)\alpha, & \text{if } \alpha \in I(n,k) \end{cases} \tag{4.1}$$

for any integers $n \geq 2$ *and* $k \geq 0$ *except* $n = 2$ *and simultaneously* $k = 1$*. If* $n = 2$ *and* $k = 1$*,* $\Delta_2^{\oplus}(\frac{1}{3}; 3) = \frac{2}{3}$*,* $\Delta_2^{\oplus}(\alpha; 4) = 2\alpha$ *for* $\alpha \in [\frac{1}{4}, \frac{1}{3}]$*, and* $\Delta_2^{\oplus}(\alpha; 5)$ *is as follows:*

$$\Delta_2^{\oplus}(\alpha; 5) = \begin{cases} \frac{3-3\alpha}{4}, & \text{if } \alpha \in (\frac{1}{5}, \frac{3}{11}], \\ 2\alpha, & \text{if } \alpha \in (\frac{3}{11}, \frac{1}{3}], \end{cases} \tag{4.2}$$

and for $m \geq 6$*,*

$$\Delta_2^{\oplus}(\alpha; m) = \begin{cases} \frac{3-3\alpha}{4}, & \textit{if } \alpha \in (\frac{1}{5}, \frac{7}{27}] \\ \alpha + \frac{2-2\alpha}{5}, & \textit{if } \alpha \in (\frac{7}{27}, \frac{2}{7}] \\ 2\alpha, & \textit{if } \alpha \in (\frac{2}{7}, \frac{1}{3}]. \end{cases} \tag{4.3}$$

Theorem 6 directly implies the result when the number of chores is not restricted, as shown in the following corollary.

Corollary 4 *For any* $0 < \alpha < 1$, $n \geq 2$, $\Delta_n^{\oplus}(\alpha) = \max_{m \geq \lceil \frac{1}{\alpha} \rceil} \Delta_n^{\oplus}(\alpha; m)$.

Actually, Corollary 4 is a special case of Theorem 6 when m is sufficiently large (e.g., $m \geq \lceil \frac{2}{\alpha} \rceil - 1$ by Lemma 2). Recall we illustrated $\Delta_2^{\oplus}(\alpha)$ and $\Delta_3^{\oplus}(\alpha)$ in Fig. 1.1. We observe two interesting and somewhat unintuitive facts about Theorem 6. First, $\Delta_n^{\oplus}(\cdot)$ is not monotone in α, just like Gourvès et al. [93] observed for the problem with goods. To characterise $\Delta_n^{\oplus}(\alpha; m)$, we want to understand the worst-case disutility in $\mathcal{V}(\alpha; m)$, for which the chores can be hardly partitioned into bundles with similar disutilities. Intuitively, when the single-chore disutility gets larger, it becomes harder to find such a balanced partition. However, this turns out to be imprecise. Second, the case of $n = 2$ makes a difference from $n \geq 3$. When $n = 2$ and $k = 1$, there are three steps in $\Delta_n^{\oplus}(\cdot)$: the worst-case MinMaxShare has two increasing intervals with different slops following a decreasing interval. For all the other values of n and k, there are two intervals with one decreasing and the other increasing.

Remark 1 *When* $n = 2$ *the problem of chores and that of goods are the same, since maximising the minimum bundle by partitioning the chores into* *two bundles is equivalent to minimising the maximum bundle. For $n = 2$, Gourvès et al. [93] provided a lower bound of the MaxMinShare for goods which is not tight. It can be verified that $1 - \Delta_2^{\oplus}(\alpha)$ is strictly larger than their bound when $\alpha \in (\frac{1}{5}, \frac{3}{10})$ (Definition 2 in [93]). Thus, as a byproduct, Corollary 4 improves the result in [93] for goods with $n = 2$ by giving the tight worst-case bound, i.e.,*

$$\min_{v \in \mathcal{V}(\alpha)} \max_{\mathbf{A} \in \mathcal{X}_2(M)} \min_{1 \leq \ell \leq 2} v(A_\ell) = 1 - \Delta_2^{\oplus}(\alpha).$$

In Remark 2, we show how to extend this result to two non-identical disutilities.

In the following of this section, we prove Theorem 6. As we have discussed, after m reaches a certain value (e.g., $m \geq \lceil \frac{2}{\alpha} \rceil - 1$ by Lemma 2), Hill's share does not increase anymore, and thus Corollary 4 is a special case of Theorem 6 when m is sufficiently large. Therefore, we first prove Corollary 4, and then carefully discuss Hill's share when m is not sufficiently large, which will complete the proof of Theorem 6 accordingly.

4.1.1 Proof of Corollary 4

We prove Corollary 4 by contradiction, and assume that there exists a disutility $v \in \mathcal{V}(\alpha)$ whose MinMaxShare is larger than $\Delta_n^{\oplus}(\alpha)$. Let $\mathbf{A} = (A_1, \ldots, A_n)$ be a lexicographical MinMax allocation of v; that is, the largest disutility of bundles in $\mathbf{A}$ is the minimised over all allocations, and among these allocations the second largest disutility is minimised, and so on. Without loss of generality, assume $v(A_1) \geq \cdots \geq v(A_n)$ and $v(A_1) = \mathsf{MMS}_n(v) > \Delta_n^{\oplus}(\alpha)$.

Let E_α denote the subset of chores whose disutilities are exactly α, i.e., $E_\alpha = \{e \in M \mid v(e) = \alpha\}$. It can be verified that $\Delta_n^{\oplus}(\alpha) \geq (k+1)\alpha$ (this is also illustrated in Fig. 1.1), which gives $v(A_1) > (k+1)\alpha$. Moreover, since $v(e) \leq \alpha$ for any $e \in M$, $|A_1| \geq k+2$. We have the following property.

Claim 2 *Letting j be an agent in $N \setminus \{1\}$, for any $S_1 \subseteq A_1$ and $S_j \subseteq A_j$ such that $v(S_1) > v(S_j)$, $v(S_1) - v(S_j) \geq v(A_1) - v(A_j)$.*

Proof. For the sake of contradiction, we assume that there exist $S_1' \subseteq A_1$ and $S_{j'}' \subseteq A_{j'}$ such that $v(S_1') > v(S_{j'}')$ and $v(S_1') - v(S_{j'}') < v(A_1) - v(A_{j'})$. Then we construct another allocation $\mathbf{B} = (B_1, \ldots, B_n)$ by exchanging S_1' and $S_{j'}'$, i.e., $B_1 = A_1 \setminus S_1' \cup S_{j'}'$, $B_{j'} = A_{j'} \setminus S_{j'}' \cup S_1'$ and $B_j = A_j$ for any $j \in N \setminus \{1, j'\}$. It follows that $v(B_1) < v(A_1)$, $v(B_{j'}) < v(A_1)$ and $v(B_j) = v(A_j)$ for any $j \in N \setminus \{1, j'\}$, which contradicts the assumption that $\mathbf{A}$ is a lexicographical MinMax allocation of v. ■

The contraposition of Claim 2 gives the following.

Claim 3 *Letting j be an agent in $N \setminus \{1\}$, for any $S_1 \subseteq A_1$ and $S_j \subseteq A_j$ such that $v(A_j \setminus S_j \cup S_1) < v(A_1)$, $v(S_j) \geq v(S_1)$.*

As a warm-up, we start from the case with large α, where $k = 0$, and distinguish two subcases depending on the domain of α.

Case 1: $n \geq 2$ and $k = 0$

Subcase 1.1: $\alpha \in D(n, 0)$

When $\alpha \in D(n, 0)$, $\frac{1}{n+1} < \alpha \leq \frac{2}{n+2}$ and $v(A_1) > \Delta_n^{\oplus}(\alpha) = \frac{2-2\alpha}{n}$. If $E_\alpha \cap A_1 \neq \emptyset$, there exists $e^* \in A_1$ such that $v(e^*) = \alpha < v(A_1)$. Then Claim

3 gives a lower bound of $v(A_j)$ for any $j \in N \setminus \{1\}$, i.e., $v(A_j) \geq v(e^*) = \alpha$. Summing up these lower bounds leads to the following contradiction

$$1 = \sum_{j \in N} v(A_j) > \frac{2-2\alpha}{n} + (n-1) \cdot \alpha = \frac{(n+1)(n-2)\alpha + 2}{n} > 1,$$

where the last inequality is because $\alpha > \frac{1}{n+1}$.

Therefore, $E_\alpha \cap A_1 = \emptyset$. Then by the definition of $\mathcal{V}(\alpha)$, there must exist $j' \in N \setminus \{1\}$ such that $E_\alpha \cap A_{j'} \neq \emptyset$, and thus $v(A_{j'}) \geq \alpha$. Recall that $|A_1| \geq k+2 = 2$, this implies there exists $S \subseteq A_1$ such that $v(A_1) > v(S) \geq \frac{1}{2}v(A_1) > \frac{1-\alpha}{n}$. According to Claim 3, $v(A_j) \geq v(S) > \frac{1-\alpha}{n}$ holds for any $j \in N \setminus \{1, j'\}$. As a result,

$$1 = \sum_{j \in N} v(A_j) > \frac{2-2\alpha}{n} + \alpha + (n-2) \cdot \frac{1-\alpha}{n} = 1,$$

which is also a contradiction. Therefore, $v(A_1) > \Delta_n^{\oplus}(\alpha)$ never holds when $\alpha \in D(n, 0)$.

For the other direction, the disutility function for this subcase (see Table 4.1) contains one chore with disutility α and n chores with disutility $\frac{1-\alpha}{n}$. Since $\frac{1}{n+1} < \alpha \leq \frac{2}{n+2}$, it follows that $\frac{1-\alpha}{n} < \alpha \leq 2 \cdot \frac{1-\alpha}{n}$. Clearly, the MinMaxShare of this disutility function is $2 \cdot \frac{1-\alpha}{n} = \Delta_n^{\oplus}(\alpha)$.

Chore Disutility	Quantity
α	1
$\frac{1-\alpha}{n}$	n

Table 4.1: Disutility function for Subcases 1.1 and 1.2.

Subcase 1.2: $\alpha \in I(n, 0)$

When $\alpha \in I(n, 0)$, by similar reasonings, we can show that $v(A_1) > \Delta_n^{\oplus}(\alpha)$ does not hold, either. In this subcase, $\frac{2}{n+2} < \alpha \le 1$ and $\Delta_n^{\oplus}(\alpha) = \alpha$. If $E_\alpha \cap A_1 \ne \emptyset$, there exists $e^* \in A_1$ such that $v(e^*) = \alpha < v(A_1)$ and Claim 3 gives a lower bound of $v(A_j)$ for any $j \in N \setminus \{1\}$, i.e., $v(A_j) \ge v(e^*) = \alpha$. Summing up these lower bounds leads to the following contradiction

$$1 = \sum_{j \in N} v(A_j) > n\alpha > \frac{2n}{n+2} \ge 1,$$

where the last inequality is because $n \ge 2$.

Therefore, it must hold that $E_\alpha \cap A_1 = \emptyset$ and moreover, there exists $j' \in N \setminus \{1\}$ with $E_\alpha \cap A_{j'} \ne \emptyset$. Thus, $v(A_{j'}) \ge \alpha$. Since $|A_1| \ge k + 2 = 2$, there exists $S \subseteq A_1$ such that $v(A_1) > v(S) \ge \frac{1}{2} v(A_1) > \frac{\alpha}{2}$. According to Claim 3, $v(A_j) \ge v(S) > \frac{\alpha}{2}$ holds for any $j \in N \setminus \{1, j'\}$. As a result, we have

$$1 = \sum_{j \in N} v(A_j) > \alpha + \alpha + (n-2) \cdot \frac{\alpha}{2} = \frac{n+2}{2}\alpha > 1,$$

which is also a contradiction.

For the other direction, the disutility function for this subcase also contains one chore with disutility α and n chores with disutility $\frac{1-\alpha}{n}$ (see Table 4.1). Since $\frac{2}{n+2} < \alpha \le 1$, it follows that $2 \cdot \frac{1-\alpha}{n} < \alpha \le 1$. Clearly, the MinMaxShare of this disutility function is $\alpha = \Delta_n^{\oplus}(\alpha)$. Up to here, the proof regarding the case of $k = 0$ is completed.

Next, we consider the general case of $k \ge 1$ excluding $n = 2$ and simultaneously $k = 1$.

Case 2: $n \geq 3$ **and** $k \geq 1$ **or** $n \geq 2$ **and** $k \geq 2$

For this case, we again start with the subcases when $\alpha \in D(n,k)$. Recall that when $\alpha \in D(n,k)$, $\alpha \in (\frac{1}{(k+1)n+1}, \frac{k+2}{n(k+1)^2+k+2}]$ and $v(A_1) > \Delta_n^{\oplus}(\alpha) = \frac{k+2}{k+1} \cdot \frac{1-\alpha}{n}$.

Subcase 2.1: $\alpha \in D(n,k)$ **and** $E(\alpha) \cap A_j = \emptyset$ **for any** $j \in N \setminus \{1\}$

In this subcase, all chores with disutility α are in A_1, and thus $v(e) < \alpha$ for any $e \in A_j$ and $j \in N \setminus \{1\}$. Due to the normalisation, there exists an agent j_0 who receives disutility at most $\frac{1-v(A_1)}{n-1}$, which gives the following lower bound of the difference between the disutilities that agents 1 and j_0 receive

$$v(A_1) - v(A_{j_0}) \geq \frac{n}{n-1} v(A_1) - \frac{1}{n-1} > \frac{1-(k+2)\alpha}{(n-1)(k+1)}.$$

It can be shown that the rightmost-hand side of the above inequality is no less than $\frac{\alpha}{2}$, which is equivalent to $\alpha \leq \frac{2}{(k+1)n+k+3}$. Since $\alpha \leq \frac{k+2}{n(k+1)^2+k+2}$, it suffices to show $\frac{2}{(k+1)n+k+3} \geq \frac{k+2}{n(k+1)^2+k+2}$, which holds since

$$\begin{aligned}&\frac{2}{(k+1)n+k+3} - \frac{k+2}{n(k+1)^2+k+2} \\ &= \frac{(k+1)(nk-k-2)}{((k+1)n+k+3)(n(k+1)^2+k+2)} \geq 0,\end{aligned}$$

where the last inequality is because $n \geq 3$ and $k \geq 1$, or $n \geq 2$ and $k \geq 2$.

Therefore, $v(A_1) - v(A_{j_0}) > \frac{\alpha}{2}$. Let e^* be a chore in A_1 with disutility α. Since $v(A_1) > (k+1)\alpha > \alpha$, for any $S \subseteq A_{j_0}$ with disutility smaller than α, Claim 2 actually gives a tighter bound of its disutility, i.e., $v(S) \leq v(e^*) - (v(A_1) - v(A_{j_0})) < \frac{\alpha}{2}$. Thus, $v(e) < \frac{\alpha}{2}$ for any $e \in A_{j_0}$. Besides, according to Claim 3, $v(A_{j_0}) \geq v(e^*) = \alpha$. These two facts together imply that there exists $S' \subseteq A_{j_0}$ such that $v(S') \in [\frac{\alpha}{2}, \alpha)$, which is a contradiction to Claim 2.

Subcase 2.2: $\alpha \in D(n,k)$ **and** $E(\alpha) \cap A_{j'} \neq \emptyset$ **for some** $j' \in N \setminus \{1\}$

In this subcase, some chores with disutility α are in $A_{j'}$. Before diving into the proof for this subcase, we present the following claim, which shows the existence of a subset of A_1 whose disutility is within a specific range.

Claim 4 *There exists a subset* $S \subseteq A_1$ *such that* $\frac{k}{k+2}v(A_1) \leq v(S) < v(A_1) - \alpha$.

Proof of Claim 4. When $k = 1$, if there exists $e \in A_1$ such that $v(e) \geq \frac{1}{3}v(A_1)$, recall that $v(A_1) > (k+1)\alpha = 2\alpha$, Claim 4 holds since $v(e) \leq \alpha < v(A_1) - \alpha$. If $v(e) < \frac{1}{3}v(A_1)$ for any $e \in A_1$, denote by (A_1^1, A_1^2) one 2-partition of A_1 that minimises the disutility difference between the two bundles among all 2-partitions. Without loss of generality, we assume $v(A_1^1) \leq v(A_1^2)$, then $v(A_1^1) \leq \frac{1}{2}v(A_1) < v(A_1) - \alpha$. Besides, $v(A_1^1) \geq \frac{1}{3}v(A_1)$ holds. Otherwise, $v(A_1^2) - v(A_1^1) = v(A_1) - 2v(A_1^1) > \frac{1}{3}v(A_1)$, implying that moving a chore from A_1^2 to A_1^1 returns another 2-partition of A_1 that has a smaller disutility difference, which contradicts the definition of (A_1^1, A_1^2).

When $k \geq 2$, we first show that $v(e) > \frac{1}{k+2}\alpha$ for any $e \in A_1$. If not, $v(A_1) > v(A_1 \setminus \{e\}) \geq v(A_1) - \frac{1}{k+2}\alpha$. Then Claim 3 gives $v(A_j) \geq v(A_1) - \frac{1}{k+2}\alpha$ for any $j \in N \setminus \{1\}$. Summing up these lower bounds gives the following inequality

$$1 = \sum_{j \in N} v(A_j) \geq v(A_1) + (n-1)v(A_1) - \frac{n-1}{k+2}\alpha$$
$$> \frac{k+2}{k+1} - \frac{(k+2)^2 + (k+1)(n-1)}{(k+1)(k+2)}\alpha.$$

It can be shown that the rightmost-hand side is at least 1, which constitutes a contradiction. This is equivalent to show that $\alpha \leq \frac{k+2}{(k+2)^2+(k+1)(n-1)}$. Since $\alpha \leq \frac{k+2}{n(k+1)^2+k+2}$, it suffices to show that $\frac{k+2}{(k+1)^2+(k+1)(n-1)} \geq \frac{k+2}{n(k+1)^2+k+2}$, which holds since

$$n(k+1)^2 + k + 2 - ((k+2)^2 + (k+1)(n-1))$$
$$= (k+1)(nk - k - 1) \geq 0,$$

where the last inequality is because $n \geq 2$ and $k \geq 1$.

We then let $S^* = \arg\min_{S \subseteq A_1, v(S) > \alpha} v(S)$ which is guaranteed to exist since $v(A_1) > (k+1)\alpha > \alpha$, and show by contradiction that $v(S^*) \leq \frac{2}{k+2}v(A_1)$. This gives $\frac{k}{k+2}v(A_1) \leq v(A_1 \setminus S^*) < v(A_1) - \alpha$. We assume for the sake of contradiction that $v(S^*) > \frac{2}{k+2}v(A_1)$. Then the definition of S^* gives the following lower bound of $v(e)$ for any $e \in S^*$

$$v(e) > v(S^*) - \alpha > \frac{2}{k+2}v(A_1) - \alpha > \frac{k}{k+2}\alpha \geq \frac{1}{2}\alpha,$$

where the second last inequality is because $v(A_1) > (k+1)\alpha$ and the last inequality is because $k \geq 2$. This lower bound implies that S^* contains exactly 2 chores. Otherwise (i.e., $|S^*| \geq 3$), for any subset $S' \subseteq S^*$ that contains exactly 2 chores, $\alpha < v(S') < v(S^*)$ holds, which contradicts the definition of S^*.

Therefore, we can denote $S^* = \{e^l, e^s\}$ and assume without loss of generality that $v(e^l) \geq v(e^s)$. Accordingly, $v(e^l) \geq \frac{1}{2}v(S^*) > \frac{1}{k+2}v(A_1) > \frac{k+1}{k+2}\alpha$. Recall that $v(e) > \frac{1}{k+2}\alpha$ holds for any $e \in A_1$. These two facts together imply that the total disutility of e^l and any other chore in A_1 is larger than α. From the definition of S^*, we know that $e^s \in \arg\min_{e \in A_1} v(e)$, which gives $v(e) \geq v(e^s) > \frac{k}{k+2}\alpha$ for any $e \in A_1$. Letting S' be the subset of A_1 that contains the two chores with the smallest disutilities, the following inequality leads to a contradiction to the definition of S^*

$$\alpha \leq \frac{2k}{k+2}\alpha < v(S') \leq \frac{2}{k+2}v(A_1) < v(S^*),$$

where the first inequality is because $k \geq 2$ and the second last inequality is because $|A_1| \geq k+2$. ■

We are now ready to reveal the contradiction in the subcase. Denote by e^* one chore in $A_{j'}$ that has disutility α and by S a subset of A_1 that satisfies Claim 4, Claim 3 gives $v(A_{j'} \setminus \{e^*\}) \geq v(S) \geq \frac{k}{k+2}v(A_1)$; that is, $v(A_{j'}) \geq \frac{k}{k+2}v(A_1) + \alpha$. For any $j \in N \setminus \{1, j'\}$, recall that $|A_1| \geq k+2$ which implies that there exists $S' \subseteq A_1$ such that $v(A_1) > v(S') \geq \frac{k+1}{k+2}v(A_1)$, Claim 3 gives $v(A_j) \geq v(S') \geq \frac{k+1}{k+2}v(A_1)$. Summing up these lower bounds leads to the following contradiction

$$\begin{aligned} 1 = \sum_{j \in N} v(A_j) &\geq v(A_1) + \frac{k}{k+2}v(A_1) + \alpha + (n-2) \cdot \frac{k+1}{k+2}v(A_1) \\ &= \frac{n(k+1)}{k+2}v(A_1) + \alpha > 1 - \alpha + \alpha = 1. \end{aligned}$$

For the other direction, the disutility function for the subcases when $\alpha \in D(n,k)$ (see Table 4.2) contains one chore with disutility α and $n(k+1)$ chores with disutility $\frac{1-\alpha}{n(k+1)}$. Since $\alpha > \frac{1}{kn+n+1}$, it follows that $\alpha > \frac{1-\alpha}{n(k+1)}$. Besides, it can be verified that $\alpha < \frac{2-2\alpha}{n(k+1)}$, which is equivalent to $\alpha < \frac{2}{nk+n+2}$. Since $\alpha \leq \frac{k+2}{n(k+1)^2+k+2}$, it suffices to show $\frac{k+2}{n(k+1)^2+k+2} < \frac{2}{n(k+1)+2}$, which holds since

$$\begin{aligned}&\frac{2}{n(k+1)+2} - \frac{k+2}{n(k+1)^2+k+2} \\ &= \frac{nk(k+1)}{(n(k+1)+2)(n(k+1)^2+k+2)} > 0.\end{aligned}$$

By the pigeonhole principle, there exists a bundle that contains at least $k+2$ chores in any allocation. This implies that the MinMaxShare of this disutility function is $(k+2) \cdot \frac{1-\alpha}{n(k+1)}$, which happens in the allocation where one bundle contains $k+2$ chores with disutility $\frac{1-\alpha}{n(k+1)}$, one bundle contains k chores with disutility $\frac{1-\alpha}{n(k+1)}$ and one chore with disutility α, and each of the other bundles contains $k+1$ chores with disutility $\frac{1-\alpha}{n(k+1)}$.

Chore Disutility	Quantity
α	1
$\frac{1-\alpha}{n(k+1)}$	$n(k+1)$

Table 4.2: Disutility function for subcases $\alpha \in D(n,k)$ with $n \geq 3$ and $k \geq 1$, or $n \geq 2$ and $k \geq 2$.

Next we consider the subcases when $\alpha \in I(n,k)$. Recall that when $\alpha \in I(n,k)$, $\alpha \in (\frac{k+2}{n(k+1)^2+k+2}, \frac{1}{kn+1}]$ and $v(A_1) > \Delta_n^{\oplus}(\alpha) = (k+1)\alpha$.

Subcase 2.3: $\alpha \in I(n,k)$ **and** $E(\alpha) \cap A_j = \emptyset$ **for any** $j \in N \setminus \{1\}$

For this subcase, we first derive a lower bound of $v(A_j)$ for any $j \in N \setminus \{1\}$, i.e., $v(A_j) \geq (\frac{(k+1)^2}{k+2} + \frac{1}{(n-1)(k+2)})\alpha$. Letting $D = \frac{(k+1)^2}{k+2} + \frac{1}{(n-1)(k+2)}$, we assume for the sake of contradiction that $v(A_{j'}) < D\alpha$ for some $j' \in N \setminus \{1\}$. It can be verified that $k < D < k+1$, where the first inequality is equivalent to $n > 0$, and the second inequality is equivalent to $(n-1)(k+1) > 1$. Denote by e^* one chore in A_1 with disutility α and by e' any chore in A_j, we have

$$v(A_{j'} \setminus (A_{j'} \setminus \{e'\}) \cup (A_1 \setminus \{e^*\})) = v(A_1 \setminus \{e^*\} \cup \{e'\}) < v(A_1).$$

Then from Claim 3, $v(A_{j'} \setminus \{e'\}) \geq v(A_1 \setminus \{e^*\})$, which gives

$$v(e') \leq v(A_{j'}) - v(A_1) + v(e^*) < D\alpha - (k+1)\alpha + \alpha = (D-k)\alpha.$$

However, we next show that the disutility of some chore in $A_{j'}$ must be larger than $(D-k)\alpha$, which leads to a contradiction. To achieve this, we denote $S^* \in \arg\min_{S \subseteq A_{j'}, v(S) > (D-1)\alpha} v(S)$, whose existence is guaranteed since Claim 3 gives $v(A_{j'}) \geq v(A_1 \setminus \{e^*\}) > k\alpha > (D-1)\alpha$. Notice that

$$v(A_{j'} \setminus S^* \cup (A_1 \setminus \{e^*\})) < D\alpha - (D-1)\alpha + v(A_1) - \alpha = v(A_1),$$

from Claim 3, $v(S^*) \geq v(A_1 \setminus \{e^*\}) > k\alpha$. Then the definition of S^* implies that the disutility of any chore in S^* is at least

$$v(S^*) - (D-1)\alpha > (k-D+1)\alpha \geq (D-k)\alpha,$$

where the last inequality is equivalent to $D - k - \frac{1}{2} = \frac{k+2-kn}{2(n-1)(k+2)} \leq 0$, which holds when $n \geq 3$ and $k \geq 1$, or $n \geq 2$ and $k \geq 2$.

Therefore, $v(A_j) \geq (\frac{(k+1)^2}{k+2} + \frac{1}{(n-1)(k+2)})\alpha$ holds for any $j \in N \setminus \{1\}$. Summing up these lower bounds leads to the following contradiction

$$\begin{aligned} 1 = \sum_{j \in N} v(A_j) &> (k+1)\alpha + (n-1) \cdot (\frac{(k+1)^2}{k+2} + \frac{1}{(n-1)(k+2)})\alpha \\ &= \frac{n(k+1)^2 + k + 2}{k+2}\alpha > 1. \end{aligned}$$

Subcase 2.4: $\alpha \in I(n,k)$ **and** $E(\alpha) \cap A_{j'} \neq \emptyset$ **for some** $j' \in N \setminus \{1\}$

The proof is similar to that of Subcase 2.2. First, it can be verified that Claim 4 still holds.

Proof of Claim 4 for $\alpha \in I(n,k)$**.** Notice that Claim 4 holds as long as $k = 1$, thus, we can focus on $k \geq 2$. We first show that $v(e) > \frac{1}{k+2}\alpha$ for any $e \in A_1$. If not, $v(A_1 \setminus \{e\}) \geq v(A_1) - \frac{1}{k+2}\alpha$. Then Claim 3 gives $v(A_j) \geq v(A_1) - \frac{1}{k+2}\alpha$ for any $j \in N \setminus \{1\}$. Summing up these lower bounds gives the following formula

$$1 = \sum_{j \in N} v(A_j) \geq v(A_1) + (n-1)v(A_1) - \frac{n-1}{k+2}\alpha > \frac{n(k+1)(k+2) - n + 1}{k+2}\alpha.$$

It can be shown that the rightmost-hand side of the above inequality is at least 1, which is a contradiction. This is equivalent to show that $\alpha \geq \frac{k+2}{n(k+1)(k+2)-n+1}$. Since $\alpha \geq \frac{k+2}{n(k+1)^2+k+2}$, it suffices to show that $\frac{k+2}{n(k+1)(k+2)-n+1} \leq \frac{k+2}{n(k+1)^2+k+2}$, which holds since

$$n(k+1)(k+2) - n + 1 - (n(k+1)^2 + k + 2) = nk - k - 1 \geq 0$$

where the last inequality is because $n \geq 2$ and $k \geq 1$.

We then let $S^* = \arg\min_{S \subseteq A_1, v(S) > \alpha} v(S)$, which is guaranteed to exist since $v(A_1) > (k+1)\alpha > \alpha$. By the same proof as the counterpart in the proof of Claim 4 for $\alpha \in D(n,k)$, we can show that $v(S^*) \leq \frac{2}{k+2} v(A_1)$, which gives $\frac{k}{k+2} v(A_1) \leq v(A_1 \setminus S^*) < v(A_1) - \alpha$. ■

We are now ready to reveal the contradiction in this subcase. Denote by e^* one chore in $A_{j'}$ that has disutility α and by S a subset of A_1 that satisfies Claim 4, Claim 3 gives $v(A_{j'} \setminus \{e^*\}) \geq v(S) \geq \frac{k}{k+2} v(A_1)$; that is, $v(A_{j'}) \geq \frac{k}{k+2} v(A_1) + \alpha$. For any $j \in N \setminus \{1, j'\}$, recall that $|A_1| \geq k+2$ which implies that there exists $S' \subseteq A_1$ such that $v(A_1) > v(S') \geq \frac{k+1}{k+2} v(A_1)$, Claim 3 gives $v(A_j) \geq v(S') \geq \frac{k+1}{k+2} v(A_1)$. Summing up these lower bounds leads to the following contradiction

$$\begin{aligned} 1 = \sum_{j \in N} v(A_j) &\geq v(A_1) + \frac{k}{k+2} v(A_1) + \alpha + (n-2) \cdot \frac{k+1}{k+2} v(A_1) \\ &= \frac{n(k+1)}{k+2} v(A_1) + \alpha > \frac{n(k+1)^2 + k + 2}{k+2} \alpha > 1. \end{aligned}$$

For the other direction, the disutility function for the subcases when $\alpha \in I(n,k)$ (See Table 4.3) containing $kn+1$ chores with disutility α and $n-1$ chores with disutility $\frac{1-(nk+1)\alpha}{n-1}$. It can be verified that $\alpha > \frac{1-(kn+1)\alpha}{n-1}$, which is equivalent to $\alpha > \frac{1}{(k+1)n}$. Since $\alpha > \frac{k+2}{n(k+1)^2+k+2}$, it suffices to show $\frac{k+2}{n(k+1)^2+k+2} \geq \frac{1}{(k+1)n}$, which holds since

$$\frac{k+2}{n(k+1)^2 + k + 2} - \frac{1}{(k+1)n} = \frac{(k+1)n - k - 2}{(n(k+1)^2 + k + 2)(k+1)n} \geq 0,$$

where the inequality is because $n \geq 3$ and $k \geq 1$, or $n \geq 2$ and $k \geq 2$. By the pigeonhole principle, there exists a bundle that contains at least

$k+1$ chores with disutility α. This implies that the MinMaxShare of this disutility function is $(k+1)\alpha$, which happens in the allocation where one bundle contains $k+1$ chores with disutility α, and each of the other bundles contains k chores with disutility α and one chore with disutility $\frac{1-(nk+1)\alpha}{n-1}$.

Chore Disutility	Quantity
α	$kn+1$
$\frac{1-(nk+1)\alpha}{n-1}$	$n-1$

Table 4.3: Disutility function for subcases $\alpha \in I(n,k)$ with $n \geq 3$ and $k \geq 1$, or $n \geq 2$ and $k \geq 2$.

Case 3: $n = 2$ **and** $k = 1$

We now prove Corollary 4 for the case of $n = 2$ and $k = 1$ (i.e., $\alpha \in D(2,1) \cup I(2,1)$), which makes a difference from the other cases and requires a more involved analysis.

Subcase 3.1: $\alpha \in (\frac{1}{5}, \frac{7}{27}]$

When $\alpha \in (\frac{1}{5}, \frac{7}{27}]$, $v(A_1) > \Delta_2^{\oplus}(\alpha) = \frac{3-3\alpha}{4}$. If $E_\alpha \cap A_2 \neq \emptyset$, A_2 contains some chores with disutility α. Notice that Claim 4 holds as long as $k = 1$, thus there exists $S \subseteq A_1$ such that $\frac{1}{3}v(A_1) \leq v(S) < v(A_1) - \alpha$. Denote by e^* one chore in A_2 with disutility α, Claim 3 gives $v(A_2 \backslash \{e^*\}) \geq v(S) \geq \frac{1}{3}v(A_1)$. As a result, we have

$$1 = v(A_1) + v(A_2) \geq v(A_1) + \frac{1}{3}v(A_1) + \alpha,$$

which gives $v(A_1) \leq \frac{3-3\alpha}{4}$, thus contradicting the assumption that $v(A_1) > \Delta_n^{\oplus}(\alpha)$.

Therefore, $E_\alpha \cap A_2 = \emptyset$, which means that all chores with disutility α are in A_1 and for any $e \in A_2$, $v(e) < \alpha$. We first derive an upper bound and a lower bound of the maximum disutility of the chores in A_2. Denote by e^* one chore in A_1 with $v(e^*) = \alpha < v(A_1)$, since $v(A_1) - v(A_2) = 2v(A_1) - 1 > \frac{1-3\alpha}{2}$, Claim 2 gives

$$\max_{e \in A_2} v(e) \leq v(e^*) - (v(A_1) - v(A_2)) < \frac{5\alpha - 1}{2}.$$

Notice that $\frac{1-3\alpha}{2} > \frac{\alpha}{3}$ since $\alpha \leq \frac{7}{27} < \frac{3}{11}$, $v(A_1) - v(A_2) > \frac{\alpha}{3}$. Then for every $S \subseteq A_2$ with $v(S) < \alpha$, Claim 2 actually gives a tighter bound of $v(S)$, i.e., $v(S) \leq v(e^*) - (v(A_1) - v(A_2)) < \frac{2}{3}\alpha$. This also implies that for every $S' \subseteq A_2$ with $v(S') \geq \frac{2}{3}\alpha$, $v(S') \geq \alpha$ actually holds. Let $S^* = \arg\min_{S \subseteq A_2, v(S) \geq \frac{2}{3}\alpha} v(S)$ whose existence is guaranteed since Claim 3 gives $v(A_2) \geq v(e^*) = \alpha$, thus, $v(S^*) \geq \alpha$. Then from the definition of S^*, $v(e) \geq v(S^*) - \frac{2}{3}\alpha \geq \frac{1}{3}\alpha$ holds for any $e \in A_2$, which implies

$$\max_{e \in A_2} v(e) \geq \frac{\alpha}{2}.$$

Otherwise (i.e., $\max_{e \in A_2} v(e) < \frac{\alpha}{2}$), the total disutility of any two chores in A_2 is at least $\frac{2}{3}\alpha$ and smaller than α, which is a contradiction to Claim 2.

We then show that $|A_1|$ is exactly 3. Otherwise (i.e., $|A_1| \geq 4$), there exists $S \subseteq A_1$ such that $v(A_1) > v(S) \geq \alpha + \frac{2}{3}(v(A_1) - \alpha)$. Then Claim 3 gives $v(A_2) \geq v(S) \geq \alpha + \frac{2}{3}(v(A_1) - \alpha)$. Summing up the lower bounds of

$v(A_1)$ and $v(A_2)$ leads to a contradiction as below

$$1 = v(A_1) + v(A_2) \geq \frac{5}{3}v(A_1) + \frac{1}{3}\alpha > \frac{15 - 11\alpha}{12} > 1,$$

where the last inequality is because $\alpha \leq \frac{7}{27} < \frac{3}{11}$. Therefore, we can denote $A_1 = \{e^1_1, e^1_2, e^1_3\}$ and assume without loss of generality that $v(e^1_1) = \alpha \geq v(e^1_2) = x \geq v(e^1_3) = y$. We then derive the lower bounds of x and y, and reveal the contradiction in this subcase. Since $x \geq y$, the following formula holds

$$x \geq \frac{x + y}{2} = \frac{v(A_1) - \alpha}{2} > \frac{3 - 7\alpha}{8} \geq \frac{5\alpha - 1}{2} > \max_{e \in A_2} v(e),$$

where the second last inequality is because $\alpha \leq \frac{7}{27}$. Then Claim 2 gives the following lower bound of x

$$x \geq \max_{e \in A_2} v(e) + (v(A_1) - v(A_2)) > \frac{\alpha}{2} + \frac{1 - 3\alpha}{2} = \frac{1 - 2\alpha}{2}.$$

Claim 2 also gives $y \geq v(A_1) - v(A_2)$. Notice that

$$2 \cdot (v(A_1) - v(A_2)) > \frac{2 - 6\alpha}{2} > \alpha - \frac{1 - 3\alpha}{2} > x - (v(A_1) - v(A_2)),$$

where the second inequality is because $\alpha \leq \frac{7}{27} < \frac{3}{11}$, we have the following lower bound of y

$$y > \frac{1}{2} \cdot (x - (v(A_1) - v(A_2))) \geq \frac{1}{2} \cdot \max_{e \in A_2} v(e) \geq \frac{\alpha}{4}.$$

Therefore, $v(A_1) = \alpha + x + y > \alpha + \frac{1-2\alpha}{2} + \frac{\alpha}{4} = \frac{2+\alpha}{4}$, which gives $v(A_1) - v(A_2) = 2v(A_1) - 1 > \frac{\alpha}{2}$. However, according to Claim 2, $v(A_1) - v(A_2) \leq \alpha - \max_{e \in A_2} v(e) \leq \frac{\alpha}{2}$, thus constituting a contradiction.

For the other direction, the disutility function for this subcase contains one chore with disutility α and four chores with disutility $\frac{1-\alpha}{4}$. Since $\frac{1}{5} < \alpha \leq \frac{7}{27}$, it follows that $\frac{1-\alpha}{4} < \alpha < 2 \cdot \frac{1-\alpha}{4}$, where the last inequality is because $\alpha \leq \frac{7}{27} < \frac{1}{3}$. Clearly, the MinMaxShare of this disutility function is $3 \cdot \frac{1-\alpha}{4}$.

Subcase 3.2: $\alpha \in (\frac{7}{27}, \frac{2}{7}]$

When $\alpha \in (\frac{7}{27}, \frac{2}{7}]$, $v(A_1) > \Delta_2^{\oplus}(\alpha) = \frac{2+3\alpha}{5}$. If $E_\alpha \cap A_2 \neq \emptyset$, the proof is similar to that for the counterpart in Subcase 3.1. That is, we also have $v(A_1) \leq \frac{3-3\alpha}{4}$, which contradicts $v(A_1) > \Delta_n^{\oplus}(\alpha)$ since $\frac{3-3\alpha}{4} < \frac{2+3\alpha}{5}$ when $\alpha > \frac{7}{27}$.

Therefore, we can focus on $E_\alpha \cap A_2 = \emptyset$. We first derive an upper bound and a lower bound of the maximum disutility of the chores in A_2, which is similar to the counterpart of Subcase 3.1. Denote by e^* one chore in A_1 with $v(e^*) = \alpha < v(A_1)$, since $v(A_1) - v(A_2) = 2v(A_1) - 1 > \frac{6\alpha-1}{5}$, Claim 2 gives

$$\max_{e \in A_2} v(e) \leq v(e^*) - (v(A_1) - v(A_2)) < \frac{1-\alpha}{5}.$$

Notice that $\frac{6\alpha-1}{5} > \frac{\alpha}{3}$ since $\alpha > \frac{7}{27} > \frac{3}{13}$, $v(A_1) - v(A_2) > \frac{\alpha}{3}$. Then for every $S \subseteq A_2$ with $v(S) < \alpha$, Claim 2 actually gives a tighter bound of $v(S)$, i.e., $v(S) \leq v(e^*) - v(A_1) - v(A_2) < \frac{2}{3}\alpha$. This also implies that for every $S' \subseteq A_2$ with $v(S') \geq \frac{2}{3}\alpha$, $v(S') \geq \alpha$ actually holds. Let $S^* = \arg\min_{S \subseteq A_2, v(S) \geq \frac{2}{3}\alpha} v(S)$ whose existence is guaranteed since Claim 3 gives $v(A_2) \geq v(e^*) = \alpha$, thus, $v(S^*) \geq \alpha$. Then from the definition of S^*,

$v(e) \geq v(S^*) - \frac{2}{3}\alpha \geq \frac{1}{3}\alpha$ holds for any $e \in A_2$, which implies

$$\max_{e \in A_2} v(e) \geq \frac{\alpha}{2}.$$

Otherwise (i.e., $\max_{e \in A_2} v(e) < \frac{\alpha}{2}$), the total disutility of any two chores in A_2 is at least $\frac{2}{3}\alpha$ and smaller than α, which is a contradiction to Claim 2.

Observe that A_1 contains exactly one chore with disutility α. Otherwise (i.e., A_1 contains at least two chores with disutility α), Claim 3 gives $v(A_2) \geq 2\alpha$ which leads to the following contradiction

$$1 = v(A_1) + v(A_2) > \frac{2 + 3\alpha}{5} + 2\alpha > 1,$$

where the last inequality is because $\alpha > \frac{7}{27} > \frac{3}{13}$. Recall that $|A_1| \geq 3$, A_1 contains at least two chores with disutility smaller than α. For each of such chores, we call it a *medium chore* if its disutility is larger than $\max_{e \in A_2} v(e)$. Otherwise, we call it a *small chore*. Then Claim 2 gives the following lower bound of the disutility of any medium chore e

$$\begin{aligned} v(e) &\geq \max_{e \in A_2} v(e) - (v(A_1) - v(A_2)) \\ &= \max_{e \in A_2} v(e) - (2v(A_1) - 1) > \frac{\alpha}{2} - \frac{6\alpha - 1}{5} = \frac{17\alpha - 2}{10}, \end{aligned}$$

as well as the following lower bound of the disutility of any small chore e'

$$v(e') \geq v(A_1) - v(A_2) = 2v(A_1) - 1 > \frac{6\alpha - 1}{5}.$$

We then reveal the contradiction by considering possible combinations of chores in A_1 and showing that no possible combination exists.

Combination 1: besides the chore with disutility α, A_1 also contains at least three small chores. Thus, $v(A_1) > \alpha + 3 \cdot \frac{6\alpha-1}{5} = \frac{23\alpha-3}{5}$. Then a lower bound of the difference between $v(A_1)$ and $v(A_2)$ is

$$v(A_1) - v(A_2) = 2v(A_1) - 1 > \frac{46\alpha - 11}{5} > \frac{\alpha}{2},$$

where the last inequality is because $\alpha > \frac{7}{27} > \frac{22}{87}$. However, according to Claim 2, $v(A_1) - v(A_2) \leq \alpha - \max_{e \in A_2} v(e) \leq \frac{\alpha}{2}$, which is a contradiction. Note that this also implies that except the chore with disutility α, the total disutility of the other chores can not exceed that of three small chores. Since the total disutility of one medium chore and one small chore is larger than

$$\frac{17\alpha - 2}{10} + \frac{6\alpha - 1}{5} = \frac{29\alpha - 4}{10} > \frac{18\alpha - 3}{5} = 3 \cdot \frac{6\alpha - 1}{5},$$

where the inequality is because $\alpha < \frac{2}{7}$, the only combination that remains to consider is that A_1 contains two small chores besides the chore with disutility α.

Combination 2: besides the chore with disutility α, A_1 contains two small chores. From the definition of small chore, $v(e') \leq \max_{e \in A_2} v(e) < \frac{1-\alpha}{5}$ holds for any small chore $e' \in A_1$. Thus, $v(A_1) < \alpha + 2 \cdot \frac{1-\alpha}{5} = \frac{2+3\alpha}{5}$, which is a contradiction to the assumption that $v(A_1) > \Delta_2^{\oplus}(\alpha)$.

For the other direction, the disutility function for this subcase contains one chore with disutility α and five chores with disutility $\frac{1-\alpha}{5}$. Since $\frac{1}{6} < \frac{7}{27} < \alpha \leq \frac{2}{7}$, it follows that $\frac{1-\alpha}{5} < \alpha \leq 2 \cdot \frac{1-\alpha}{5}$. Clearly, the MinMaxShare of this disutility function is $\alpha + 2 \cdot \frac{1-\alpha}{5}$.

Subcase 3.3: $\alpha \in (\frac{2}{7}, \frac{1}{3}]$

When $\alpha \in (\frac{2}{7}, \frac{1}{3}]$, $v(A_1) > \Delta_2^{\oplus}(\alpha) = 2\alpha$. If $E_\alpha \cap A_2 \neq \emptyset$, the proof is similar to those for the counterparts of Subcases 3.1 and 3.2. That is, we also have $v(A_1) \leq \frac{3-3\alpha}{4}$, which contradicts $v(A_1) > \Delta_2^{\oplus}(\alpha)$ since $\frac{3-3\alpha}{4} < 2\alpha$ when $\alpha > \frac{2}{7} > \frac{3}{11}$.

Then we focus on $E_\alpha \cap A_2 = \emptyset$. Since $|A_1| \geq 3$, there exists $S \subseteq A_1$ such that $\alpha + \frac{1}{2}(v(A_1) - \alpha) \leq v(S) < v(A_1)$. From Claim 3, we have a lower bound of $v(A_2)$, i.e., $v(A_2) \geq \alpha + \frac{1}{2}(v(A_1) - \alpha)$. Summing up the lower bounds of $v(A_1)$ and $v(A_2)$ leads to a contradiction,

$$1 = v(A_1) + v(A_2) \geq \frac{3}{2} v(A_1) + \frac{\alpha}{2} > \frac{7\alpha}{2} > 1,$$

where the last inequality is because $\alpha > \frac{2}{7}$.

For the other direction, the disutility function for this subcase contains three chores with disutility α and one chore with disutility $1 - 3\alpha$ (if $\alpha < \frac{1}{3}$). Since $\alpha > \frac{2}{7} > \frac{1}{4}$, it follows that $1 - 3\alpha < \alpha$. Clearly, the MinMaxShare is 2α.

Up to here, we have computed Hill's share for unrestricted m.

4.1.2 Proof of Theorem 6

We now carefully discuss Hill's share when m is not sufficiently large, which completes the proof of Theorem 6. For the sake of contradiction, we assume that there exists a disutility $v \in \mathcal{V}(\alpha; m)$ such that $\mathsf{MMS}_n(v) > \Delta_n^{\oplus}(\alpha; m)$, and let $\mathbf{A} = (A_1, \ldots, A_n)$ be an allocation that gives the MinMaxShare of v. Without loss of generality, assume $v(A_1) \geq \cdots \geq v(A_n)$. We now split the proof into several cases based on the values of n and k, and it suffices to compute the share for the case where m is smaller than the number of chores in the worst-case disutility function in the unrestricted setting.

Case 1: $n \neq 2$ **or** $k \neq 1$

We consider the subcases $\alpha \in D(n,k)$ and $\alpha \in I(n,k)$, separately.

Subcase 1.1: $\alpha \in D(n,k)$

Recall that when $\alpha \in D(n,k)$ with $n \neq 2$ or $k \neq 1$, the disutility function constructed in the setting when m is not restricted contains $kn+n+1$ chores (see Tables 4.1 and 4.2). Therefore, if $m \geq kn+n+1$, the tight bound remains unchanged.

Thus we can focus on $m \leq kn+n$. Since $v(A_1) > \Delta_n^{\oplus}(\alpha; m) = (k+1)\alpha$, by Claim 3, $v(A_j) \geq v(A_1) - \alpha > k\alpha$ for any $j \in N \setminus \{1\}$. Moreover, since the disutility of any chore is at most α, A_1 contains at least $k+2$ chores and A_j contains at least $k+1$ ones, i.e., $|A_1| \geq k+2$ and $|A_j| \geq k+1$. Accordingly, the total number of chores is at least $k+2+(n-1)(k+1) = kn+n+1 > m$, a contradiction. The disutility function that shows tightness (see Table 4.4) contains $\lceil \frac{1}{\alpha} \rceil - 1$ chores with disutility α, one chore with disutility $1 - (\lceil \frac{1}{\alpha} \rceil - 1)\alpha$, and $m - \lceil \frac{1}{\alpha} \rceil$ chores with disutility 0. This disutility function is valid since $m \geq \lceil \frac{1}{\alpha} \rceil$. Since $\alpha \in D(n,k)$, $\frac{1}{\alpha} \geq \frac{n(k+1)^2+k+2}{k+2} \geq kn+1$, where the last inequality is because $n \geq 0$. Therefore, the disutility function contains at least $kn+1$ chores with disutility α By the pigeonhole principle, the MinMaxShare is at least $(k+1)\alpha$.

Chore Disutility	Quantity
α	$\lceil\frac{1}{\alpha}\rceil - 1$
$1 - (\lceil\frac{1}{\alpha}\rceil - 1)\alpha$	1
0	$m - \lceil\frac{1}{\alpha}\rceil$

Table 4.4: Disutility function for subcase $\alpha \in D(n,k)$ with $n \neq 2$ or $k \neq 1$, and $m \leq kn + n$.

Subcase 1.2: $\alpha \in I(n,k)$

The bound for $\alpha \in I(n,k)$ remains unchanged regardless of the value of m, since there always exists a disutility function whose MinMaxShare is at least $\Delta_n^{\oplus}(\alpha; m) = (k+1)\alpha$. Specifically, the disutility function (see Table 4.4) also contains $\lceil\frac{1}{\alpha}\rceil - 1$ chores with disutility α, one chore with disutility $1 - (\lceil\frac{1}{\alpha}\rceil - 1)\alpha$, and $m - \lceil\frac{1}{\alpha}\rceil$ chores with disutility 0. Since $\alpha \in I(n,k)$, $\frac{1}{\alpha} \geq kn+1$, which means that there are at least $kn+1$ chores with disutility α. By the pigeonhole principle, the MinMaxShare is at least $(k+1)\alpha$.

Case 2: $n = 2$ **and** $k = 1$

Recall that when $n = 2$ and $k = 1$, $\alpha \in (\frac{1}{5}, \frac{1}{3}]$, thus $m \geq \lceil\frac{1}{\alpha}\rceil \geq 3$. We prove for this case by considering different values of m and α. When $m = 3$, α can only be $\frac{1}{3}$. The tight bound remains unchanged (i.e., $\Delta_2^{\oplus}(\frac{1}{3}; 3) = \Delta_2^{\oplus}(\frac{1}{3})$), since the disutility function constructed in the unrestricted setting (i.e., Subcase 3.3 in Subsection 4.1.1) contains 3 chores when $\alpha = \frac{1}{3}$.

When $m = 4$, $\alpha \in [\frac{1}{4}, \frac{1}{3})$. Since $v(A_1) > \Delta_2^{\oplus}(\alpha; 4) = 2\alpha$, by Claim 3, $v(A_2) > \alpha$. Therefore, A_1 contains at least three chores and A_2 contains at least 2 chores, a contradiction to $m = 4$. For the tightness, the disutility function contains $\lceil\frac{1}{\alpha}\rceil - 1$ chores with disutility α, and one chore with disutility $1 - (\lceil\frac{1}{\alpha}\rceil - 1)\alpha$. Sine $\frac{1}{\alpha} > 3$, by the pigeonhole principle, the MinMaxShare is at least 2α.

When $m = 5$, $\alpha \in (\frac{1}{5}, \frac{1}{3}]$. If $\alpha \in (\frac{1}{5}, \frac{7}{27}]$ or $(\frac{2}{7}, \frac{1}{3}]$, the disutility functions constructed in the unrestricted setting (i.e., Subcases 3.1 and 3.3 in Subsection 4.1.1) contain 5 and 4 chores respectively, thus the tight bounds do not change. If $\alpha \in (\frac{7}{27}, \frac{2}{7}]$, since $v(A_1) > \Delta_2^{\oplus}(\alpha; 5) \geq 2\alpha$, by Claim 3, $v(A_2) > \alpha$, thus A_1 contains at least 3 chores and A_2 contains at least 2 chores. More accurately, since $m = 5$, $|A_1|$ is exactly 3 and $|A_2|$ is exactly 2. Moreover, it can be verified that the largest disutility in A_1 is at most the smallest disutility in A_2. Since otherwise, by exchanging one chore in A_1 with a strictly larger disutility and one chore in A_2 with a strictly smaller disutility, one can get another allocation $\mathbf{A}' = (A'_1, A'_2)$ such that $v(A'_1) < v(A_1)$ and $v(A'_2) \leq 2\alpha < v(A_1)$. Letting $A_2 = \{e_1, e_2\}$, it follows that $v(e_1) = \alpha$ and $v(e_2) \geq \frac{1}{3} \cdot v(A_1)$. Therefore,

$$v(A_1 \cup A_2) > v(A_1) + \alpha + \frac{1}{3}v(A_1) = \frac{4}{3} \cdot v(A_1) + \alpha.$$

If $\alpha \in (\frac{7}{27}, \frac{3}{11}]$, $v(A_1) > \Delta_2^{\oplus}(\alpha; 5) = \frac{3-3\alpha}{4}$, thus

$$v(A_1 \cup A_2) > \frac{4}{3} \cdot \frac{3 - 3\alpha}{4} + \alpha = 1,$$

a contradiction. If $\alpha \in (\frac{3}{11}, \frac{2}{7}]$, $v(A_1) > \Delta_2^{\oplus}(\alpha; 5) = 2\alpha$, also a contradiction since

$$v(A_1 \cup A_2) > \frac{11}{3}\alpha > 1.$$

The disutility function that shows tightness for $\alpha \in (\frac{7}{27}, \frac{3}{11}]$ is the same as that in Subcase 3.1 in Subsection 4.1.1, i.e., one chore with disutility

α and four chores with disutility $\frac{1-\alpha}{4}$. Again, since $\frac{1}{5} < \frac{7}{27} < \alpha \leq \frac{3}{11} < \frac{1}{3}$, $\frac{1-\alpha}{4} < \alpha < 2 \cdot \frac{1-\alpha}{4}$, which gives that the MinMaxShare is $\frac{3-3\alpha}{4}$. For $\alpha \in (\frac{3}{11}, \frac{2}{7}]$, the disutility function is the same as that in Subcase 3.3 in Subsection 4.1.1, i.e., three chores with disutility α and one chore with disutility $1-3\alpha$. Since $\alpha > \frac{3}{11} > \frac{1}{4}$, $1-3\alpha < \alpha$, thus the MinMaxShare is 2α.

When $m \geq 6$, $\alpha \in (\frac{1}{5}, \frac{1}{3}]$. Since the disutility functions constructed in the subcases of the unrestricted setting contain no more than 6 chores, thus the tight bounds remain unchanged.

4.2 Hill's Guarantee for Indivisible Chores

We next prove the counterpart result of Hill's guarantee for indivisible chores. Consider the general case, where each one of the n agents now has an arbitrary disutility v_i in $\mathcal{A}dd(M)$ (by convention $m = |M|$). Given m and n, a *guarantee* specifies an upper bound $\Gamma_n(v_i; m)$ on agent i's disutility when she shares the m chores with $n-1$ other agents of unknown disutilities in $\mathcal{A}dd(M)$. By construction the mapping Γ_n is the same for every agent i. As part of its definition, a guarantee must be feasible: for any profile $(v_i)_{i=1}^n \in [\mathcal{A}dd(M)]^n$ there exists an allocation $(A_1, \ldots, A_n)$ of M such that

$$v_i(A_i) \leq \Gamma_n(v_i; m) \text{ for all } 1 \leq i \leq n. \tag{4.4}$$

We know from [21] and [79] that the MinMaxShare value $\mathsf{MMS}_n(v_i)$ is not a guarantee because at some (rare!) profiles no allocation meets all inequalities in (4.4). By applying Inequalities (4.4) to an arbitrary guarantee

Γ_n at the unanimous profile $v_i = v$ for all i, we see it is lower bounded by the MinMaxShare:

$$\Gamma_n(v;m) \geq \mathsf{MMS}_n(v) \text{ for all } v \in \mathcal{A}dd(M).$$

Our second main theorem shows that the monotone hull of $\Delta_n^{\oplus}$ serves as the best guarantee in Hill's model. Recall that $\mathcal{U}(\alpha;m)$ contains all the disutility functions $v(\cdot)$ on chores M such that $\max_{e\in M} v(e) \leq \alpha$, and $\mathcal{U}(\alpha) = \bigcup_m \mathcal{U}(\alpha;m)$. For simplicity in the presentation and analysis, we ignore the restriction of the number of chores m, and the result can be extended to the setting with parameter m using the same approach in the first part of our work. The definition of $\mathcal{U}(\alpha)$ is the same as in [100, 117, 93]. Note that $\mathcal{U}(\alpha') \subseteq \mathcal{U}(\alpha)$ if $\alpha' \leq \alpha$, and the difference between $\mathcal{V}(\alpha)$ and $\mathcal{U}(\alpha)$ is that the disutilities in $\mathcal{U}(\alpha)$ do not require that there must be one chore with disutility α. It is straightforward that the tight guarantee regarding $\mathcal{U}(\cdot)$ must be monotone non-decreasing since any worst-case disutility in $\mathcal{U}(\beta)$ is also a disutility in $\mathcal{U}(\alpha)$ for $\beta \leq \alpha$. We write V_n the monotone hull of $\Delta_n^{\oplus}$

$$V_n(\alpha) = \max_{0 \leq \beta \leq \alpha} \Delta_n^{\oplus}(\beta),$$

as illustrated in Fig. 4.1 when $n = 2, 3$. In more detail, we have the following formula of V_n:

$$V_n(\alpha) = \begin{cases} \frac{k+2}{(k+1)n+1}, & \text{if } \alpha \in NI(n,k) \\ (k+1)\alpha, & \text{if } \alpha \in I(n,k) \end{cases}$$

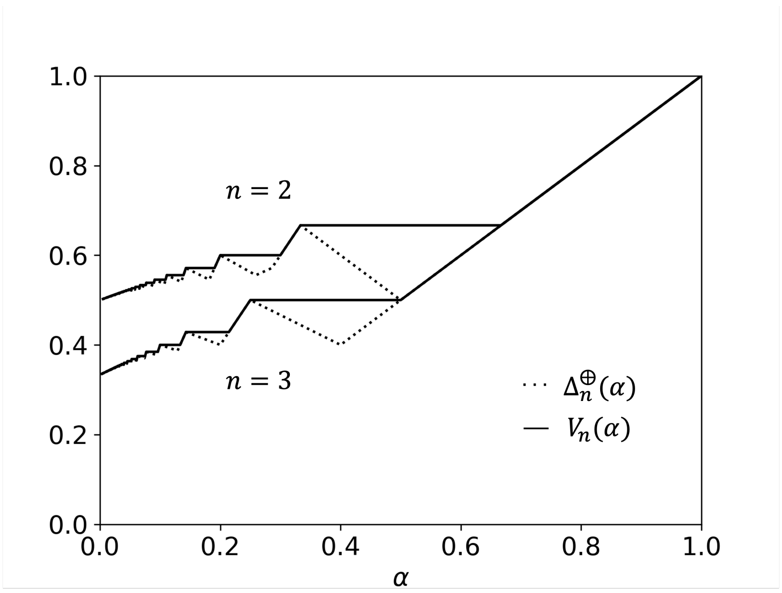


Figure 4.1: The characterisation for Heterogeneous Agents

where for any integer $k \geq 0$,

$$NI(n,k) = \left(\frac{1}{(k+1)n+1}, \frac{k+2}{(k+1)((k+1)n+1)} \right)$$

and

$$I(n,k) = \left[\frac{k+2}{(k+1)((k+1)n+1)}, \frac{1}{kn+1} \right].$$

By Theorem 6 and the construction of $V_n(\cdot)$, $V_n(\cdot)$ provides the tight bound of the worst-case MinMaxShare regarding $\mathcal{U}(\cdot)$. We further prove that $V_n(\cdot)$ is a guarantee and moreover an allocation satisfying $V_n(\cdot)$ can be found in polynomial time.

Theorem 7 $\Gamma_n(v) = V_n(\max_{e \in M} v(e))$ *defines a canonical guarantee. That is, given any* $0 \leq \alpha_i \leq 1$ *and* $v_i \in \mathcal{U}(\alpha_i)$ *for* $i = 1, \ldots, n$, *there exists an* *allocation* $(A_1, \ldots, A_n)$ *with*

$$v_i(A_i) \leq V_n(\alpha_i) \text{ for all } i = 1, \ldots, n$$

and such an allocation can be computed in polynomial time. Moreover, for any $0 \leq \alpha \leq 1$*, there exists* $\{v'_i\}_{i=1}^n$ *with* $v'_i \in \mathcal{U}(\alpha)$ *for any* $i \in [n]$ *such that* $V_n(\alpha)$ *is the best possible guarantee, i.e., for any allocation* $(B_1, \ldots, B_n)$*,*

$$\text{there exists } i \in N \text{ such that } v'_i(B_i) \geq V_n(\alpha).$$

As for $\Delta_n^{\oplus}(\cdot)$ in Theorem 1 the two key features of this guarantee are: its computation is elementary and it does not depend on the number of chores to allocate. As far as we know, no other similarly simple guarantee for the allocation of chores has been identified.

Remark 2 *By Theorem 7,* $V_n(\alpha)$ *is the best guarantee for disutilities in* $\mathcal{U}(\alpha)$*, and thus we get the tight counterpart result of [100] for chores. However, it may not be the best in the model of Gourvès et al. [93], i.e., for disutilities in* $\mathcal{V}(\alpha)$*. For example, when* $n = 2$*, we can show that* $\Delta_2^{\oplus}(\max_{e \in M} v_i(e))$ *is a tighter guarantee in the later model. Given two disutility functions* v_1 *and* v_2*, without loss of generality, suppose* $\Delta_2^{\oplus}(\max_e v_1(e)) \leq \Delta_2^{\oplus}(\max_e v_2(e))$*. Then we find the MinMax partition of* v_1 *so that the disutilities of both bundles are no greater than* $\Delta_2^{\oplus}(\max_e v_1(e))$ *to agent 1. We ask agent 2 to choose a better bundle whose disutility must be no greater than* $\frac{1}{2}$ *and thus no greater than* $\Delta_2^{\oplus}(\max_e v_2(e))$ *to agent 2. It is still open whether* $\Delta_n^{\oplus}(\max_{e \in M} v_i(e))$ *is a guarantee or not when* $n \geq 3$ *in Gourvès et al. [93]'s model, which is an* *interesting future research direction.*

4.2.1 Proof of Theorem 7

Next, we prove Theorem 7. We derive a variation of the moving-knife algorithm to compute an allocation satisfying the required bound in Theorem 7. When the items are goods and divisible, Dubins and Spanier [68] proved that such an algorithm (also known as Dubins-Spanier moving knife algorithm) gives the optimal worst-case bound, i.e., every agent gets value for at least $\frac{1}{n}$. Markakis and Psomas [117] further proved that a variation of this algorithm also guarantees the optimal worst-case bound for indivisible goods. In a nutshell, towards proving Theorem 7, we first use the reduction proved in [46, 104] to restrict our attention to the ordered instances when agents have the same ranking over all chores, which significantly simplifies our analysis. Then we show that using $V_n(\cdot)$ to set the parameters in the moving-knife algorithm always returns an allocation ensuring the bound in Theorem 7.

The following lemma says that it suffices to only focus on the ordered instances.

Lemma 4 *Suppose there is an algorithm that takes any ordered instance as input, runs in $T(n,m)$ time and returns an allocation where each agent i's disutility is at most $V_n(\alpha_i)$. Then, we have an algorithm that takes any instance as input, runs in $T(n,m)+O(nm\log m)$ time and returns an allocation with the same disutility guarantee.*

Our approach is similar to that in [117], but the detailed proof differs. Our algorithm runs in recursions. In each recursion, the algorithm allocates a bundle of chores to one agent in a moving-knife fashion. Each time, each of the remaining agents moves her "knife" one chore towards the chores with smaller disutilities, until for every agent i the total disutility of the chores before her "knife" is larger than $V_n(\alpha_i)$. After that, one of the last agents (denoted by agent k) for whom the total disutility of the chores before her knife is larger than $V_n(\alpha_k)$ receives the chores except the one right before her knife. If there remains only one agent who has not received a bundle, she will get all remaining chores. Otherwise, all remaining agents enter the next recursion with their disutility functions being normalised such that for each of them the total disutility of the remaining chores is 1. The formal description of our algorithm is presented in Algorithm 8.

Then we are going to prove Theorem 7. Without loss of generality, let $1, \ldots, n$ be the order in which agents receive bundles in Algorithm 8. Denote $C_i = v_i(A_1)$ for every $N \setminus \{1\}$, the following lemma gives a lower bound of C_i.

Lemma 5 *For any agent $i \in N \setminus \{1\}$ with $\alpha_i \in NI(n,k) \cup I(n,k)$ for some $k \geq 0$, we have*

$$C_i \geq \frac{1 - V_n(\alpha_i)}{n-1}.$$

Proof. Denote by q the index such that $\sum_{e=1}^{q} v_i(e) \leq V_n(\alpha_i)$ and $\sum_{e=1}^{q+1} v_i(e) > V_n(\alpha_i)$, whose existence is guaranteed since $v_i(M) > V_n(\alpha_i)$. Since $V_n(\alpha_i) \geq (k+1)\alpha_i$ (this can be easily verified from the definition of $V_n(\alpha)$ and can also be seen from Fig. 4.1) and $v_i(e) \leq \alpha_i$ for every $e \in M$, $q \geq k+1$. Otherwise, $\sum_{e=1}^{q+1} v_i(e) \leq (k+1)\alpha_i \leq V_n(\alpha_i)$, which contradicts the definition of q. According to Algorithm 8, $C_i \geq \sum_{e=1}^{q} v_i(e)$. Since only ordered instances are

Algorithm 8 Algorithm for heterogeneous disutilities

Require: An ordered instance with agents N, chores M and disutility functions $\{v_i\}_{i \in N}$.

Ensure: An allocation $\mathbf{A} = \{A_1, \ldots, A_n\}$ with $v_i(A_i) \leq V_n(\alpha_i)$ for every $i \in N$.

Initialize $S_i = \emptyset$ for every $i \in N$.
while there exists an agent j with $v_j(S_j) \leq V_n(\alpha_j)$ **do**
    for every $i \in N$ **do**
        $S_i \leftarrow S_i \cup$ {the chore in $M \setminus S_i$ with the largest disutility for agent i (tie breaks arbitrarily)}.
    end for
end while
Pick the agent $k \in N$ with $v_k(S_k \setminus \{\widetilde{e}\}) \leq V_n(\alpha_k)$ where $\widetilde{e}$ is the last chore that k added into S_k (tie breaks arbitrarily).
$A_k \leftarrow S_k \setminus \{\widetilde{e}\}$.
if $|N| = 2$ **then**
    Allocate $M \setminus A_k$ to the remaining agent.
else
    Construct a new disutility function v'_i for every $i \in N \setminus \{k\}$ by setting $v'_i(e) = \frac{v_i(e)}{1 - v_i(A_k)}$ for every $e \in M \setminus A_k$.
    Run Algorithm 8($N \setminus \{k\}$, $M \setminus A_k$, $\{v'_i\}_{i \in N \setminus \{k\}}$).
end if

considered, $v_i(\{q+1\}) \le v_i(\{q\}) \le \frac{C_i}{k+1}$, which gives

$$C_i + \frac{C_i}{k+1} \ge \sum_{e=1}^{q+1} v_i(e) > V_n(\alpha_i).$$

Therefore, $C_i > \frac{k+1}{k+2} \cdot V_n(\alpha_i)$. We consider the following two cases regarding the ranges of α_i.

Case 1: $\alpha_i \in I(n,k)$. In this case, $\frac{k+2}{(k+1)((k+1)n+1)} \le \alpha_i \le \frac{1}{kn+1}$ and $V_n(\alpha_i) = (k+1)\alpha_i$. Then,

$$C_i > \frac{k+1}{k+2} \cdot V_n(\alpha) \ge \frac{1 - V_n(\alpha)}{n-1},$$

where the last inequality holds since $\alpha_i \ge \frac{k+2}{(k+1)((k+1)n+1)}$.

Case 2: $\alpha_i \in NI(n,k)$. In this case, $V_n(\alpha_i) = \frac{k+2}{(k+1)n+1}$, which gives

$$C_i > \frac{k+1}{k+2} \cdot V_n(\alpha) = \frac{1 - V_n(\alpha)}{n-1},$$

which completes the proof. ∎

Interestingly, the following lemma shows the connection between the ranges of α_i and $\frac{\alpha_i}{1 - \frac{1-V_n(\alpha_i)}{n-1}}$.

Lemma 6 *For any $\alpha_i \in NI(n,k) \cup I(n,k)$ for some $k \ge 0$, we have*

$$\frac{\alpha_i}{1 - \frac{1-V_n(\alpha_i)}{n-1}} \in \begin{cases} I(n-1,k), & \text{if } \alpha_i \in I(n,k) \\ NI(n-1,k), & \text{if } \alpha_i \in NI(n,k) \end{cases}$$

Proof. We consider the following two cases regarding the ranges of α_i.

Case 1: $\alpha_i \in I(n,k)$. In this case, $\frac{k+2}{(k+1)((k+1)n+1)} \le \alpha_i \le \frac{1}{kn+1}$ and $V_n(\alpha_i) = (k+1)\alpha_i$. Then, we have

$$\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}} = \frac{(n-1)\alpha_i}{n-2+(k+1)\alpha_i} \le \frac{n-1}{(n-2)(kn+1)+k+1}$$
$$= \frac{1}{k(n-1)+1},$$

where the inequality is because $\alpha_i \le \frac{1}{kn+1}$. Besides,

$$\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}} = \frac{(n-1)\alpha_i}{n-2+(k+1)\alpha_i} \ge \frac{(k+2)(n-1)}{(k+1)((n-2)(kn+n+1)+k+2)}$$
$$= \frac{k+2}{(k+1)((k+1)(n-1)+1)},$$

where the inequality is because $\alpha_i \ge \frac{k+2}{(k+1)((k+1)n+1)}$.

Case 2: $\alpha_i \in NI(n,k)$. In this case, $\frac{1}{(k+1)n+1} < \alpha_i < \frac{k+2}{(k+1)((k+1)n+1)}$ and $V_n(\alpha_i) = \frac{k+2}{(k+1)n+1}$. Then, we have

$$\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}} = \frac{((k+1)n+1)\alpha_i}{(k+1)(n-1)+1} < \frac{k+2}{(k+1)((k+1)(n-1)+1)},$$

where the inequality is because $\alpha_i < \frac{k+2}{(k+1)((k+1)n+1)}$. Besides,

$$\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}} = \frac{((k+1)n+1)\alpha_i}{(k+1)(n-1)+1} > \frac{1}{(k+1)(n-1)+1},$$

where the inequality is because $\alpha_i > \frac{1}{(k+1)n+1}$. ■

Proof of Theorem 7. We prove Theorem 7 by mathematical induction. When $n = 2$, it is easy to see the correctness of Theorem 7 from Lemma 5 since $v_1(A_1) \leq V_2(\alpha_1)$ and $v_2(A_2) = 1 - v_2(A_1) \leq 1 - (1 - V_2(\alpha_2)) = V_2(\alpha_2)$, We assume as our induction hypothesis that Theorem 7 holds for $n-1$. Then we aim to prove the correctness for n.

From Algorithm 8, $v_1(A_1) \leq V_n(\alpha_1)$ clearly holds for agent 1. For any other agent $i \in N \setminus \{1\}$, denote $\widetilde{\alpha}_i = \max_{e \in M \setminus A_1} v'_i(e)$. We know from Algorithm 8 that $\widetilde{\alpha}_i \leq \frac{\alpha_i}{1-C_i}$ and from the induction hypothesis that $v'_i(A_i) \leq V_{n-1}(\widetilde{\alpha}_i)$, which together give

$$v_i(A_i) = (1 - C_i)v'_i(A_i) \leq (1 - C_i)V_{n-1}(\widetilde{\alpha}_i) \leq (1 - C_i)V_{n-1}(\frac{\alpha_i}{1 - C_i}),$$

where the last inequality holds by recalling that $V_{n-1}(\widetilde{\alpha}_i)$ is an non-decreasing function of $\widetilde{\alpha}_i$. Therefore, it remains to show

$$(1 - C_i)V_{n-1}(\frac{\alpha_i}{1 - C_i}) \leq V_n(\alpha_i).$$

Note that $(1 - C_i)V_{n-1}(\frac{\alpha_i}{1-C_i})$ is an non-increasing function of C_i. This is because when $\frac{\alpha_i}{1-C_i} \in I(n-1, k)$ for some k, $(1 - C_i)V_{n-1}(\frac{\alpha_i}{1-C_i}) = (1 - C_i)(k + 1)\frac{\alpha_i}{1-C_i} = (k + 1)\alpha_i$, which is a constant with respect to C_i; when $\frac{\alpha_i}{1-C_i} \in NI(n-1, k)$ for some k, $(1 - C_i)V_{n-1}(\frac{\alpha_i}{1-C_i}) = (1 - C_i)\frac{k+2}{(k+1)(n-1)+1}$, a decreasing function of C_i. Then, the following formula completes the proof of Theorem 7

$$(1 - C_i)V_{n-1}(\frac{\alpha_i}{1 - C_i}) \leq (1 - \frac{1 - V_n(\alpha_i)}{n - 1})V_{n-1}(\frac{\alpha_i}{1 - \frac{1-V_n(\alpha_i)}{n-1}}),$$

where the first inequality is due to $C_i \geq \frac{1-V_n(\alpha_i)}{n-1}$ (according to Lemma 5), and the second inequality can be verified by considering the following two cases regarding the ranges of α_i,

Case 1: $\alpha_i \in I(n,k)$. In this case, Lemma 6 gives $\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}} \in I(n-1,k)$. Thus, we have

$$(1-\frac{1-V_n(\alpha_i)}{n-1})V_{n-1}(\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}}) = (1-\frac{1-V_n(\alpha_i)}{n-1})\cdot(k+1)\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}}$$
$$= (k+1)\alpha_i = V_n(\alpha_i).$$

Case 2: $\alpha_i \in NI(n,k)$. In this case, $\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}} \in NI(n-1,k)$. Thus, we have

$$(1-\frac{1-V_n(\alpha_i)}{n-1})V_{n-1}(\frac{\alpha_i}{1-\frac{1-V_n(\alpha_i)}{n-1}}) = (1-\frac{1-\frac{k+2}{(k+1)n+1}}{n-1})\cdot\frac{k+2}{(k+1)(n-1)+1}$$
$$= \frac{k+2}{(k+1)n+1} = V_n(\alpha_i).$$

Therefore, we complete the proof of Theorem 7. ■

The instances provided in Section 4.1 show the tightness of Theorem 7.

4.3 Numerical Experiments

To demonstrate that $\Delta_n^{\oplus}(\alpha;m)$ can serve as a good alternative of MinMaxShare, we first evaluate the worst-case ratio of $\Delta_n^{\oplus}(\alpha;m)$ and $\Delta_n^{\ominus}(\alpha;m)$ (recall that $\Delta_n^{\ominus}(\alpha;m)$ is the best-case MinMaxShare over all disutilities in $\mathcal{V}(\alpha;m)$). Denote by $r_n(\alpha;m) = \frac{\Delta_n^{\oplus}(\alpha;m)}{\Delta_n^{\ominus}(\alpha;m)}$. It is clear that $r_n(\alpha;m)$ is no smaller than the ratio between $\Delta_n^{\oplus}(\alpha;m)$ and the real MinMaxShare, and we have illustrated $r_n(\alpha;\infty)$ in Fig. 1.2 for $n = 2, 10, 100$. As we can see, although the worst-case ratio can be close to 2, it only happens for sufficiently large n and a small range of values of α. For any n and most values of α, the ratio is better than $\frac{4}{3}$ and $\frac{11}{9}$, which are two fractions of the MinMaxShare that are known to be achievable. Actually, it is not hard to verify that $r_n(\alpha; m) \le \frac{2n}{n+1} < 2$ for all α, and $r_n(\alpha; m) \le \frac{4}{3}$ for all α outside of $(\frac{4}{9n}, \frac{3}{2n+3})$. Note that $\frac{3}{2n+3} - \frac{4}{9n} < \frac{7}{6n}$.

Claim 5 *For any $n \ge 2$, $\alpha \in (0, 1]$ and $m \ge \lceil \frac{1}{\alpha} \rceil$, $r_n(\alpha; m) \le \frac{2n}{n+1}$.*

Proof. Notice that by Lemma 1 and Lemma 2, $r_n(\alpha; m)$ is weakly increasing in m. Therefore, it suffices to prove the claim for the setting when m is unrestricted, i.e., $r_n(\alpha) \le \frac{2n}{n+1}$. We first consider the case where $n = 2$ and $k = 1$. In this case, $\alpha \in (\frac{1}{5}, \frac{1}{3}]$. Since $\alpha < \frac{1}{n} = \frac{1}{2}$, $\Delta_2^{\ominus}(\alpha) = \frac{1}{2}$. When $\alpha \in (\frac{1}{5}, \frac{7}{27}]$, $\Delta_2^{\oplus}(\alpha) = \frac{3-3\alpha}{4}$, thus $r_2(\alpha) = \frac{3-3\alpha}{2} < \frac{6}{5} < \frac{4}{3}$; when $\alpha \in (\frac{7}{27}, \frac{2}{7}]$, $\Delta_2^{\oplus}(\alpha) = \frac{2+3\alpha}{5}$, thus $r_2(\alpha) = \frac{4+6\alpha}{5} \le \frac{8}{7} < \frac{4}{3}$; when $\alpha \in (\frac{2}{7}, \frac{1}{3}]$, $\Delta_2^{\oplus}(\alpha) = 2\alpha$, thus $r_2(\alpha) = 4\alpha \le \frac{4}{3}$.

We next consider the cases when $n \ge 3$ or $k \ne 1$. When $\alpha > \frac{1}{n}$ which means $\alpha \in I(n, 0)$ or $\alpha \in (\frac{1}{n}, \frac{2}{n+2}] \in D(n, 0)$, $\Delta_n^{\ominus}(\alpha) = \alpha$. Thus, when $\alpha \in I(n, 0)$, $\Delta_n^{\oplus}(\alpha) = \alpha$ and $r_n(\alpha) = 1 < \frac{4}{3} \le \frac{2n}{n+1}$ since $n \ge 2$; when $\alpha \in (\frac{1}{n}, \frac{2}{n+2}]$, $\Delta_n^{\oplus}(\alpha) = \frac{2 \cdot (1-\alpha)}{n}$ and $r_n(\alpha) = \frac{2}{n} \cdot \frac{1-\alpha}{\alpha} < 2 \cdot (1 - \frac{1}{n}) < \frac{2n}{n+1}$. When $\alpha \le \frac{1}{n}$, it follows that $\alpha \in (\frac{1}{n+1}, \frac{1}{n}] \in D(n, 0)$ or $\alpha \in I(n, k)$ with $k \ge 1$ or $\alpha \in D(n, k)$ with $k \ge 1$. In these cases, $\Delta_n^{\ominus}(\alpha) = \frac{1}{n}$. When $\alpha \in (\frac{1}{n+1}, \frac{1}{n}]$, $\Delta_n^{\oplus}(\alpha) = \frac{2 \cdot (1-\alpha)}{n}$ and $r_n(\alpha) = 2 \cdot (1 - \alpha) < \frac{2n}{n+1}$; when $\alpha \in I(n, k) = (\frac{k+2}{n(k+1)^2+k+2}, \frac{1}{kn+1}]$ with $k \ge 1$, $\Delta_n^{\oplus}(\alpha) = (k + 1)\alpha$ and $r_n(\alpha) = n(k + 1) \cdot \alpha \le \frac{kn+n}{kn+1} \le \frac{2n}{n+1}$; when $\alpha \in D(n, k) = (\frac{1}{kn+n+1}, \frac{k+2}{n(k+1)^2+k+2}]$ with $k \ge 1$, $\Delta_n^{\oplus}(\alpha) = \frac{k+2}{k+1} \cdot \frac{1-\alpha}{n}$ and $r_n(\alpha) = \frac{k+2}{k+1} \cdot (1 - \alpha) < \frac{kn+2n}{kn+n+1} \le \frac{3n}{2n+1} < \frac{2n}{n+1}$. ∎

Claim 6 *$r_n(\alpha; m) > \frac{4}{3}$ only when $\alpha \in (\frac{2}{9}, \frac{1}{3})$ if $n = 3$, or $\alpha \in (\frac{1}{6}, \frac{3}{11})$ if $n = 4$, or $\alpha \in (\frac{4}{45}, \frac{1}{9}) \cup (\frac{2}{15}, \frac{3}{13})$ if $n = 5$, or $\alpha \in (\frac{4}{9n}, \frac{3}{2n+3})$ if $n \geq 6$.*

Proof. Note that we actually derive the ranges of α that satisfy $r_n(\alpha; +\infty) > \frac{4}{3}$, which are necessary conditions for $r_n(\alpha; m) > \frac{4}{3}$ but may not be sufficient ones. We use the formulas of $r_n(\alpha)$ derived in the proof of Claim 5, and only consider the following cases when $r_n(\alpha)$ may be larger than $\frac{4}{3}$.

- When $\alpha \in (\frac{1}{n}, \frac{2}{n+2}]$, $r_n(\alpha) = \frac{2}{n} \cdot \frac{1-\alpha}{\alpha}$, which is larger than $\frac{4}{3}$ when $\alpha < \frac{3}{2n+3}$. Since $\frac{3}{2n+3} > \frac{1}{n}$ only when $n \geq 4$, the range is $\alpha \in (\frac{1}{n}, \frac{3}{2n+3})$ with $n \geq 4$.

- When $\alpha \in (\frac{1}{n+1}, \frac{1}{n}]$, $r_n(\alpha) = 2 \cdot (1 - \alpha)$, which is larger than $\frac{4}{3}$ when $\alpha < \frac{1}{3}$. Since $\frac{1}{n+1} < \frac{1}{3}$ only when $n \geq 3$ and $\frac{1}{n} \leq \frac{1}{3}$ when $n \geq \frac{1}{3}$, the range is $\alpha \in (\frac{1}{n+1}, \frac{1}{n})$ with $n \geq 3$.

- When $\alpha \in I(n,k) = (\frac{k+2}{n(k+1)^2+k+2}, \frac{1}{kn+1}]$ with $k \geq 1$, $r_n(\alpha) = n(k+1) \cdot \alpha$, which is larger than $\frac{4}{3}$ when $\alpha > \frac{4}{3n(k+1)}$. Note that $\frac{4}{3n(k+1)} < \frac{1}{kn+1}$ is equivalent to $(3-k)n > 4$, which can be satisfied only when $k = 1$ or $k = 2$. When $k = 1$, $(3-k)n > 4$ gives $n \geq 3$, $\alpha > \frac{4}{3n(k+1)}$ is equivalent to $\alpha > \frac{2}{3n}$, and $\frac{k+2}{n(k+2)^2+k+2} = \frac{3}{4n+3}$. Since $\frac{3}{4n+3} \geq \frac{2}{3n}$ when $n \geq 6$, the ranges are $\alpha \in (\frac{2}{3n}, \frac{1}{n+1})$ with $3 \leq n \leq 5$, and $\alpha \in (\frac{3}{4n+3}, \frac{1}{n+1})$ with $n \geq 6$. When $k = 2$, $(3-k)n > 4$ gives $n \geq 5$, $\alpha > \frac{4}{3n(k+1)}$ is equivalent to $\alpha > \frac{4}{9n}$, and $\frac{k+2}{n(k+2)^2+k+2} = \frac{1}{4n+1}$. Since $\frac{4}{9n} > \frac{1}{4n+1}$, the range is $\alpha \in (\frac{4}{9n}, \frac{1}{2n+1})$ with $n \geq 5$.

- When $\alpha \in D(n,k) = (\frac{1}{kn+n+1}, \frac{k+2}{n(k+1)^2+k+2}]$ with $k \geq 1$, $r_n(\alpha) = \frac{k+2}{k+1} \cdot (1-\alpha)$, which is larger than $\frac{4}{3}$ when $\alpha < \frac{2-k}{3k+6}$. Note that $\frac{2-k}{3k+6} > 0$ only when $k = 1$. Then, $\alpha \leq \frac{2-k}{3k+6}$ is equivalent to $\alpha < \frac{1}{9}$, $\frac{1}{kn+n+1} = \frac{1}{2n+1}$ and $\frac{k+2}{n(k+2)^2+k+2} = \frac{3}{4n+3}$. Since $\frac{3}{4n+3} \leq \frac{1}{9}$ when $n \geq 6$ and $\frac{1}{9} > \frac{1}{2n+1}$ when $n \geq 5$, the ranges are $(\frac{1}{2n+1}, \frac{3}{4n+3})$ with $n \geq 6$, and $(\frac{1}{2n+1}, \frac{1}{9})$ with $n = 5$.

By summarising the above ranges, we complete the proof. ■

From the formula of $r_n(\alpha; m)$, as well as Fig. 1.2, we have the following observations:

Observation 1 As n increases, the worst-case ratio of $r_n(\alpha; m)$, i.e., $\max_\alpha r_n(\alpha; m)$, increases.

Observation 2 As n increases, large values of $r_n(\alpha; m)$ happen increasingly more rarely if α is randomly generated from $[0, 1]$.

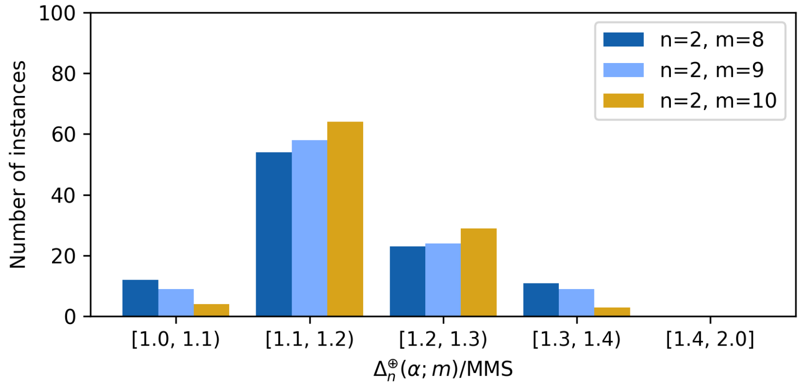
Next, we conduct numerical experiments with synthetic and real-world data to illustrate the real distances between $\Delta_n^{\oplus}(\alpha; m)$ and the MinMaxShare of specific disutility functions, which also validate the above two observations.

4.3.1 Experiments with Synthetic Data

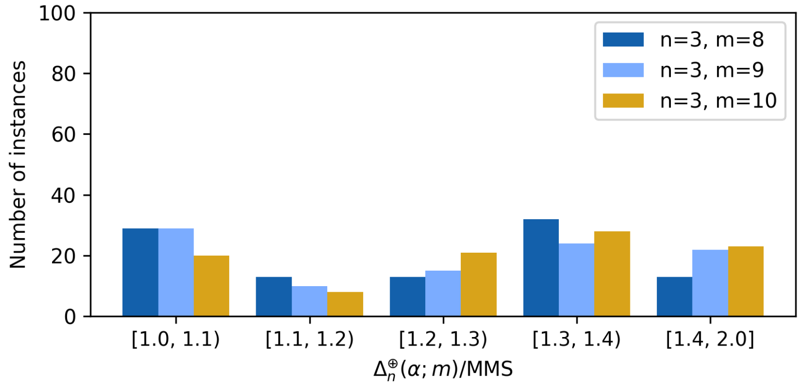
We randomly generate a number of disutility functions, and for each of them, we compute the ratio between the corresponding Hill's share and the MinMaxShare. In particular, for each given n and m, we randomly generate 100 instances; for each instance, we randomly generate $m - 1$ numbers in $[0, 1]$. These $m-1$ numbers separate the interval $[0, 1]$ into m contiguous segments, and the lengths of these segments are used as the disutilities of the m chores. Then we compute the $\Delta_n^{\oplus}(\alpha; m)$ value using the maximum of these values and the MinMaxShare. For each instance, we record the ratio of these two quantities.

The results are summarized in Fig. 4.2 and Fig. 4.3. We slice the ratios into small intervals, each of which has a length of 0.1, and count the number of instances falling into each interval for each setting. The figures validate the previous two observations: when $n = 2$ and 3, the largest ratio can only reach interval $[1.3, 1.4)$ and $[1.4, 1.5)$, but when $n \geq 4$, it reaches $[1.5, 1.6)$; however, looking at the number of instances, for larger n, fewer and fewer instances fall into these large intervals, and instead, the number of instances in $[1.0, 1.1)$ significantly dominates the other intervals. Specifically, when $n = 6$ and 7, $[1.0, 1.1)$ contains over 80% of all random instances, and none of them reaches a ratio beyond 1.6, while the worst-case ratio can be greater than 1.7.

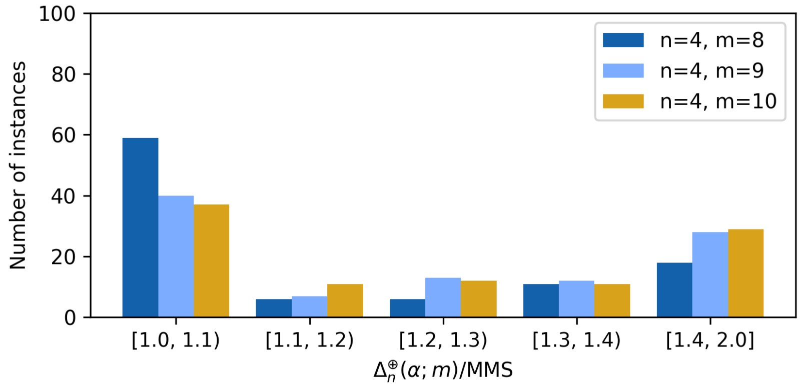
We also conduct more experiments by fixing $n = 2$ and increasing the value of m and report the change in the distribution of the ratios. We observe that in Fig. 4.2, when $n = 2$, the majority of random instances fall into the interval of $[1.1, 1.2)$, in contrast to the other values of n that are concentrated within $[1.0, 1.1)$. This is in part because the ratio of m over n is larger than $n > 2$, given each m. One may be curious that when m becomes larger and larger to n, the majority may be close to the worst-case ratio. Due to this curiosity, we further conduct the following experiment by setting $m = 15 \pm 1$ and $m = 20 \pm 1$, where n is fixed at 2. The results are shown in Fig. 4.4. As we can see, the instances get more concentrated within $[1.1, 1.2)$, and the number of instances whose ratios are above 1.2 get less and less.



(a) $n = 2$, $m = 8, 9, 10$

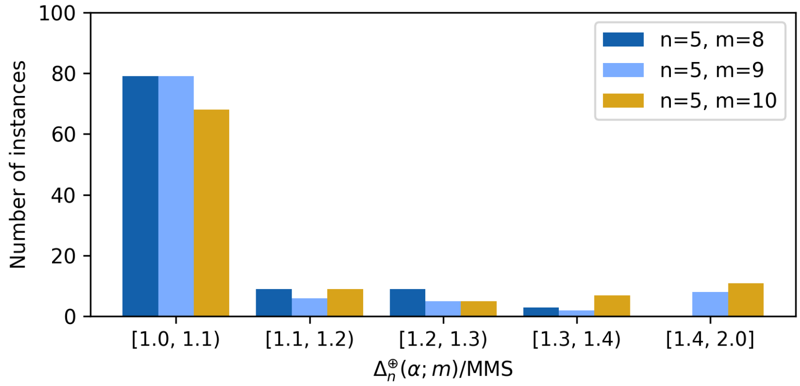


(b) $n = 3$, $m = 8, 9, 10$

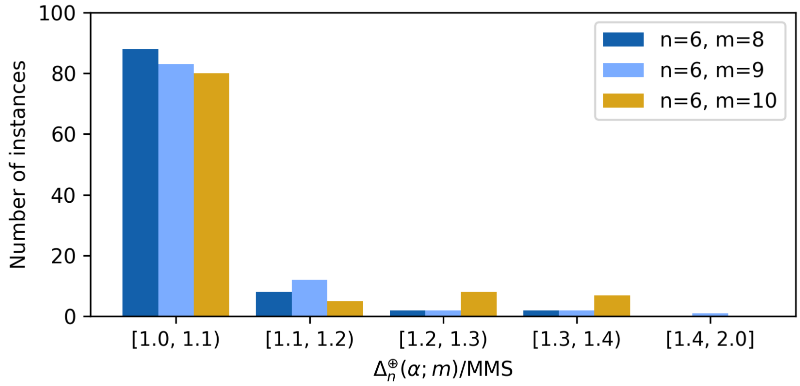


(c) $n = 4$, $m = 8, 9, 10$

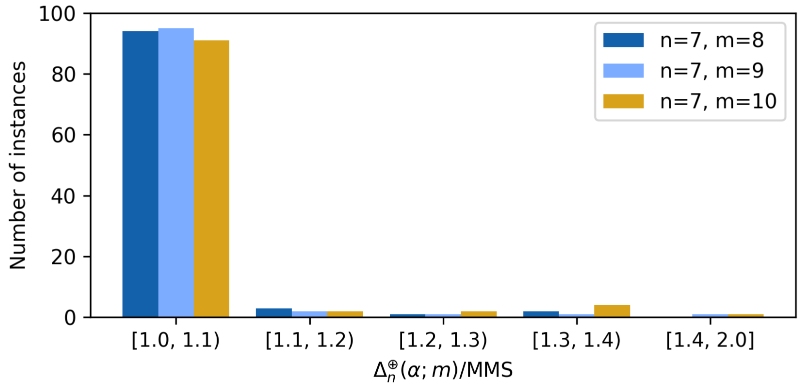
Figure 4.2: Ratios in random data when $n = 2, 3, 4$, $m = 8, 9, 10$



(a) $n = 5$, $m = 8, 9, 10$

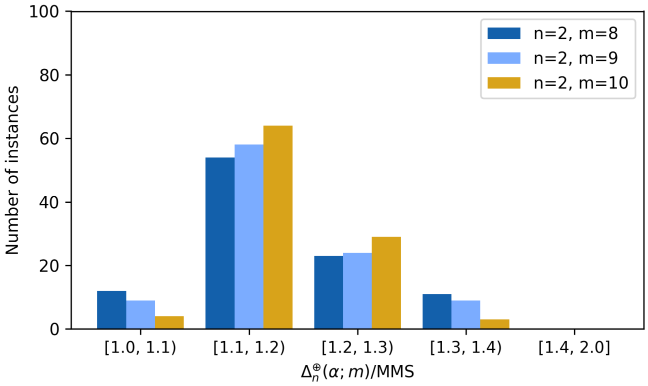


(b) $n = 6$, $m = 8, 9, 10$

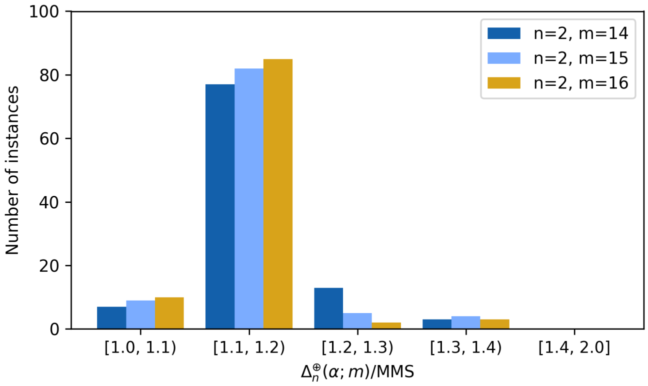


(c) $n = 7$, $m = 8, 9, 10$

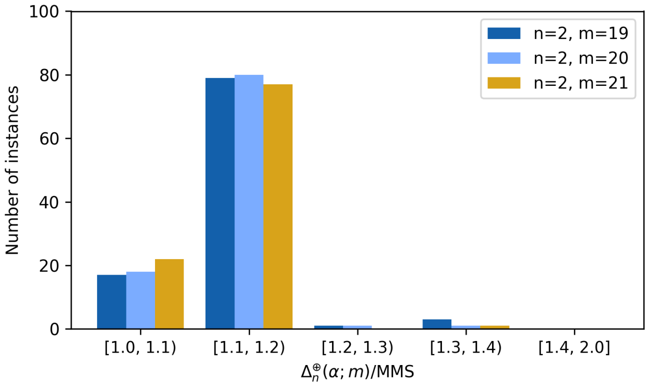
Figure 4.3: Ratios in random data when $n = 5, 6, 7$, $m = 8, 9, 10$



(a) $n = 2$, $m = 8, 9, 10$



(b) $n = 2$, $m = 14, 15, 16$



(c) $n = 2$, $m = 19, 20, 21$

Figure 4.4: Fixing $n = 2$ and increasing the value of m.

4.3.2 Experiments with Real-World Data

The real-world data set is collected from the Spliddit platform (spliddit.org) – a well-known platform that provides implementations of fair allocation algorithms for various practical problems [90]. The data set contains 8,409 instances created between October 2014 and May 2020, involving 22,530 agents and 42,469 chores. We randomly select 10,000 disutility functions from the data, where the largest value of n is 14. After normalising all disutility functions, for each of them, we record the ratio of the corresponding Hill's share and the MinMaxShare. The results are shown in Fig. 4.5. As we can see, very few instances have ratios higher than 1.4, and over 65% of the instances have ratios within $[1.0, 1.1)$. Actually, there are only 173 (= 1.73%) and 26 (= 0.26%) instances falling into $[1.6, 1.7)$ and $[1.7, 1.8)$ respectively, and none is beyond 1.8. Note that in the 10,000 disutility functions, there are only 14 instances with $n \geq 9$, which further amplifies the rare happening of large ratios.

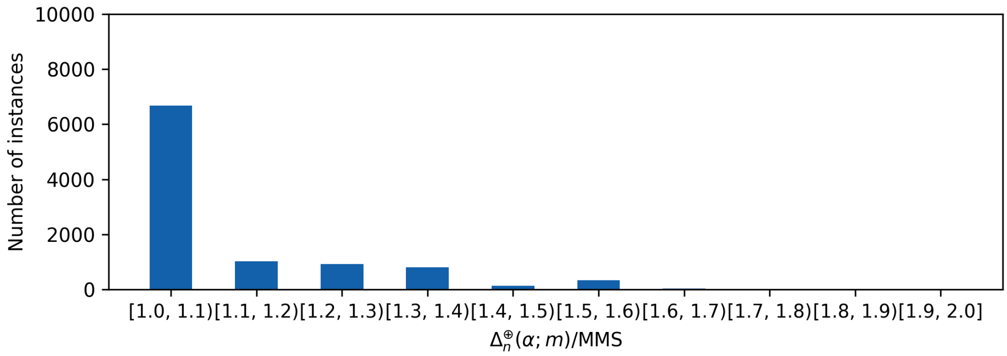


Figure 4.5: Ratios in Spliddit data.

CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we first study the fair allocation of indivisible chores when the disutilities are subadditive and the fairness is measured by MMS. There are many open problems and further directions. First, for general subadditive and bin packing disutility functions, we provide the tight approximation ratios, but for the job scheduling model, we only have a lower bound of 44/43 which is inherited from additive disutilities. One immediate direction is to design better approximation algorithms or lower-bound instances for the job scheduling disutilities. Second, we show that for the bin packing model, there exists an allocation where everyone's disutility is no more than $\frac{3}{2}$ times her MMS plus 1. We suspect that the multiplicative factor can be improved to 1. Third, for job scheduling disutilities, we restricted us on the case of related machines, it is worth interest to consider the general model of unrelated machines. As we mentioned, the notion of collective maximin share fairness in the job scheduling model can be viewed as a group-wise fairness notion (for both goods and chores), which could be studied of independent interest. Finally, we can investigate other combinatorial disutilities that can better characterise real-world problems.

We next give the tight characterisation of Hill's share for allocating indivisible chores, i.e., the exact upper bound of the MinMaxShare of disutility functions with the same largest single-chore disutility. Hill's share exhibits several advantages including elementary computation, being close to the MinMaxShare, and displaying the effect of an agent's disutility on her share of all chores. More importantly, the monotonic cover of Hill's share serves as a canonical guarantee; as far as we know, no other similarly simple guarantee for the allocation of chores has been identified. There are some open problems. Hill's guarantee is tight for the domain of disutility functions whose largest single-chore disutility is *no greater than* a given parameter, but we do not know whether it is tight when the domain only contains the disutility functions whose largest single-chore disutility *equals* this parameter. The same problem is also open for the mirror problem of allocating goods, for which the tight characterisation of Hill's share is also unknown (when $n \geq 3$). Our work also uncovers some other related research problems, such as the algorithmic problem of finding a Pareto optimal allocation satisfying Hill's share and the game-theoretic problem of designing truthful mechanisms that incentivize the agents to report their disutility functions honestly while achieving (approximations of) Hill's share.

We can also study our problems under the online setting where the chores or the agents or both are coming in an online fashion, and allocation decisions must be made immediately when they come. We would like to know the extent to which fairness can be guaranteed when the decisions cannot be revoked, and the number of swaps that are needed to guarantee better fairness when swaps between agents are allowed.

Bibliography

[1] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online fair division: analysing a food bank problem. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 2540–2546.

[2] Martin Aleksandrov and Toby Walsh. 2017. Expected Outcomes and Manipulations in Online Fair Division. In *KI (Lecture Notes in Computer Science, Vol. 10505)*. Springer, 29–43.

[3] Martin Aleksandrov and Toby Walsh. 2017. Most Competitive Mechanisms in Online Fair Division. In *KI (Lecture Notes in Computer Science, Vol. 10505)*. Springer, 44–57.

[4] Martin Aleksandrov and Toby Walsh. 2017. Pure Nash Equilibria in Online Fair Division. In *IJCAI*. ijcai.org, 42–48.

[5] Martin Aleksandrov and Toby Walsh. 2019. Strategy-Proofness, Envy-Freeness and Pareto Efficiency in Online Fair Division with Additive Utilities. In *PRICAI (1) (Lecture Notes in Computer Science, Vol. 11670)*. Springer, 527–541.

[6] Martin Aleksandrov and Toby Walsh. 2020. Online Fair Division: A Survey. In *AAAI*. AAAI Press, 13557–13562.

[7] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A. Voudouris. 2020. Maximum Nash Welfare and Other Stories About EFX. In *IJCAI*. ijcai.org, 24–30.

[8] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. 2016. On Truthful Mechanisms for Maximin Share Allocations. In *IJCAI*. IJCAI/AAAI Press, 31–37.

[9] Georgios Amanatidis, Georgios Birmpas, and Vangelis Markakis. 2018. Comparing Approximate Relaxations of Envy-Freeness. In *IJCAI*. ijcai.org, 42–48.

[10] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. 2017. Approximation Algorithms for Computing Maximin Share Allocations. *ACM Trans. Algorithms* 13, 4 (2017), 52:1–52:28.

[11] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. 2020. Multiple Birds with One Stone: Beating 1/2 for EFX and GMMS via Envy Cycle Elimination. In *AAAI*. AAAI Press, 1790–1797.

[12] Eshwar Ram Arunachaleswaran, Siddharth Barman, Rachitesh Kumar, and Nidhi Rathi. 2019. Fair and efficient cake division with connected pieces. In *Web and Internet Economics: 15th International Conference, WINE 2019, New York, NY, USA, December 10–12, 2019, Proceedings*. Springer, 57–70.

[13] Orit Arzi, Yonatan Aumann, and Yair Dombb. 2011. Throw one's cake—and eat it too. In *Algorithmic Game Theory: 4th International Symposium, SAGT 2011, Amalfi, Italy, October 17-19, 2011. Proceedings 4*. Springer, 69–80.

[14] Yonatan Aumann and Yair Dombb. 2010. The efficiency of fair division with connected pieces. In *Internet and Network Economics: 6th International Workshop, WINE 2010, Stanford, CA, USA, December 13-17, 2010. Proceedings 6*. Springer, 26–37.

[15] Yonatan Aumann, Yair Dombb, and Avinatan Hassidim. 2012. Computing socially-efficient cake divisions. *arXiv preprint arXiv:1205.3982* (2012).

[16] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. 2019. Fair Allocation of Indivisible Goods and Chores. In *IJCAI*. ijcai.org, 53–59.

[17] Haris Aziz, Hau Chan, and Bo Li. 2019. Weighted Maxmin Fair Share Allocation of Indivisible Chores. In *IJCAI*. ijcai.org, 46–52.

[18] Haris Aziz, Bo Li, and Xiaowei Wu. 2022. Approximate and strategyproof maximin share allocation of chores with ordinal preferences. *Mathematical Programming* (2022), 1–27.

[19] Haris Aziz and Simon Mackenzie. 2016. A Discrete and Bounded Envy-Free Cake Cutting Protocol for Any Number of Agents. In *FOCS*. IEEE Computer Society, 416–427.

[20] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. 2020. A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation. *Oper. Res. Lett.* 48, 5 (2020), 573–578.

[21] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. 2017. Algorithms for Max-Min Share Fair Allocation of Indivisible Chores. In *AAAI*. AAAI Press, 335–341.

[22] Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2021. Fair-Share Allocations for Agents with Arbitrary Entitlements. In *EC*. ACM, 127.

[23] Moshe Babaioff, Noam Nisan, and Inbal Talgam-Cohen. 2019. Fair Allocation through Competitive Equilibrium from Generic Incomes. In *FAT*. ACM, 180.

[24] Moshe Babaioff, Noam Nisan, and Inbal Talgam-Cohen. 2021. Competitive Equilibrium with Indivisible Goods and Generic Budgets. *Math. Oper. Res.* 46, 1 (2021), 382–403.

[25] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. 2021. Achieving Proportionality up to the Maximin Item with Indivisible Goods. In *AAAI*. AAAI Press, 5143–5150.

[26] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. 2021. PROPm Allocations of Indivisible Goods to Multiple Agents. In *IJCAI*. ijcai.org, 24–30.

[27] Julius Barbanel. 1996. Game-theoretic algorithms for fair and strongly

fair cake division with entitlements. In *Colloquium Mathematicae*, Vol. 69. 59–73.

[28] Siddharth Barman, Arpita Biswas, Sanath Kumar Krishna Murthy, and Yadati Narahari. 2018. Groupwise Maximin Fair Allocation of Indivisible Goods. In *AAAI*. AAAI Press, 917–924.

[29] Siddharth Barman, Arindam Khan, Sudarshan Shyam, and K. V. N. Sreenivas. 2022. Finding Fair Allocations under Budget Constraints. *CoRR* abs/2208.08168 (2022).

[30] Siddharth Barman and Sanath Kumar Krishnamurthy. 2019. On the Proximity of Markets with Integral Equilibria. In *AAAI*. AAAI Press, 1748–1755.

[31] Siddharth Barman and Sanath Kumar Krishnamurthy. 2020. Approximation Algorithms for Maximin Fair Division. *ACM Trans. Economics and Comput.* 8, 1 (2020), 5:1–5:28.

[32] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Finding Fair and Efficient Allocations. In *EC*. ACM, 557–574.

[33] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. 2021. The Price of Connectivity in Fair Division. In *AAAI*. AAAI Press, 5151–5158.

[34] Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. 2020. Fair Division of Mixed Divisible and Indivisible Goods. In *AAAI*. AAAI Press, 1814–1821.

[35] Xiaohui Bei, Shengxin Liu, Xinhang Lu, and Hongao Wang. 2021. Maximin Fairness with Mixed Divisible and Indivisible Goods. In *AAAI*. AAAI Press, 5167–5175.

[36] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. 2021. (Almost Full) EFX Exists for Four Agents (and Beyond). *CoRR* abs/2102.10654 (2021).

[37] Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. 2011. The price of fairness. *Operations research* 59, 1 (2011), 17–31.

[38] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S. Zwicker. 2019. Almost Envy-Free Allocations with Connected Bundles. In *ITCS (LIPIcs, Vol. 124)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 14:1–14:21.

[39] Arpita Biswas and Siddharth Barman. 2018. Fair Division Under Cardinality Constraints. In *IJCAI*. ijcai.org, 91–97.

[40] Anna Bogomolnaia, Herve Moulin, Fedor Sandomirskiy, and Elena Yanovskaya. 2016. Dividing goods or bads under additive utilities. *arXiv preprint arXiv:1608.01540* (2016).

[41] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaya. 2017. Competitive division of a mixed manna. *Econometrica* 85, 6 (2017), 1847–1871.

[42] Shant Boodaghians, Bhaskar Ray Chaudhury, and Ruta Mehta. 2022.

Polynomial Time Algorithms to Find an Approximate Competitive Equilibrium for Chores. In *SODA*. SIAM, 2285–2302.

[43] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. 2017. Fair Division of a Graph. In *IJCAI*. ijcai.org, 135–141.

[44] Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. 2019. Chore division on a graph. *Auton. Agents Multi Agent Syst.* 33, 5 (2019), 540–563.

[45] Sylvain Bouveret and Michel Lemaître. 2014. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. In *AAMAS*. IFAAMAS/ACM, 1321–1328.

[46] Sylvain Bouveret and Michel Lemaître. 2016. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Auton. Agents Multi Agent Syst.* 30, 2 (2016), 259–290.

[47] Steven J Brams, Michael A Jones, Christian Klamler, et al. 2006. Better ways to cut a cake. *Notices of the AMS* 53, 11 (2006), 1314–1321.

[48] Steven J Brams and Alan D Taylor. 1995. An envy-free cake division protocol. *The American Mathematical Monthly* 102, 1 (1995), 9–18.

[49] Steven J. Brams and Alan D. Taylor. 1996. *Fair division - from cake-cutting to dispute resolution.* Cambridge University Press.

[50] Steven J Brams and Alan D Taylor. 1996. *Fair Division: From cake-cutting to dispute resolution.* Cambridge University Press.

[51] Simina Brânzei and Fedor Sandomirskiy. 2019. Algorithms for Competitive Division of Chores. *CoRR* abs/1907.01766 (2019).

[52] Johannes Brustle, Jack Dippel, Vishnu V. Narayan, Mashbat Suzuki, and Adrian Vetta. 2020. One Dollar Each Eliminates Envy. In *EC*. ACM, 23–39.

[53] Eric Budish. 2010. The combinatorial assignment problem: approximate competitive equilibrium from equal incomes. In *BQGT*. ACM, 74:1.

[54] Ioannis Caragiannis, Nick Gravin, and Xin Huang. 2019. Envy-Freeness Up to Any Item with High Nash Welfare: The Virtue of Donating Items. In *EC*. ACM, 527–545.

[55] Ioannis Caragiannis and Stavros Ioannidis. 2021. Computing Envy-Freeable Allocations with Limited Subsidies. In *WINE (Lecture Notes in Computer Science, Vol. 13112)*. Springer, 522–539.

[56] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. 2012. The efficiency of fair division. *Theory of Computing Systems* 50 (2012), 589–610.

[57] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Economics and Comput.* 7, 3 (2019), 12:1–12:32.

[58] Mithun Chakraborty, Ayumi Igarashi, Warut Suksompong, and Yair

Zick. 2020. Weighted Envy-Freeness in Indivisible Item Allocation. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems, 231–239.

[59] Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. 2021. Competitive Allocation of a Mixed Manna. In *SODA*. SIAM, 1405–1424.

[60] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. 2020. EFX Exists for Three Agents. In *EC*. ACM, 1–19.

[61] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. 2020. A Little Charity Guarantees Almost Envy-Freeness. In *SODA*. SIAM, 2658–2672.

[62] Yiling Chen, John K Lai, David C Parkes, and Ariel D Procaccia. 2013. Truth, justice, and cake cutting. *Games and Economic Behavior* 77, 1 (2013), 284–297.

[63] Edward G Coffman, János Csirik, Gábor Galambos, Silvano Martello, and Daniele Vigo. 2013. Bin packing approximation algorithms: survey and classification. In *Handbook of combinatorial optimization*. 455–531.

[64] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. 2017. Fair Public Decision Making. In *EC*. ACM, 629–646.

[65] Sijia Dai, Guichen Gao, Shengxin Liu, Boon Han Lim, Li Ning, Yicheng Xu, and Yong Zhang. 2022. EFX Under Budget Constraint. In *FAW (Lecture Notes in Computer Science, Vol. 13461)*. Springer, 3–14.

[66] Sina Dehghani, Alireza Farhadi, Mohammad Taghi Hajiaghayi, and Hadi Yami. 2018. Envy-free Chore Division for An Arbitrary Number of Agents. In *SODA*. SIAM, 2564–2583.

[67] Amitay Dror, Michal Feldman, and Erel Segal-Halevi. 2021. On Fair Division under Heterogeneous Matroid Constraints. In *AAAI*. AAAI Press, 5312–5320.

[68] Lester E Dubins and Edwin H Spanier. 1961. How to cut a cake fairly. *The American Mathematical Monthly* 68, 1P1 (1961), 1–17.

[69] Ronald Dworkin. 1981. What is equality? Part 1: Equality of welfare. *Philosophy & public affairs* (1981), 185–246.

[70] Ronald Dworkin. 1981. What is equality? Part 2: Equality of resources. *Philosophy & public affairs* (1981), 283–345.

[71] Jeff Edmonds and Kirk Pruhs. 2006. Balanced Allocations of Cake. In *FOCS*. IEEE Computer Society, 623–634.

[72] Jeff Edmonds and Kirk Pruhs. 2006. Cake cutting really is not a piece of cake. In *SODA*, Vol. 6. 271–278.

[73] Francis Edward Su. 1999. Rental harmony: Sperner's lemma in fair division. *The American mathematical monthly* 106, 10 (1999), 930–942.

[74] Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. 2021. Keep Your Distance: Land Division With Separation. In *IJCAI*. ijcai.org, 168–174.

[75] Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. 2022. Mind the gap: Cake cutting with separation. *Artificial Intelligence* 313 (2022), 103783.

[76] Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. 2023. Keep your distance: Land division with separation. *Computational Geometry* (2023), 102006.

[77] Shimon Even and Azaria Paz. 1984. A note on cake cutting. *Discrete Applied Mathematics* 7, 3 (1984), 285–296.

[78] Alireza Farhadi, MohammadTaghi Hajiaghayi, Mohammad Ghodsi, Sébastien Lahaie, David M. Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2017. Fair Allocation of Indivisible Goods to Asymmetric Agents. In *AAMAS*. ACM, 1535–1537.

[79] Uriel Feige, Ariel Sapir, and Laliv Tauber. 2021. A Tight Negative Example for MMS Fair Allocations. In *WINE (Lecture Notes in Computer Science, Vol. 13112)*. Springer, 355–372.

[80] AM Fink. 1964. A note on the fair division problem. *Mathematics Magazine* 37, 5 (1964), 341–342.

[81] Marc Fleurbaey. 1996. *Théories économiques de la justice*. Economica.

[82] Yotam Gafni, Xin Huang, Ron Lavi, and Inbal Talgam-Cohen. 2021. Unified Fair Allocation of Goods and Chores via Copies. *CoRR* abs/2109.08671 (2021).

[83] DR GEORGE GAMON and DR MARVIN STERN. [n. d.]. Puzzle-Math. ([n. d.]).

[84] Jiarui Gan, Bo Li, and Xiaowei Wu. 2021. Approximately Envy-Free Budget-Feasible Allocation. *CoRR* abs/2106.14446 (2021).

[85] Jugal Garg and Aniket Murhekar. 2021. Computing Fair and Efficient Allocations with Few Utility Values. In *SAGT (Lecture Notes in Computer Science, Vol. 12885)*. Springer, 345–359.

[86] Jugal Garg and Setareh Taki. 2021. An improved approximation algorithm for maximin shares. *Artif. Intell.* 300 (2021), 103547.

[87] Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2021. Fair Allocation of Indivisible Goods: Improvement. *Math. Oper. Res.* 46, 3 (2021), 1038–1053.

[88] Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2022. Fair allocation of indivisible goods: Beyond additive valuations. *Artif. Intell.* 303 (2022), 103633.

[89] Paul Goldberg, Alexandros Hollender, and Warut Suksompong. 2020. Contiguous cake cutting: Hardness results and approximation algorithms. *Journal of Artificial Intelligence Research* 69 (2020), 109–141.

[90] Jonathan R. Goldman and Ariel D. Procaccia. 2014. Spliddit: unleashing fair division algorithms. *SIGecom Exch.* 13, 2 (2014), 41–46.

[91] Laurent Gourvès and Jérôme Monnot. 2019. On maximin share allocations in matroids. *Theor. Comput. Sci.* 754 (2019), 50–64.

[92] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. 2014. Near Fairness in Matroids. In *ECAI (Frontiers in Artificial Intelligence and Applications, Vol. 263)*. IOS Press, 393–398.

[93] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. 2015. Worst case compromises in matroids with applications to the allocation of indivisible goods. *Theor. Comput. Sci.* 589 (2015), 121–140.

[94] Robert Grandl, Ganesh Ananthanarayanan, Srikanth Kandula, Sriram Rao, and Aditya Akella. 2014. Multi-resource packing for cluster schedulers. In *SIGCOMM*. ACM, 455–466.

[95] Daniel Halpern and Nisarg Shah. 2019. Fair Division with Subsidy. In *SAGT (Lecture Notes in Computer Science, Vol. 11801)*. Springer, 374–389.

[96] Daniel Halpern and Nisarg Shah. 2021. Fair and Efficient Resource Allocation with Partial Information. In *IJCAI*. ijcai.org, 224–230.

[97] Sönke Hartmann and Dirk Briskorn. 2022. An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research* 297, 1 (2022), 1–14.

[98] Jiafan He, Ariel D. Procaccia, Alexandros Psomas, and David Zeng. 2019. Achieving a Fairer Future by Changing the Past. In *IJCAI*. ijcai.org, 343–349.

[99] Theodore P Hill. 1983. Determining a fair border. *The American Mathematical Monthly* 90, 7 (1983), 438–442.

[100] Theodore P Hill. 1987. Partitioning general probability measures. *The Annals of Probability* (1987), 804–813.

[101] Hadi Hosseini, Andrew Searns, and Erel Segal-Halevi. 2022. Ordinal Maximin Share Approximation for Chores. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 597–605.

[102] Hadi Hosseini, Andrew Searns, and Erel Segal-Halevi. 2022. Ordinal Maximin Share Approximation for Goods. *J. Artif. Intell. Res.* 74 (2022).

[103] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2021. Fair and Efficient Allocations under Lexicographic Preferences. In *AAAI*. AAAI Press, 5472–5480.

[104] Xin Huang and Pinyan Lu. 2021. An Algorithmic Framework for Approximating Maximin Share Allocation of Chores. In *EC*. ACM, 630–631.

[105] Halvard Hummel and Magnus Lie Hetland. 2022. Guaranteeing Half-Maximin Shares Under Cardinality Constraints. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 1633–1635.

[106] Ayumi Igarashi and Dominik Peters. 2019. Pareto-Optimal Allocation

of Indivisible Goods with Connectivity Constraints. In *AAAI*. AAAI Press, 2045–2052.

[107] Ian Kash, Ariel D Procaccia, and Nisarg Shah. 2014. No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research* 51 (2014), 579–603.

[108] Koki Kubo and Yoshifumi Manabe. 2016. A non-blocking online cake-cutting protocol. In *2016 Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*. IEEE, 258–263.

[109] Rucha Kulkarni, Ruta Mehta, and Setareh Taki. 2021. Indivisible Mixed Manna: On the Computability of MMS+PO Allocations. In *EC*. ACM, 683–684.

[110] David Kurokawa. 2017. Fair Division in Game Theoretic Settings. *PhD thesis, Carnegie Mellon University* (2017).

[111] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. 2016. When Can the Maximin Share Guarantee Be Guaranteed?. In *AAAI*. AAAI Press, 523–529.

[112] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. 2018. Fair Enough: Guaranteeing Approximate Maximin Shares. *J. ACM* 65, 2 (2018), 8:1–8:27.

[113] Maria Kyropoulou, Warut Suksompong, and Alexandros A. Voudouris.

2019. Almost Envy-Freeness in Group Resource Allocation. In *IJCAI*. ijcai.org, 400–406.

[114] Bo Li, Yingkai Li, and Xiaowei Wu. 2022. Almost (Weighted) Proportional Allocations for Indivisible Chores. In *WWW*. ACM, 122–131.

[115] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *EC*. ACM, 125–131.

[116] Zbigniew Lonc and Miroslaw Truszczynski. 2018. Maximin Share Allocations on Cycles. In *IJCAI*. ijcai.org, 410–416.

[117] Evangelos Markakis and Christos-Alexandros Psomas. 2011. On Worst-Case Allocations in the Presence of Indivisible Goods. In *WINE (Lecture Notes in Computer Science, Vol. 7090)*. Springer, 278–289.

[118] Nicholas Mattei, Abdallah Saffidine, and Toby Walsh. 2017. Mechanisms for Online Organ Matching. In *IJCAI*. ijcai.org, 345–351.

[119] Nicholas Mattei, Abdallah Saffidine, and Toby Walsh. 2018. Fairness in deceased organ matching. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 236–242.

[120] Neeldhara Misra, Chinmay Sonar, P. R. Vaidyanathan, and Rohit Vaish. 2021. Equitable Division of a Path. *CoRR* abs/2101.09794 (2021).

[121] Elchanan Mossel and Omer Tamuz. 2010. Truthful fair division. In *Al-*

gorithmic Game Theory: Third International Symposium, SAGT 2010, Athens, Greece, October 18-20, 2010. Proceedings 3. Springer, 288–299.

[122] Hervé Moulin. 2004. *Fair division and collective welfare*. MIT press.

[123] Hervé Moulin. 2019. Fair division in the internet age. *Annual Review of Economics* 11 (2019), 407–441.

[124] Hoon Oh, Ariel D. Procaccia, and Warut Suksompong. 2019. Fairly Allocating Many Goods with Few Queries. In *AAAI*. AAAI Press, 2141–2148.

[125] Benjamin Plaut and Tim Roughgarden. 2018. Almost Envy-Freeness with General Valuations. In *SODA*. SIAM, 2584–2603.

[126] Ariel D. Procaccia, Rodrigo A. Velez, and Dingli Yu. 2018. Fair Rent Division on a Budget. In *AAAI*. AAAI Press, 1177–1184.

[127] Ariel D. Procaccia and Junxing Wang. 2014. Fair enough: guaranteeing approximate maximin shares. In *EC*. ACM, 675–692.

[128] John Rawls. 1971. Atheory of justice. *Cambridge (Mass.)* (1971).

[129] Jack M. Robertson and William A. Webb. 1998. *Cake-cutting algorithms - be fair if you can*. A K Peters.

[130] John E Roemer. 1996. *Theories of distributive justice*. Harvard University Press.

[131] Masoud Seddighin and Saeed Seddighin. 2022. Improved Maximin

Guarantees for Subadditive and Fractionally Subadditive Fair Allocation Problem. In *AAAI*. AAAI Press, 5183–5190.

[132] Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. 2017. Fair and square: Cake-cutting in two dimensions. *Journal of Mathematical Economics* 70 (2017), 1–28.

[133] Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. 2020. Envy-free division of land. *Mathematics of Operations Research* 45, 3 (2020), 896–922.

[134] Amartya Sen. 1970. Collective Choice and Social Welfare. (1970).

[135] Hugo Steinhaus. 1948. The problem of fair division. *Econometrica* 16 (1948), 101–104.

[136] Walter Stromquist. 1980. How to cut a cake fairly. *The American Mathematical Monthly* 87, 8 (1980), 640–644.

[137] Walter Stromquist. 2008. Envy-free cake divisions cannot be found by finite protocols. *the electronic journal of combinatorics* 15, 1 (2008), R11.

[138] Warut Suksompong. 2017. Fairly Allocating Contiguous Blocks of Indivisible Items. In *SAGT (Lecture Notes in Computer Science, Vol. 10504)*. Springer, 333–344.

[139] Warut Suksompong. 2021. Constraints in fair division. *SIGecom Exch.* 19, 2 (2021), 46–61.

[140] Hal R Varian. 1973. Equity, envy, and efficiency. (1973).

[141] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *EuroSys*. ACM, 18:1–18:17.

[142] Toby Walsh. 2011. Online cake cutting. In *Algorithmic Decision Theory: Second International Conference, ADT 2011, Piscataway, NJ, USA, October 26-28, 2011. Proceedings 2*. Springer, 292–305.

[143] Toby Walsh. 2011. Online Cake Cutting. In *ADT (Lecture Notes in Computer Science, Vol. 6992)*. Springer, 292–305.

[144] Xiaowei Wu, Bo Li, and Jiarui Gan. 2020. Budget-feasible Maximum Nash Social Welfare Allocation is Almost Envy-free. *CoRR* abs/2012.03766 (2020).

[145] Shengwei Zhou and Xiaowei Wu. 2022. Approximately EFX Allocations for Indivisible Chores. In *IJCAI*. ijcai.org, 783–789.