Fair and Efficient Resource Allocation

Thesis submitted in partial
fulfillment of the requirements of the degree of

Master of Science

in

Computer Science and Engineering by Research

by

Shaily Mishra
2019701007
shaily.mishra@research.iiit.ac.in

Advised by Dr. Sujit Prakash Gujar

International Institute of Information Technology
Hyderabad - 500 032, INDIA
October, 2023

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Fair and Efficient Resource Allocation" by Shaily Mishra, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Sujit P Gujar

To, my family and friends,

who provided me with unwavering love and continuous inspiration and support during this journey.

Acknowledgments

I am immensely grateful for the opportunity to pursue my studies at IIIT Hyderabad and to have been a part of the Machine Learning Lab (MLL). I want to express my deep appreciation to Professor Sujit Gujar, my advisor, whose unwavering support, encouragement, and positive outlook have been instrumental in my personal and academic growth. Prof. Sujit consistently challenged me to exceed my limits and provided invaluable guidance and feedback, enabling me to expand my knowledge, implement new ideas, and conduct thorough research. Even in the face of rejection, Prof. Sujit always reassured me of the value of my research work and motivated me to persevere. I am genuinely grateful for his support throughout my academic journey.

Not only has Prof. Sujit been an exceptional mentor, but he has also fostered a sense of family within our lab. Daily tea breaks, team lunches, and dinners he organized have created a strong sense of community and togetherness. I am deeply thankful for the opportunity to collaborate with other lab members, including Manisha Padala, Sankarshan Damle, Samhita Kanaparthy, Anurag Jain, Debojit Das, Shantanu Das, Varul Srivastava, and Sanjay Chandlekar. Each of them has brought unique perspectives and experiences to our discussions, teaching me invaluable lessons on problem-solving and critical thinking.

My gratitude extends to my mentor, Manisha Padala, who has been with me from the beginning and without whom my work would not have been possible. Their enthusiasm, assistance, and willingness to engage in discussions have constantly inspired me within our vibrant work culture. I also want to express my heartfelt thanks to all of my labmates for their unwavering support and constructive feedback, which have made my journey at MLL even more enjoyable.

I would be remiss if I didn't acknowledge the unwavering support of my parents, who have been my pillars of strength through every challenge. Their love and positivity have sustained me in every situation, and I cannot thank them enough for always standing by my side and instilling confidence in me.

I also take this opportunity to express my sincere gratitude to all the teachers who have played a pivotal role in shaping the person I am today. Your guidance and mentorship have been invaluable, and I am forever grateful for the profound impact you have had on my life.

Lastly, I want to extend my heartfelt appreciation to my dear friends Pranali Raval, Apoorva Srivastava, Sai Amreet, Neel Mishra, Parth Mishra, and Vigneshwar Ilango for their unwavering support and inspiration. Your encouragement and belief in me have meant more than words can express. Special thanks go to BTS for being a constant source of inspiration.

Last but certainly not least, I would like to express my gratitude to all the staff and administration members who have supported me throughout my college journey.

Abstract

We address the problem of achieving fair and efficient allocation of resources in the era of rapid technological advancements, including Artificial Intelligence (AI) and the Internet of Things (IoT). The allocation of resources among autonomous agents is a crucial problem, and achieving fairness while ensuring efficiency is challenging. The objective is to determine the best way to divide resources among interested agents so that everyone is happy with their allocation, while ensuring fairness. We explore various fairness notions and relaxations that could apply to the problem of resource allocation. One of the main issues addressed in the thesis is the challenge of achieving fairness in the allocation of indivisible resources. We address several critical questions related to fair division, such as how to ensure fairness, how to generalize the notion of fairness, and what criteria to choose for different contexts. On top of it, fair division for goods and chores requires different approaches. We examine this issue in more detail and survey the existing approaches and provide solutions to overcome the challenges. We provide a detail survey on fairness notion and their approximate relaxations and evaluates their effectiveness in achieving fair and efficient allocations.

In recent literature, researchers have delved into data-driven approaches in game theory and mechanism design to tackle the challenges of traditional approaches. Given the success of neural networks (NNs) in learning algorithms, and mechanisms, and solving mixed-integer programs, they offer a promising tool for achieving fair and efficient allocations of resources. Thus, we aim to use NNs to find an approximate fair allocation that maximizes social welfare in our pursuit of equitable and efficient resource distribution. Another critical

issue addressed in the thesis is the presence of externalities, where the utility of an agent depends not only on their allocated resources but also on the resources allocated to other agents. Such scenarios are prevalent, especially in allocating critical resources such as hospital beds, ventilators, and vaccines during the COVID-19 pandemic. We evaluate the challenges of achieving fair and efficient allocations in the presence of externalities and propose potential solutions.

Overall, in this thesis, we provide a comprehensive analysis of the challenges of achieving fair and efficient allocation of resources in the era of rapid technological advancements. The research proposes efficient algorithms and solutions to overcome the challenges and achieve approximately fair allocation in different contexts. Achieving fair and efficient resource allocation is crucial for various domains, including economics, politics, and social welfare, and the results of this could have significant implications in these domains.

Research Papers from the Thesis Work

Conference Papers

1. Shaily Mishra, Padala Manisha, Sujit Gujar. "Fair Allocation with Special Externalities." In The 19th Pacific Rim International Conferences on Artificial Intelligence 2022 **Best Paper Award Runner Up (PRICAI'22)**

2. Shaily Mishra, Padala Manisha, Sujit Gujar. "EEF1-NN: Efficient and EF1 Allocations Through Neural Networks." In The 19th Pacific Rim International Conferences on Artificial Intelligence 2022 **(PRICAI 2022)**

Under Review

1. Shaily Mishra, Padala Manisha, Sujit Gujar. "Fair Allocation Of Goods And Chores – Tutorial And Survey of Recent Results" **(arxiv.org/pdf/2307.10985.pdf)**

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *"Is not the whole land before thee? Separate thyself, I pray thee; if thou wilt take the left hand, then I will go to the right; or if thou depart to the right hand, I will go to the left."* — Bible Genesis 13

In the book of Genesis, specifically in Chapter 13, a significant event takes place where Abraham and Lot decide to part ways. This decision was prompted by the need to divide the land of Canaan between them. Abraham proposed a fair division of the land and let Lot have the first choice in selecting his portion.

Let's consider the context of distributing vaccinations. The recent COVID-19 pandemic has created an urgent and substantial demand for critical life-saving resources such as hospital beds, ventilators, and vaccines. Unfortunately, there has been a significant imbalance in the global distribution of these resources. To illustrate this, suppose we have only three vaccinations remaining and need to allocate them among seven individuals: a family of four (parents and two children), an elderly person, a young healthy person, and a young person with underlying health conditions. In this scenario, determining a fair distribution of these vaccinations and establishing a method to calculate it becomes a crucial question. Additionally, since these resources are limited, we also aim to ensure an efficient allocation of resources. Therefore, our priority is not only achieving fairness but also ensuring efficiency when allocating items among a group of individuals.

Due to the rapid advancement in technology, such as Artificial Intelligence (AI) and the Internet of Things (IoT), many autonomous agents encounter the challenge of task or resource allocation. In this setting, we explore several questions. Firstly, we define what constitutes a fair allocation and whether such allocations even exist. If they do, how can we compute them? Alternatively, if fair allocations do not exist, can we relax our fairness criteria? Once we have achieved a fair allocation, can we further obtain an allocation that maximizes efficiency among the set of fair allocations? How can we achieve both fairness and efficiency in our allocations? This thesis primarily focuses on addressing fundamental questions concerning the attainment of *fair and efficient allocations*.

To illustrate, we'll look at a simple example.

Example 1.1. *Two intelligent friends, Agastya and Noorie, won a contest and received an uncut veggie pizza as a reward. These friends need to divide this pizza fairly.*

Suppose both of them like veggie pizza equally; hence the fairest way to divide it is to cut it equally. Let's say that the pizza has various toppings, such as pineapple, mushroom, olives, and corn. If both like all the toppings, they can cut the pizza equally for each topping. However, what if they have their preferences? Imagine Agastya dislikes pineapple toppings, and Noorie is allergic to mushrooms. Can they cut the pizza according to their preferences?

What if they used Abraham and Lot's method? Noorie suggests splitting the pizza in two, and Agastya could choose first what he likes, while she will take the remaining piece. Would this be fair? Let's analyze this approach. Noorie cuts the pizza in such a way that both the partitions are equal to her. As a result, she is happy with any piece she gets. Agastya picks the pizza part he likes the most. This division is famously known as the *Cut-and-Choose algorithm, 'I cut, you choose'*.

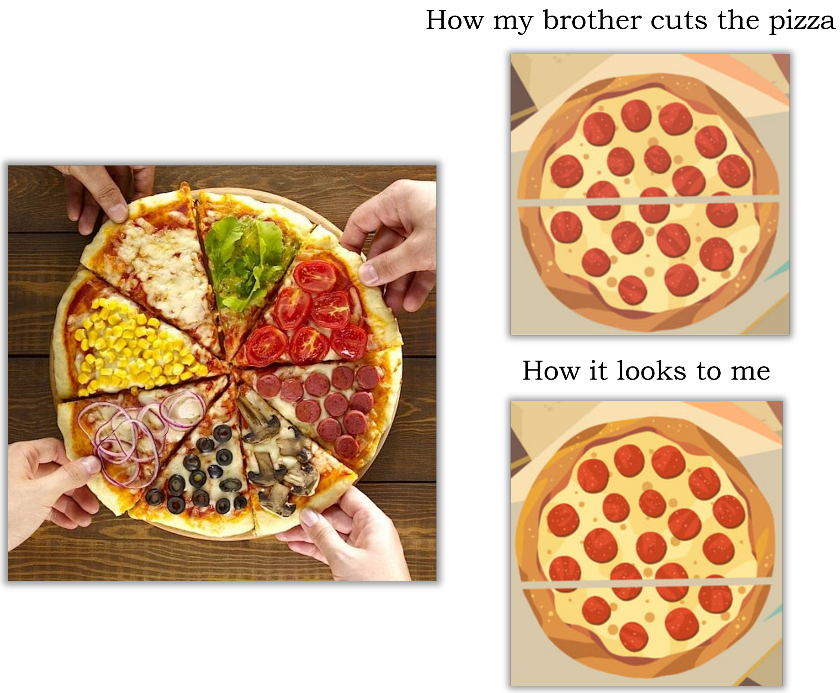

Figure 1.1: Pizza Cutting Example [88, 89]

The above pizza was uncut, and the agents were free to cut the pizza in any way they liked. These types of items are called *divisible items*.

Example 1.2. *Let us, however, consider a modification where Agastya and Noorie receive eight pieces of pizza and cannot cut the pieces any further. Either they take the whole piece, or they don't. They both like every piece equally. This setting is called as* indivisible items. *Each person can take four pieces, so the allocations are fair. Consider, however, if they only received seven pieces. Is it possible to distribute the pieces fairly among them in such a case?*

Before we can answer this, let's look at the following situation.

Example 1.3. *Agastya and Noorie need to clean their house. This includes tasks such as cleaning the living room, clearing the kitchen mess, dusting the furniture, etc. Having fewer chores is what everyone wants. What is the best way to divide the tasks such that everyone gets their fair share?*

Comparing Example 1.1 to Example 1.3, in the first one, everyone would like to obtain a bigger share, while in the second, everyone is more interested in obtaining a smaller share. It is commonly known (in the literature) that items (resources/tasks) with such positive value are called *goods*, and those with negative value are called *chores*. The approach to solving fair division for goods and chores is quite different, even though it may seem like just a matter of negation. We will examine this in more detail later.

Suppose there are multiple friends – *agents* in the system. How should we generalize this notion of *fairness*? The idea that agents should be happy with their allocations (allocated bundles) is *envy-freeness* [72]. Envy-freeness means agents/players value their bundle more than others, i.e., no agent envies others. We can always divide divisible items without envy. Allocating indivisible items without inciting envy is, however, a challenging task. For example, if there is only one good, and Noorie receives it, Agastya will envy her. It is widely known that EF is not guaranteed to exist for indivisible items.

With the negative results concerning the existence of envy-freeness for indivisible items, the question naturally arises, can we ensure envy-freeness approximately, i.e., to relax the fairness notion? For Example 1.2, giving three of the seven pieces to one person and the remaining four to another is arguably the fairest way to allocate them. Assuming Agastya receives three pieces and Noorie receives four, he envies Noorie by at most one piece. Budish introduced such a relaxation of envy-freeness, called *Envy-freeness up to 1 item (EF1)* [51]. We will formally define all the fairness notions in detail later on, but in essence, EF1 implies that if envy exists in the system, an agent can only envy the other agent by at most one item.

Finding a fair allocation in a simple example involving two friends was challenging. Here, we present a highly complex example to illustrate the problem's difficulty. Imagine that the Government wishes to invest in improving national security, economy, agriculture, and healthcare. It picks four renowned scientists in these individuals' domains and asks them to set up a research lab for the nation's improvement. It provides them with a wide variety of resources such as professors, laboratory assistants, engineers, vehicles, books, subscriptions, journals, drones, microscopes, biometrics systems, computer servers, land, water facility, electricity facility, supercomputers, etc. This intricate scenario involves distributing many divisible and indivisible goods and responsibilities to the four scientists forming Research Centers. The Government wants to ensure that all four centers feel equal so that each one gives its best to develop this project. In addition, it wants to ensure that the resources are allocated to the Lab, which will put them to best use; for example, microscopes will benefit the pathology department in the Healthcare Center rather than the Economic Center. There are divisible items such as land, electric facilities, water facilities, etc., and indivisible items such as computer systems, servers, books, etc. An item may be beneficial to one person but a chore to another. Microscopes, for example, may be valued positively by the Healthcare Center but negatively by the Economic Center since they are of no use to them. We should also strive to ensure *efficiency* among fair allocations; since, otherwise, not distributing any item to an agent is fair. We are looking for a *fair and efficient* allocation.

It's hard to divide things fairly for a reason, however. The notion of fairness depends heavily on the context; what may be fair in one setting may not be fair in another. Suppose Agastya and Noorie are supposed to purchase a new refrigerator. Agastya likes Samsung, and Noorie likes LG. Regardless of what kind of refrigerator they buy, the concept of envy is no longer applicable. We choose fairness criteria depending on the context. Later, we will discuss various kinds of fairness notions in detail. In essence, we will aim to formulate broadly acceptable fairness criteria. In doing so, we seek to determine whether it always exists. If it does, can we find it in a reasonable amount of time? In essence, we will aim to formulate broadly acceptable fairness criteria. In doing so, we seek to determine whether it always exists. If it does, can we find it in a reasonable amount of time? Is it possible to relax the fairness criteria to apply to the setting if one does not exist?

It is an *NP-complete* problem to determine whether there is an envy-free allocation when there are indivisible items, even when the valuation is binary [15, 86]. Even if it exists, finding such an allocation is *NP-hard* [42]. It is natural to look for fairness relaxations that exist along with polynomial-time algorithms. Researchers are interested in studying relaxed fairness notions in light of the impossibility of fairness for indivisible items. Relaxations can be categorized into two: additive relaxations and multiplicative relaxations. Ideally, if we cannot find a fair distribution, we would prefer an allocation that approximates the fairness guarantees in some multiplicative way. Suppose, for example, that we cannot find a fair share allocation; we're aiming to find an allocation that approximates the fair share in some multiplicative way for each agent. However, we see that fairness is difficult to achieve even with a multiplicative approximation. Therefore, we turn to additive approximation, which exhibits both possibilities and impossibilities.

In addition to finding fair allocation, *eliciting agent preferences* is a difficult task. Consider dividing 30 items among several agents. To provide a complete valuation of items, each agent should provide their value for every bundle of the items, i.e., they would have to evaluate the value of more than 1 billion bundles. In order to capture agents' preferences in a simple manner, we look for ways to model agents' valuations. We will discuss these shortly.

Now we define these notions informally and present an example demonstrating how they can be applied. We state four popular fairness criteria below, Envy-freeness (EF), Proportionality (PROP), Maximin Share (MMS), and Equitable (EQ).

Fairness Notions.

Envy-freeness (EF) [72] is the most common notion of fairness. According to this theory, no agent should value the allocations of others more than what they value for their own. An envy-free allocation is one in which no agent envies another. Therefore, every agent feels their bundle is at least as good as anyone else's.

Proportionality (PROP) [131] ensures that all agents receive allocations which they value at least $1/n$ of their total value, i.e., their proportional value.

Next, we consider a relaxation of PROP, the *max-min share (MMS)* allocation.

Maximin share (MMS) [51] ensures that all agents receive an allocation worth at least its MMS share value. Imagine we asked the agent to divide the items into n bundles and take the bundle with the lowest value. The agent would split the bundles to maximize the minimum utility, their MMS share.

Equitable (EQ) [63] ensures that agents value their allocation at the same level as other agents. By EQ, we mean that all agents value their allocations equally.

It is common to ensure some efficiency notion, along with fairness. Or else, assigning no item to anyone will always be fair. Efficiency is yet another crucial aspect of item allocation. Informally, we mention popular efficiency notions such as Pareto Optimality (PO), Maximum Utilitarian Welfare (MUW), Maximum Nash Welfare (MNW), and Maximum Egalitarian Welfare (MEW).

Efficiency Notions.

Pareto Optimal(PO). An allocation is Pareto optimal if no other allocation is Pareto dominated by it, i.e., no other allocation makes some agents strictly better without making other agents worse.

Utilitarian Social Welfare (USW) A Utilitarian Welfare Function is the sum of the individual agents' valuation for their allocation. It assumes that agents are treated equally. An allocation that maximizes USW is known as Maximum Utilitarian Welfare (MUW).

Nash Social Welfare (NSW) A Nash Welfare Function is the product of the individual agents' valuation for their allocation. An allocation that maximizes NSW is known as

Maximum Nash Welfare (MNW). As we will discuss later, NSW has several interesting characteristics that combine fairness and efficiency.

Egalitarian Social Welfare (ESW) An Egalitarian Welfare Function is the minimum individual agents' valuation for their allocation. An allocation that maximizes ESW is known as Maximum Egalitarian Welfare (MEW). The philosophy is to maximize the most marginalized agents in the system. This welfare is also known as the Santa Claus Problem.

Now, we will illustrate all these notions using another example.

		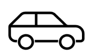					**total**
Agent 1	1	6	1	2	1	1	12
Agent 2	4	1	2	1	1	3	12
Agent 3	1	1	3	3	3	1	12

Figure 1.2: Example to illustrate fairness and efficiency notions

The above example has three agents, agent1, agent2, and agent3, and six items, plants (P), a car (C), books (B), a laptop (L), markers (M), and fruits (F). In total, all agents value complete items at 12. Consider allocation $A_1 = \{(C, F), (T), (B, M, L)\}$, i.e., (C, F) is allocated to agent1, (T) is allocated to agent2 and (B, M, L) is allocated to agent3. This allocation is envy-free since no agent feels envious of others. Let us consider another allocation $A_2 = \{(L, M, F), (T), (C, B)\}$. This allocation is not an envy-free allocation as agent1 envies agent2. However, it is a proportional allocation. The proportional value of each agent in this example is 4, and each agent has a value of 4 in A_2. Also, note that A_2 is an EQ allocation since all agents value their bundle equally, i.e., 4. The maximin share of agent1, agent2, and agent3 is 3,4 and 4. Allocation $A_3 = \{(L, M), (T), (C, B, F)\}$ is not a proportional allocation; however it is an MMS allocation, since all agents receive at least their MMS value. We will explore the relationship between all the notions in later sections. Allocation $A_4 = \{(C), (T, F), (B, M, L)\}$ is MUW as well as EF. Also, note that there are multiple allocations satisfying fairness criteria. For example, allocations A_1, A_2, and A_3 satisfy MMS guarantees.

These ideas represent a general conception of fairness (efficiency) and may or may not correspond to fairness (efficiency) in every case. We will illustrate this by providing two examples.

Example 1.4. *Consider Agastya and Noorie have to mop their house (divisible chore) and allocate a bike among them (indivisible good). What is a fair allocation in this setting? How to fairly allocate a set of divisible and indivisible items among interested agents?*

We cannot consider EF fairness overall since it need not exist for indivisible items, so we must consider the relaxation of EF. Shall we consider EF for divisible and EF1 for indivisible items? So accordingly, both will clean the house, and any one of them will get the bike. Is this a fair allocation? A fairer allocation would have been that the one who cleans the house gets the bike. Hence we need to re-consider our fairness and efficiency notions in such settings.

Next, we consider an example of an externality. Externality implies that the agent's utility depends not only on their bundle but also on the bundles allocated to other agents. Such a scenario is relatively common, mainly in allocating necessary commodities. For example, the COVID-19 pandemic resulted in a sudden and steep requirement for life-supporting resources like hospital beds, ventilators, and vaccines. There has been a heavy disparity in handling resources across the globe. Even though there was a decrease in GDP worldwide, low-income countries suffered more than high-income countries [64].

Ensuring above fairness measures is non-trivial in the presence of externalities.

Example 1.5. *For example, let's imagine that Coke and Pepsi are required to split raw materials A and B between them. According to Coke, material A is worth 10, and material B is worth 5. Pepsi values material B at 10 and material A at 5. Pepsi suffers a loss of -100 if material A is assigned to Coke and -5 if material B is assigned to Coke. In the same way, Coke loses 100 if material B is allocated to Pepsi and 5 if material A is allocated to Pepsi. To illustrate, consider allocation* A^**, assigning material B to Coke and material A to Pepsi. This allocation value for Coke is 5-5=0, and the value for Pepsi is also 0.*

In Example 1.5, the proportional value for both agents, i.e., Coke and Pepsi, is 7.5. It is easy to verify that no allocation satisfies proportional guarantees for both agents. Proportionality does not consider externalities, i.e., it does not consider the dis-utility that results from not receiving items. Hence we need to redefine certain notions accordingly to the given setting. We will go over a wide range of settings in the later sections.

1.1 Problem Addressed

The thesis examines the fair and efficient allocation of indivisible goods, chores, or both. Our research addresses two problems in particular.

- There is extensive literature available for fair allocations, but most of the existing alogithms for fair division do not consider externalities. Under *externalities*, the utility of an agent depends not only on its allocation but also on other agents' allocation. An agent has a positive (negative) value for the assigned goods (chores). In this work, we study a special case of externality which we refer to as 2-D. In 2-D, an agent receives a positive or negative value for unassigned items independent of who receives them. We primarily focus on leveraging existing algorithms to 2-D. However, not all notions apply when externalities are present; therefore, we redefine them and study them in detail.

- Next, we bring our focus on *finding fair and efficient allocation*, an NP-Hard problem. Given the theoretical limitations, We aim to compute approximately fair and efficient allocation, i.e., EF1 and efficient allocation of goods/chores using a data-driven approach. *Neural Networks (NNs)* have proven to be superior to existing approaches in finding an optimal mapping between input and output data. For instance, Neural Networks are capable of learning algorithms (Kim et al. [94] learned the Viterbi algorithm for decoding), mechanisms (Duetting et al. [66] learned optimal auctions), or solving mixed-integer programs (Nair et al. [111] uses graph Neural Networks to solve MIPs). The success of Neural Networks has led us to aim for the formation of Neural Networks that learn approximately fair and efficient allocations.

Our major contributions are as follows,

1.2 Contributions

- In Chapter 2, we provide a comprehensive examination of fair allocation, delving into intricate details and presenting a thorough analysis. Furthermore, in Chapter 4, we present a meticulous investigation into the realm of mechanism design, specifically focusing on approaches based on Neural Networks.

- In Chapter 3, we present our study of fair allocation in present of externalities. We propose a simple valuation transformation and show that we can adapt existing algorithms using it to retain some of the fairness and efficiency notions in 2-D. However, proportionality doesn't extend in 2-D. We redefine PROP and its relaxation and show that we can adapt existing algorithms. Further, we prove that maximin share (MMS) may not have any multiplicative approximation in this setting. We propose

Shifted α-MMS, a novel way of approximating MMS in 2-D. Studying this domain is a stepping stone towards full externalities where ensuring fairness is much more challenging.

- In Chapter 5, we propose a Neural Network- *EEF1-NN* inspired by U-Net [121] to learn approximately fair and efficient allocation. By fair and efficient, we mean EEF1, i.e., efficient (maximizing utilitarian social welfare) and EF1. We transform our valuations and augment them with additional channels to enhance the network's performance. We have a series of convolutional and up-convolutional layers to learn EEF1; hence EEF1-NN is generalized for any number of agents and items. We sample valuations from various distributions and report the expected fairness and efficiency achieved. Even for large instances, our network performs well with more than 10 agents and 100 items. Moreover, we show that, for our setting, bagging multiple networks improves performance.

1.3 Organization of the Thesis

In Chapter 2, we survey the fair division of indivisible goods or/and chores in various fairness notions in depth. Then in Chapter 3, we describe fair division in the presence of special externalities. Recently, the EconCS community has been interested in learning mechanisms/algorithms based on Neural Networks, especially in the context of theoretical limitations. We survey the Neural Network-based approach in Chapter 4. Further, we describe our Neural Network-based solution for finding approximately fair and efficient allocations in Chapter 5. Finally, in Chapter 6, we conclude our thesis and outline directions for the future.

Chapter 2

Background for Fair Resource Allocation

Fair resource allocation is an important problem in many real-world scenarios, where resources such as goods and chores must be allocated among agents. However, ensuring fair allocation becomes more challenging when the resources are indivisible. In this chapter, we focus on the problem of fair allocation of indivisible goods and chores. We first introduce the definitions of fairness and efficiency in the context of goods and chores. We then present a survey of algorithms for each fairness criterion, including envyfreeness, proportionality, MMS, and their relaxations. Further, we discuss algorithms that, besides satisfying the above fairness notions, are efficient, such as Pareto Optimal and maximize Utilitarian Welfare. In our discussions, we provide details regarding the computational complexity of the algorithms. For each notion of fairness, we study the likelihood of finding fair allocations and the price of fairness. We extend our discussions to mixed instances of indivisible and divisible items while examining different valuation and allocation settings.

2.1 Preliminaries

First, we define all essential concepts pertaining to fairness and introduce all needed notation.

2.1.1 Fair Resource Allocation: Basic Model and Notations

We consider the problem $\langle N, M, V\rangle$ of allocating M indivisible resources/items among N interested agents with valuation profile $v \in V$, where $M = [m]$, i.e., $M = \{1, 2, \ldots, m\}$, $N = [n]$ and $m, n \in \mathbb{N}$. We define allocation $A \in \Pi_n(M)$ as the partition of M indivisible items into n bundles, i.e., $A = (A_1, A_2, \ldots, A_n)$, where A_i is the bundle allocated to agent i. We only allow complete allocation, and no two agents can receive the same item. That is, $A = (A_1, A_2, \ldots, A_n)$, s.t., $\forall i, j \in N$, $i \neq j; A_i \cap A_j = \emptyset$ and $\bigcup_i A_i = M$. We denote the allocation for all the agents except i as A_{-i}. An allocation is called partial if $\bigcup_i A_i \subset M$.

Goods vs Chores

Each agent $i \in N$ has a valuation function $v_i : 2^M \rightarrow \mathbb{R}$, and $v_i(S)$ represents its valuation for the subset $S \subseteq M$. We represent the valuation profile as $v = (v_1, v_2, \ldots, v_n)$. For all agents, the valuations are normalized, i.e., $v_i(\emptyset) = 0$. We denote the valuation of an item $k \in M$ for any agent $i \in N$ as $v_i(\{k\})$ or v_{ik}. For an agent i, an item $k \in M$ is a *good* if, $v_{ik} \geq 0$, and a *chore* if, $v_{ik} < 0$. Goods are desirable/positively valued and chores are undesirable/negatively valued items. There could be three settings - pure goods, pure chores, and a combination of goods and chores. In combination, $\forall k \in M$ it is possible that, for some agent $i \in N$, $v_{ik} > 0$ and for another agent $j \in N$, $v_{jk} < 0$. That is, an item may be good for one agent and a chore for another. Note that we use the terms – valuation and utility interchangeably in this chapter.

Different Types of Valuation Functions

In most of the literature, we assume that the valuation functions of agents are *monotonic* for goods and *anti-monotonic* for chores. If we increase goods in an agent's bundle, the utility increases, and if we increase chores, the utility decreases. Formally, we define it as follows,

Definition 2.1 (*Monotonicity*). *A valuation function v_i is monotonic if, $\forall S \subseteq T \subseteq M, v_i(S) \leq v_i(T)$.*

Definition 2.2 (*Anti-monotonicity*). *A valuation function v_i is anti-monotonic if, $\forall S \subseteq T \subseteq M, v_i(S) \geq v_i(T)$.*

We say that agents have *additive valuations* when, for each agent, the valuation of a non-empty bundle equals the sum of the valuations of the individual items. However, it is also possible that agents value an allocation more or less than the sum of the values of the individual items, i.e., super-additive or sub-additive. Furthermore, agents may have marginal decreases or increases in their valuation with every additional item, i.e., sub-modular, super-modular valuations.

In the most general setting, the agents may have arbitrary valuations for every subset i.e., the *general valuation* setting. General valuations are rich and complex functions. Consider dividing 30 items among several agents. To provide a complete valuation of items, each agent should provide their value for every bundle of the items, i.e., they would have to report the value of more than 1 billion bundles. Hence in literature additive valuations are widely studied.

Definition 2.3 (*Additive Valuations*). *Agents have additive valuations if, an agent i, $\forall i \in N$ values any non-empty bundle A_i as $v_i(A_i) = \sum_{k \in A_i} v_{ik}$.*

A valuation instance is said to be *identical* when all agents have the same valuation for all subsets of items, formally,

Definition 2.4 (*Identical Valuations*). *The valuations are identical if,* $\forall i, j \in N, \forall S \subseteq M, v_i(S) = v_j(S)$.

Often, agents do not have identical valuations, but they order the items likewise, i.e., agents have the same rank for the items. We call such valuations *Identical Ordering (IDO).*

Definition 2.5 (*Identical Ordering (IDO)*). *Valuations are IDO when when all agents agree on the same ranking of the items, i.e., for* $\forall i \in N$, $v_{i1} \geq v_{i2} \ldots \geq v_{im}$.

The cardinal cost functions of agents in an IDO instance may still differ. In approval-based settings, agents have binary valuations, where they either approve or disapprove an item.

Definition 2.6 (*Binary Valuations*). *The valuations are binary, if* $\forall i \in N$, $\forall k \subseteq M$, $v_{ik} \in \{1, 0\}$.

We next define fairness and efficiency properties considered in this paper.

2.1.2 Important Definitions.

Fairness Related Definitions

The concept of envy-freeness (EF) introduced by Foley 1967 [72] is a well-established notion of fairness. It ensures that no agent envies the bundle of any other agent. Unfortunately, an EF allocation of indivisible items may not exist; for example, when there is a single good and two agents, the agent who doesn't get the good feels envious of the agent that does. The researchers were interested in relaxing the concept of EF in order to limit the envy of every agent. They introduced the idea of ϵ-EF, which ensures that if envy exists, it is at most ϵ. However, ϵ-EF allocations also need not exist. The non-existence of EF and ϵ-EF has prompted researchers to propose additive relaxed notions of EF, such as EF1 and EFX.

For goods, an allocation is EF1 when all agents value their bundle at least as much as they value another agent's bundle with their *most* valued item removed. EFX is stronger than EF1 and requires that all agents value their bundle no less than the other agents' bundle with their *least* valued item removed. For chores, a similar definition applies, but unlike in goods, a chore is removed from the agent's bundle and then compared with the other agents'. In the case of a combination of goods and chores, EF1 implies that if an agent envies another agent, we can eliminate the envy either by virtually removing the most valued good from the other agent's bundle or removing the least favorite chore from the agent's own bundle. A standard definition is as follows,

Definition 2.7 (*Envy-free (EF) and relaxations* [14, 51, 55, 72, 137])**.** *For the items (goods or chores), an allocation A that satisfies $\forall i, j \in N$,*

$$v_i(A_i) \geq v_i(A_j) \;\; \textit{is}\ \text{EF}$$

$$\left.\begin{array}{l} v_{ik} < 0, v_i(A_i \setminus \{k\}) \geq v_i(A_j); \forall k \in A_i \\ v_{ik} > 0, v_i(A_i) \geq v_i(A_j \setminus \{k\}); \forall k \in A_j \end{array}\right\} \;\textit{is}\ \text{EFX}$$

$$v_i(A_i \setminus \{k\}) \geq v_i(A_j \setminus \{k\}); \exists k \in \{A_i \cup A_j\} \;\; \textit{is}\ \text{EF1}$$

$$v_i(A_i) + \epsilon \geq v_i(A_j) \;\; \textit{is}\ \epsilon\text{-EF}$$

$$v_i(A_i) \geq \alpha \cdot v_i(A_j) \;\; \textit{is}\ \alpha\text{-EF}$$

In addition to EF and its relaxations, proportional allocations are also well studied in the context of fairness. As introduced by Steihaus 1948 [131], Proportionality requires that each agent gets at least 1/n of their share of the total value. Furthermore, even proportional allocations may not always exist; for this reason, researchers have considered relaxations. Researchers then looked into α-prop, which requires that every agent receives an α fraction of its proportional share. But unfortunately, that solution does not exist as well. Similar to EF, we adapted additive relaxation for proportionality as well. Proportionality up to one item, PROP1 requires that every agent is guaranteed to receive their proportionality guarantee if they lose their least valued chore or receive their most valuable good from any other agent's bundle. Proportionality up to any item, PROPX requires that every agent is guaranteed to receive their proportionality guarantee if they lose their most valued chore or receive their least valued good allocated to another agent.

Definition 2.8 (*Proportionality (PROP)* [20, 24, 61, 131])**.** *For the items (chores or goods), an allocation A that satisfies $\forall i \in N$,*

$$v_i(A_i) \geq 1/n \cdot v_i(M) \ \textit{is}\ \text{PROP}$$

$$\left.\begin{array}{l} v_{ik} > 0, v_i(A_i \cup \{k\}) \geq 1/n \cdot v_i(M); \forall\ k \in \{M \setminus A_i\} \\ v_{ik} < 0, v_i(A_i \setminus \{k\}) \geq 1/n \cdot v_i(M); \forall\ k \in A_i \end{array}\right\} \ \textit{is}\ \text{PROPX}$$

$$\left.\begin{array}{l} v_i(A_i \cup \{k\}) \geq 1/n \cdot v_i(M); \exists\ k \in \{M \setminus A_i\}\ \textit{or,} \\ v_i(A_i \setminus \{k\}) \geq 1/n \cdot v_i(M); \exists\ k \in A_i \end{array}\right\} \ \textit{is}\ \text{PROP1}$$

$$v_i(A_i \cup \{k\}) \geq 1/n \cdot v_i(M); k \in \max_{j \neq i} \min_{k \in A_j} v_i(A_{jk}) \ \textit{is}\ \text{PROPm (only for goods)}$$

$$v_i(A_i) \geq \alpha \cdot 1/n \cdot v_i(M) \ \textit{is}\ \alpha\text{-PROP}$$

The content of this Chapter is available as Survey in [127]

Relationship between EF and PROP.

When agents have sub-additive valuations, Envy-freeness is stronger than proportionality, i.e., EF implies PROP. Similarly, when agents have additive valuations, EF1 implies PROP1. It may seem counter-intuitive at first glance that EFX implies PROPX when agents have additive valuations.

Following, we explore Maximin share (MMS) introduced by Budish 2011 [51], extending the concept of Cut and Choose to indivisible goods. Imagine telling an agent to divide the items into n bundles and pick the one they value the least. The agent would divide the items into n partitions so that they maximize the value of the least valued bundle. We call this value their Maximin Share. An allocation satisfies MMS guarantees if every agent receives at least their MMS value.

Definition 2.9 (*Maximin Share MMS* [51])**.** *An allocation A is said to be MMS if $\forall i \in N, v_i(A_i) \geq \mu_i$, where*

$$\mu_i = \max_{(A_1,A_2,\ldots,A_n)\in\prod_n(M)} \min_{j\in N} v_i(A_j)$$

Introduced by Procaccia and Wang [118], an allocation A is said to be α-MMS if it guarantees $v_i(A_i) \geq \alpha \cdot \mu_i$ when $\mu_i \geq 0$ and $v_i(A_i) \geq \frac{1}{\alpha} \cdot \mu_i$ when $\mu_i < 0$, where $\alpha \in (0, 1]$.

Efficiency Related Definitions

In the previous subsection, we discussed some popular fairness notions. Note that refusing to assign any item to any agent is also Envy-free. However, we also desire efficiency, so fairness is considered in connection with efficiency criteria as well. One of the most frequently studied efficiency criteria in fairness literature is Pareto-Optimality. A Pareto optimal (PO) allocation ensures that there is no other allocation which Pareto dominates, i.e., better for all agents and strictly better for at least one. It is interesting to consider PO and fair allocations.

Definition 2.10 (*Pareto-Optimal (PO)*)**.** *An allocation A' is said to Pareto dominate allocation A, if $\forall i \in N$, $v_i(A'_i) \geq v_i(A_i)$ and $\exists i \in N$, $v_i(A'_i) > v_i(A_i)$. When there is no allocation that Pareto dominates allocation A, it is said to be Pareto-optimal.*

We then consider utilitarian welfare, the sum of agents' utilities. On the other hand, Nash welfare corresponds to the product of agents' utilities, and egalitarian welfare, to the minimum of individual agents' utility.

Definition 2.11. *Given an instance $(N, M, \mathcal{V})$, an allocation A^* satisfies,*

Maximum Utilitarian Welfare, $MUW(v)$, if

$$A^* \in \arg\max_A \sum_{i=1}^{n} v_i(A_i) \tag{2.1}$$

Maximum Nash Welfare, $MNW(v)$ if

$$A^* \in \arg\max_A \prod_{i=1}^{n} v_i(A_i) \tag{2.2}$$

Maximum Egalitarian Welfare, $MEW(v)$ if

$$A^* = \arg\max_A \min_i v_i(A_i) \tag{2.3}$$

We proceed to the next section, summarizing the existential and computational results pertaining to various fairness notions. We discuss various results for goods in Section 2.2 and further categorize them in subsections 2.2.1 and 2.2.2, which deal with the results of determining the fair allocation and fair and efficient allocations, respectively. Our discussion of chores is similarly addressed in Section 2.3, and the combination of goods and chores is described in Section 2.4.

2.2 Indivisible Goods

2.2.1 Fair Allocation

Envy-Freeness (EF):

As defined in Definition 5.1, EF is a popular and strong notion of fairness. However, EF allocation may not exist for $n \geq 2$, for instance, imagine that we have to distribute a single good among n agents. Even when agents have binary additive valuations for goods, the problem of checking whether EF allocation exists or not is *NP-complete* [15, 86]. Hence, next, we study a prominent relaxation of EF, i.e., EF1 - Envy-free up to one item. The notion of EF1 was introduced by Budish 2011 [51] formally and implicitly it appeared in Lipton et al. 2004 [101].

Envy-Freeness up to one item (EF1):

As given in Definition 5.1, in the case of goods, Envy-free up to one item (EF1) implies that for any pair of agents i and j, if agent i envies agent j, the envy can be eliminated by virtually taking out agent i's most valuable item from j's bundle, i.e., $v_i(A_i) \geq v_i(A_j \setminus \{g\})$, $\exists g \in A_j$. EF1 allocation always exists and can be obtained in *polynomial time*.

Theorem 2.1 ([51, 55, 101]). *EF1 allocation always exist and can be found in polynomial time.*

When agents have *additive valuations*, round-robin algorithm gives EF1 for goods in polynomial time, i.e., $\mathcal{O}(mn \log m)$ [55]. We formally describe the steps in Algorithm 5. To allocate items in a round-robin fashion, we select an arbitrary sequence of agents. Each agent selects their favorite items when their turn comes. For any $i, j \in N$, if agent i picks before agent j, he/she will never envy agent j. If agent i picks after agent j, he/she may envy agent j's first selected item. Let's assume that we remove agent j's first pick, which was the item it most preferred. Because we removed agent j's first selected item, hypothetical agent i picks first. We know that the agent who comes earlier in the sequence has no envy for the agent who comes after them. Thus, agent i does not envy agent j if we remove agent j's first selected item, i.e., the envy is bounded up to one item.

Algorithm 1 *Round Robin*

Set $\forall i, A_i = \emptyset$
Every agent sort the items in decreasing order
Arrange agents in an arbitrary sequence w.l.o.g $\{1, \dots, n\}$
while $M \neq \emptyset$ **do**
  for $i \leftarrow 1$ *to* n **do**
    $A_i \leftarrow A_i \cup \arg\max_{k \in M} v_{ik}$
    $M \leftarrow M \setminus \{k\}$
  end for
end whilereturn Allocation A

When agents have *general monotone valuations*, Lipton et al. 2004 [101] proposed an envy-cycle elimination algorithm that gives EF1 allocation for goods in *polynomial time* $\mathcal{O}(mn^3)$. This algorithm bounds the envy of any agent by the maximum marginal value of any good; it corresponds to the notion of EF1 for goods. We define the envy graph first. An envy graph of allocation A consists of nodes for each agent and directed edges from agent i to agent j if i envies j, i.e., $v_i(A_i) < v_i(A_j)$. The algorithm selects an unenvied agent in each iteration, i.e., an agent with no edges directed towards them, and assigns an arbitrary good. If there are no such agents, there must be cycles of envy, and the agents can exchange bundles until no more cycles remain. Upon receiving the good, other agents may envy this agent; we can eliminate this envy by removing the good he/she just received since he/she were previously unenvied. Thus, if agent i envy agent j, the envy is bounded up to one good, i.e., by the recently added good in agent j's bundle. After each round of partial allocation, this algorithm ensures that EF1 is satisfied. In contrast, Bérczi et al. 2020 [38] showed that the envy-cycle elimination algorithm fails to find an EF1 allocation when agents have *general non-monotone valuations*.

Algorithm 2 *Envy-Cycle Elimination*

```
Set ∀i, A_i = ∅
for k ← 1 to m do
    Find an unenvied agent i
    A_i ← A_i ∪ k
    if Envy-Cycle exists then
        Swap bundles to resolve
    end if
end forreturn Allocation A
```

Apart from EF1, *weighted EF1 allocation* always exists for goods [56], i.e., in presence of asymmetric agents. Now in order to eliminate envy, EF1 removes the most valuable item. However, the exciting results of EF1 may probably not apply to the real world; i.e., EF1 can be relatively unfair, but the instance may still admit a fairer allocation.

Example 2.1. *Assume Agastya and Noorie have one Toberlone and 99 Eclairs to divide among themselves. Both prefer Toberlone over Eclairs. The round-robin allocation gives 49 Eclairs, one Toberlone to Noorie, and 50 Eclairs to Agastya. Even though the allocation is EF1, it is unfair to Agastya.*

Therefore, it is desirable to strengthen fairness in many cases. Caragiannis et al. 2019 [55] proposed the concept of EFX, i.e., Envy-free up to any item as defined in Definition 5.1. Since then, a lot of work has been focused on EFX allocation's existence, but the question is yet to be resolved beyond the case of two agents. The existence of EFX is an open question for $n > 3$ for additive valuations and $n > 2$ for general valuations. Researchers consider EFX to be the best fairness notion for indivisible items. In contrast to the weaker notion of EF1, the study of EFX existence remains limited and elusive.

Envy-Freeness up to any item (EFX):

As defined in Definition 5.1, in the case of goods, EFX implies that for any pair of agents i and j, if i envies j, the envy can be eliminated by hypothetically removing any good from j's bundle, i.e., by removing the least valued item for i from j's bundle, i.e., $v_i(A_i) \geq v_i(A_j \setminus \{g\})$, $\forall g \in A_j$. Note that EFX implies EF1.

Theorem 2.2 (Plaut et al. 2020 [117])**.** *When agents have general but identical valuations, a modified Leximin algorithm, i.e., Leximin++, ensures EFX.*

Plaut et al. 2020 [117] showed that when agents have *general but identical valuations*, a modification of the Leximin, i.e, leximin++ solution is EFX. As mentioned in Preliminaries 5.3, in the case of identical valuations, for any pair of agents i and j, $\forall S \subseteq M,\ v_i(S) = v_j(S)$. Leximin selects the allocation that maximizes the minimum individual utility; further, if multiple allocations achieve it, it chooses the allocation that maximizes the second minimum individual utility, and so forth. It is well-known that finding the Leximin solution can take *exponential time*. Plaut modified the Leximin algorithm as follows; Leximin++ selects the allocation that maximizes the minimum individual utility; further, if multiple allocations achieve it, it chooses the allocation that maximizes the size of the minimum utility agent's bundle. Further, if multiple allocations achieve it, it chooses the allocation that maximizes the second minimum individual utility, followed by the cardinality maximum of the second minimum agent's bundle, and so forth.

Algorithm 3 *Leximin++ Algorithm*

1: Set $\forall i, A_i = \emptyset$

2: $A \leftarrow \max_{A' \in \prod_n(M)} \min_{i \in N} v_i(A'_i)$

3: **while** $|A| > 1$ **do**

4: $\quad i \leftarrow \min_{j \in N} v_j(A_j)$

5: $\quad A \leftarrow \max_A |A_i|$

6: $\quad A \leftarrow \max_{A' \in A} \min_{i \in N} v_i(A'_i)$

7: **end whilereturn** Allocation A

Remark 2.1. *However, Leximin++ may not ensure EFX allocation when agents have distinct valuations*

Theorem 2.3 (Plaut et al. 2020 [117])**.** *An EFX allocation exists for two agents with general monotone valuations.*

Using Leximin++ combined with Cut-and-Choose, Plaut et al. 2020 [117] demonstrated that there exists EFX allocation for two agents. As described in algorithm 4, the first agent applies Leximin++ to partition the goods into two bundles (i.e., they compute Leximin++ with two copies of themselves). The second agent picks their favorite bundle. As can be seen, the second agent never envies the first agent. Indeed, according to agent 1, the allocation is EFX regardless of what they receive because the partition was created using Leximin++.

Algorithm 4 *Leximin++ Combined with Cut-and-Choose*

1: Run Algorithm 3 with two copies of first agent

2: $A_2 \leftarrow \max v_2(A_1), v_2(A_2)$

3: $A_1 \leftarrow M \setminus A_2$ **return** Allocation A

Even when two agents have identical submodular valuations, Plaut et al. 2020 [117] showed that finding EFX takes exponential time. On the other hand, the algorithm of Lipton et al. 2004 [101] finds an EF1 allocation in polynomial time for any number of agents with monotone valuations. Thus, EFX is indeed significantly stronger than EF1. In addition, Plaut et al. 2020 [117] proposed an algorithm for EFX allocation in polynomial time, i.e., $\mathcal{O}(mn^3)$ when agents have additive valuations with identical ranking, which relies on envy-cycle elimination [101]. Plaut et al. 2020 [117] showed that this algorithm combined with Cut-and-Choose results in a polynomial-time EFX allocation for two agents with additive valuations. The first agent runs the algorithm with two copies of themselves, and the second agent picks first, just like algorithm 4. Gourvès et al. 2014 [84] also presented a polynomial time algorithm for EFX (called as Near Envy-freeness) when for two agents with additive valuations.

A recent study by Chaudhury et al. 2020 [57] showed that EFX allocation always exists for instances with *three agents with additive valuations*. Their rigorous, constructive proof required an extensive and cumbersome case analysis to prove the existence of EFX that involves *pseudo-polynomial* time complexity. Despite the ongoing efforts, the question of EFX's existence remains unanswered for any valuation system that involves more than three agents.

Theorem 2.4 (Chaudhury et al. 2020 [57])**.** *EFX always exists in the case of three agents with additive valuations.*

EFX is arguably the most compelling fairness notion for indivisible items. However, its existence has not been settled and remains one of the major open questions in the fair division. In an attempt to resolve this problem, a range of impressive solutions has been explored, such as considering approximately EFX or EFX with charity, i.e., we leave some low-value items unallocated or solve the cases when agents have special valuation functions. The authors in [1, 5, 23] studied EFX allocations when agents have restricted valuation functions. Babaioff et al. 2021 [23] presented a polynomial-time algorithm for finding EFX allocations for *submodular dichotomous valuations*. Amanatidis et al. 2021 [5] proposed a polynomial-time algorithm called *"MATCH & FREEZE"* for allocating EFX allocations (they refer to it as EFX_0) for 2-value instances, i.e., $\forall i \in N, k \in M, v_{ik} \in I$ and $|I| = 2$. In addition, they presented a modified round-robin, in which they reverse the order of agents in the last round, giving EFX when agents have a valuation in an interval, i.e., $\forall i \in N, k \in M, v_{ik} \in [x_i, 2x_i]$, where $x_i > 0$.

EFX with charity We search for partial EFX allocations, which do not require all items to be allocated and satisfy EFX. A trivial allocation involves no items assigned to any agent. Therefore, the goal is to determine allocations with some bound on the set of unallocated items. Authors in [53, 39, 59] showed that an EFX allocation always exists if we do not allocate at most $n-1$ goods. This result was improved by [39] to $n-2$ goods. They also proved that for four agents, EFX could be ensured by leaving at most one good. For general valuations, Chaudhury et al. 2020 [57] showed that an EFX allocation *always exists* with a pool of unallocated items C such that no one envies the pool and C has less than n goods. Caragiannis et al. 2019 [53] demonstrated that for goods with additive valuations, there exists a partial EFX allocation where all agents get at least half of their MNW allocation value, thus indicating that unallocated goods are low-valued. In [53], the authors conjectured that, "In particular, we suspect that adding an item to an allocation problem (that provably has an EFX allocation) yields another problem that also has an EFX allocation with at least as high Nash welfare as the initial one." Nevertheless, Chaudhury et al. 2020 [57] disproved the conjecture by displaying an example of a partial EFX allocation with a higher Nash welfare than a complete EFX allocation. In parallel to this line of research, many impressive results show the existence of approximate EFX allocations.

Approximately EFX Plaut et al. 2020 [117] proposed approximate EFX, which implies that for any pair of agents i and j, $v_i(A_i) \geq \alpha \cdot v_i(A_j \setminus \{g\}),\ \forall g \in A_j$, where $\alpha > 0$.

Note that α-EFX and EF1 are not comparable, meaning none of the properties implies the other. Plaut et al. 2020 [117] showed that 1/2-EFX allocations always exist when players have sub-additive valuations using a modified envy-cycle elimination method. Amanatidis et al. 2020 [8] proved the existence of 0.618-*EFX* allocations for additive valuations. For $\epsilon \in (0, 1/2]$, Chaudhury et al. 2021 [58] showed that there is always an $(1-\epsilon)$-EFX allocation with a sub-linear number of unallocated goods and a high Nash welfare. Amanatidis et al. 2021 [5] also proposed α-EFX-value notion.

The envy-freeness concept doesn't always make sense, for instance, in the context of *Public Decision Making*. As explained in the Introduction 1, Agastya and Noorie are supposed to purchase a new refrigerator; Agastya likes Samsung, and Noorie likes LG. Regardless of what kind of refrigerator they buy, the concept of envy is no longer applicable. In the next section, we discuss the results related to proportionality for indivisible goods. We now move to another intuitive fairness notion introduced by Steihaus 1948 [131], *Proportionality*.

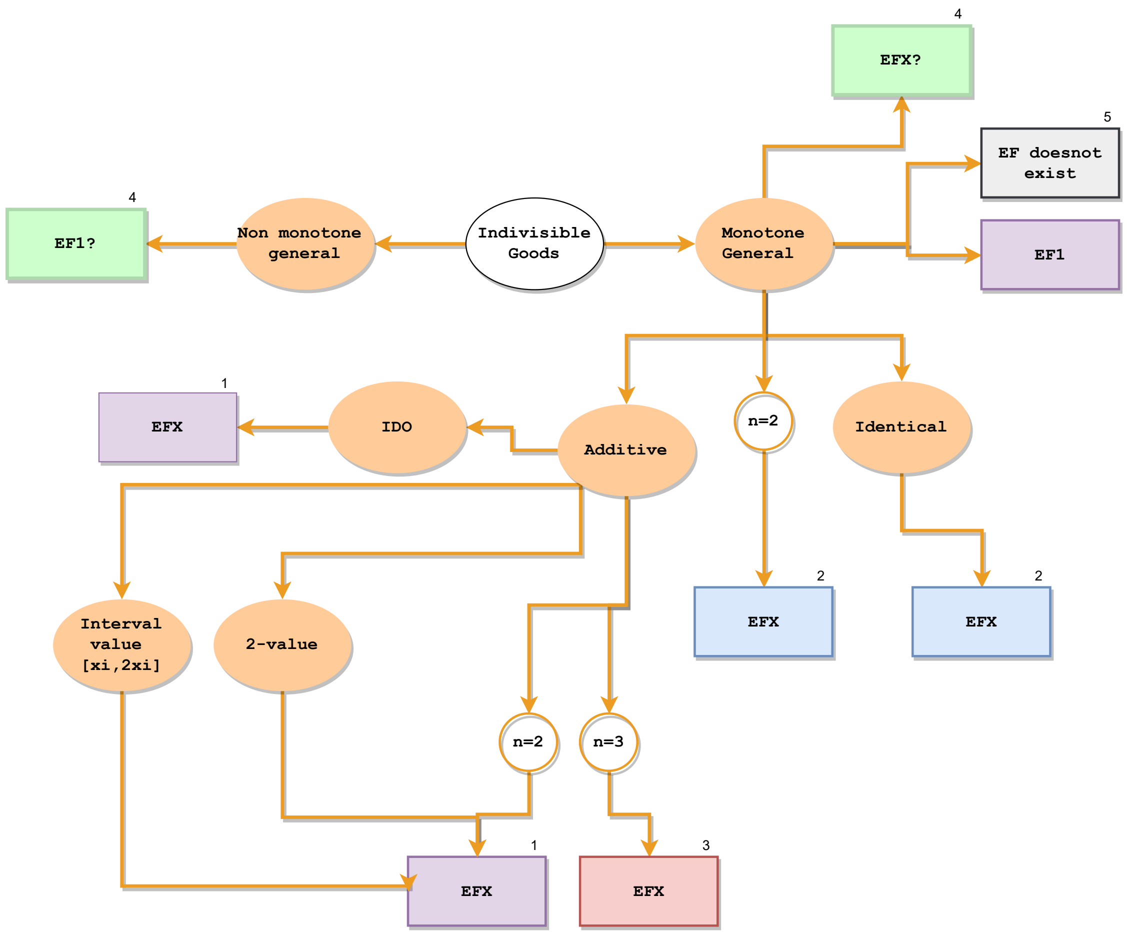

Figure 2.1: Summary of Envy-freeness fairness notion for indivisible goods. The Violet box (number 1) represents the polynomial-time algorithm. The blue box (number 2) represents exponential time. The red box (number 3) represents Pseudo-polynomial time. The green box (number 4) represents an open question. The grey box (number 5) represents such an allocation does not exist.

Proportionality (PROP):

To attain proportionality, each agent would have to receive at least $1/n$ of their value for all items, as stated in Definition 3.2. However, it is known that it is not always possible to assign indivisible items proportionality. The allocation of a single good among agents cannot ever be proportional. Note that EF allocation is always PROP with sub-additive valuation. For $\alpha > 0$, α-proportional allocation may not always exist; for example, in the distribution of single good among two agents, one receives a utility of zero. Naturally, researchers explored the *additive approximations of PROP*, i.e., PROP1, PROPm, and PROPX. In any situation, we look for the fairest allocation possible, and if it is not possible, we relax our criteria in a way that maintains fairness.

Proportionality up to one item (PROP1):

Conitzer et al. 2017 [61] introduced the notion of PROP1 in the setting of Public Decision Making, which is more generic than indivisible item allocations. Along the lines of EF1, PROP1 requires each agent to receive a utility at least their proportional share if we add the largest good allocated to another agent to their bundle, as stated in Definition 3.2. Each agent receives its proportional share after hypothetically including one extra good from another agent's allocation into its bundle.

Under additive valuation, an EF1 allocation is also a PROP1 allocation; therefore, every algorithm that gives EF1 also gives PROP1. In the setting of Public Decision Making, Conitzer et al. 2017 [61] shows that a PROP1 allocation always exists in *polynomial time*. In addition to fairness, Aziz et al. 2022 [14] demonstrated that one could achieve another requirement besides fairness: connectivity. They modeled the situation in which items are placed on a path, and each agent wants a connected bundle of the path. In many scenarios, finding a connected set of items is essential. For example, the items could be a set of rooms along a corridor, and the agents could be research groups seeking adjacent rooms. The authors showed that contiguous proportional allocations of goods exist for additive utilities and that they can be computed in polynomial time. In the later sections, we will discuss how the combination of efficiency and fairness becomes more intriguing in these contexts.

Theorem 2.5 (Aziz et al. 2022 [14]). *Connected PROP1 exists in polynomial time for indivisible goods.*

Proportionality up to any item (PROPX):

PROPX requires each agent to receive a utility at least their proportional share if we add the small good allocated among the remaining agents to their bundle, as stated in Definition 3.2. Each agent receives its proportional share after hypothetically including one extra smallest good from the remaining agents' allocations into its bundle. The PROPX notion is stronger than PROP1. It is also important to note that EFX is weaker than PROPX. While EFX exists for three agents with additive valuations [57], Aziz et al. 2020 [20] presented the following counter-example where PROPX *doesn't exist* with three agents having identical valuations over five goods, i.e., they all value each good 3,3,3,3, and 1 respectively.

In contrast to PROP1, which is easy to satisfy and much weaker, PROPX is much more demanding and is not known to exist even for small instances. Therefore this led researchers to explore alternate ways to relax proportionality.

Proportionality up to the maximin item (PROPm):

The problem is that PROP1 is too weak, and PROPX may not exist; there has to be a fairness concept that is stricter than PROP1 but weaker than PROPX; Baklanov et al. 2021 [24] introduced the notion of PROPm. PROPX requires each agent to receive a utility worth at least their proportional share if we add the largest good among the set of smallest goods allotted to the remaining agents to their bundle, as stated in Definition 1. In other words, PROPm considers the least valued item (from i's perspective) in each of the other agent's bundles and then takes the highest value among them. Note that PROPX implies PROPm implies PROP1.

Baklanov et al. 2021 [24] presented that PROPm exists at least for up to five agents with additive valuations, though the algorithm is exponential in terms of the number of items.

There is a limit of five agents, not because of the rigid limitations of the constructive proof of their algorithm, but because as agents increases, the complexity of analyzing all cases increases. As a result, they conjectured that PROPm will always exist for any number of agents or items. Additionally, Baklanov et al. 2021 [24] explored various relaxations of PROP, such as allowing the value to be added to be equal to a median, mode, or minimax value of the agent. However, their results showed that this fails to exist for as little as three agents. They presented a counter-example with three agents having identical additive valuation over seven items, i.e., all value the items $1 - 6\epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon$ and ϵ, respectively. It is easy to verify that there always exists an agent with utility at most 3. Hence, it violates approximate proportionality with the mean (< 0.25), the median (ϵ), the mode (ϵ), and the minimax item value (ϵ). Later on, Baklanov et al. 2021 [25] demonstrated that PROPm allocations exist in *polynomial time*.

Theorem 2.6 (Baklanov et al. 2021 [25])**.** *PROPm exists, and we can find it in polynomial time in terms of the number of agents and goods.*

The heart of the algorithm lies in breaking the allocation problem into subparts and finding PROPm in each of them, i.e., it recursively solves PROPm by splitting into agent- and item-disjoint sub-problems. This allocation is also PROPm for the original instance. The algorithm starts by selecting an arbitrary divider agent i, and this agent sorts the items in non-decreasing order and partitions m items into n bundles, $(S_1, S_2, \ldots, S_n)$, such that $\forall j \in [n], v_i(S_j) = 1/(n - j + 1) \cdot v_i(M \setminus (\cup_{i=1}^{j-1} S_i))$. The algorithm then evaluates how other agents value these bundles, splits the problem into two, and assigns a bundle to this divider agent. Initially, this is one problem; hence, the proportional Decomposition set D is empty. We set N_d as an empty set representing the agents interested in D, i.e., their valuation of the items in D is proportional. We set N_r as N^{-i} representing the remaining agents. The algorithm iterates for n rounds; in any round t, it computes c as the number of agents in N_r who value the first t bundles $> t/n$. If c is 0 and $|N_d|$ is less than t, it assigns S_t to the divider agent. It further decomposes the agents-items into two sub-problems involving N_d agents composed of $t-1$ agents with D items consisting of the first $t-1$ bundles and N_r agents composed of $n-t$ agents with the last $n-t$ bundles. If $c>0$, there is an over-demand of the initial set of bundles, so the algorithm calls a subroutine UpdateDecomposition. The subroutine uses graph analysis (novel graph representation) to update D by either reducing c by one or increasing the agents in N_d by one.

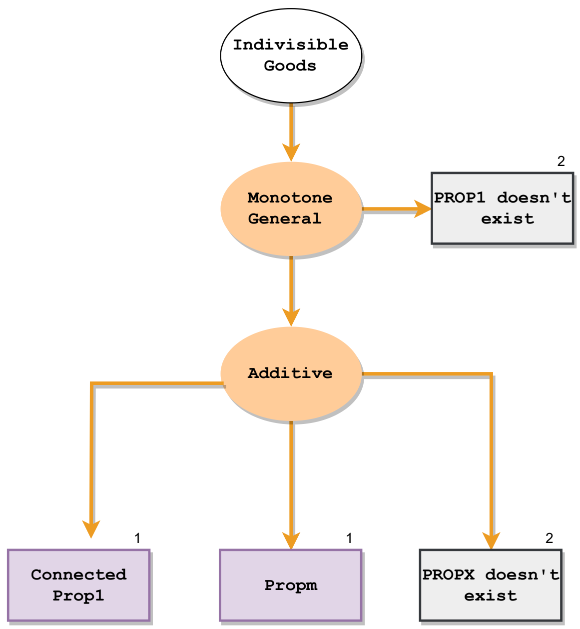

Figure 2.2: Summary of Proportionality fairness notion for indivisible goods. The Violet box (number 1) represents the polynomial-time algorithm. The grey box (number 2) represents such an allocation does not exist.

Additionally, an alternate relaxation that has received considerable attention is Budish 2011 [51]'s maximin share (MMS). We study MMS allocation in detail in the following section.

Maximin Share (MMS):

In essence, MMS is an extension of the well-known *"Cut-and-Choose"* algorithm used in solving the cake-cutting problem. Let's say we ask an agent to divide m items into n bundles and take the bundle that they're least interested in. This risk-averse agent would divide the bundles to maximize the minimum utility, which is the MMS share of the agent. An MMS allocation guarantees every agent their MMS share. Note that MMS is a weaker fairness property than proportionality; a proportional allocation is always MMS. Even though MMS is weaker, MMS is still too demanding in the case of indivisible items.

Despite its appealing formulation, MMS does have a computational disadvantage. For one, even the computation of the maximin share for an agent with additive valuations is an *NP-Complete* problem [45]. And computing an MMS allocation is strongly *NP-Hard*. The problem is weakly NP-hard even for two agents [21, 45]. However, a PTAS for computing MMS exists [141]. Originally, Woeginger 1997 [141] gave PTAS to compute a maximum partition for a particular agent within the context of job scheduling. However, the problem is identical to computing a maximin partition for the given agent.

Bouveret et al. 2016 [45] addressed the question of whether MMS allocation exists for indivisible items. While they answered the question for special cases, they left the question of MMS allocation for additive valuations open. Their results show that when agents have additive valuations, MMS allocations exist in the following cases: (i) *Binary valuations*, (ii) *For two agents*, and (iii) $m \leq n + 3$. When agents have binary valuations, Bouveret et al. 2016 [45] showed that an MMS allocation exists in polynomial time and can be found by employing a round-robin algorithm with an alternating sequence, whereby each agent iteratively selects their most favorite item from the remaining ones. By using polynomial-time case analysis, Bouveret et al. 2016 [45] demonstrated that there exists MMS allocation for instances having n agents and $n+3$ goods. An MMS allocation exists in instances with two agents with additive valuations using the "Cut-and-Choose" protocol. Bouveret et al. 2016 [45] showed that it is challenging to find an MMS allocation when agents have similar preferences, i.e., the more conflict, the more difficult it is to find an MMS allocation. The authors proved that if one can find an MMS allocation for an identical preference instance, every permutation derived from that preference instance can also be allocated as an MMS allocation.

Bouveret et al. 2016 [45] showed that an MMS allocation *need not exist* when agents have general valuations, even in the case of two agents, and left the existence of MMS allocation for additive valuations as an open question. In the case of additive valuations, despite the experimental results of [45], which had MMS allocations consistently, Procaccia et al. 2014 [118] presented an intricate example in which every allocation fails to achieve MMS guarantees. Thus, MMS is not guaranteed to exist in instances with more than two agents with additive valuations. The authors of [70, 97, 118] provided counter-examples to demonstrate that the MMS allocation for goods may not always exist, even with additive valuations. Kurokawa et al. 2016 [97] provided an example involving m that is linear in n, while Procaccia et al. 2014 [118] provided an example involving m that is exponential in n.

Theorem 2.7. *Given a resource allocation problem $(N, M, \mathcal{V})$, there exists M and (additive) valuations such that no allocation is MMS for any $n \geq 3$,*

Because MMS need not exist, naturally, we seek relaxation in the allocation. The goal is to construct an efficient algorithm to provide approximate MMS guarantees. Fortunately, a *multiplicative approximation* of MMS always exists. We discuss α-MMS in the following section.

α-Maximin Share (MMS):

Procaccia et al. 2014 [118] introduced a new concept of approximate maximin share (α-MMS) to solve this problem. An α-MMS allocation guarantees every agent at least the α fraction of the MMS share. Note that all the results mentioned below are in the presence of additive valuations for MMS allocation.

The papers [7, 30, 118] showed that 2/3-MMS for goods always exists. Paper [77, 79] showed that 3/4-MMS for goods always exists. Authors in [77] provide an algorithm that guarantees $3/4 + 1/12n$-MMS for goods.

Procaccia et al. 2014 [118] proved that the 2/3-MMS allocation for goods always exists and devised an exponential-time algorithm. The algorithm enables one of the agents to divide the items as its maximum partition and creates a bipartite graph (between agents and bundles so that edges exist if a bundle is acceptable to the agent). Using Hall's Theorem, they showed that such a graph has a perfect matching, and the algorithm runs recursively on the remaining agents and items. If n is constant, the (PTAS) algorithm runs in polynomial time. They also showed that in the case of three and four agents, 3/4-MMS allocation always exists using the same algorithm.

Amanatidis et al. 2017 [7] presented a PTAS algorithm that computes $2/3 - \epsilon$-MMS allocation, polynomial for any number of agents and goods. They redesigned the algorithm presented in [118] by manipulating matchings of the bipartite graph. Further, they proved that in the case of three agents, there always exists 7/8-MMS allocation and $7/8 - \epsilon$ in polynomial time using the PTAS algorithm. Lastly, when the valuation of every agent for each item, i.e., $\forall i, k, v_{ik} \in \{0, 1, 2\}$, there always exists MMS allocation and presented a polynomial-time algorithm, a variant of a round-robin algorithm. They also provided a much simpler and faster algorithm that computes at least 1/2-MMS, based on a round-robin algorithm with a slight modification of allocating the most valuable goods first. Barman et al. 2020 [30] developed an algorithm that achieves the same level of approximation guarantee as these results. Garg et al. 2019 [77] presented a simple polynomial-time algorithm for 2/3-MMS allocation with straightforward analysis.

Later, Ghodsi et al. 2018 [79] proved that 3/4-MMS always exists and gave a PTAS algorithm for allocating $3/4-\epsilon$-MMS in polynomial time. This result is exciting because Procaccia et al. 2014 [118] proved that a 2/3-MMS allocation was tight, which means using their algorithm would not guarantee better performance. Furthermore, Ghodsi et al. 2018 [79] demonstrated that the 4/5-MMS allocation always exists if there are four agents as well as $4/5-\epsilon$ in polynomial time using the PTAS algorithm. Then Garg et al. 2021 [78] presented a polynomial-time algorithm for finding the 3/4-MMS allocation of goods. The most recent results of Garg et al. 2021 [78]'s work indicates that $3/4 + 1/12n$ MMS always exists.

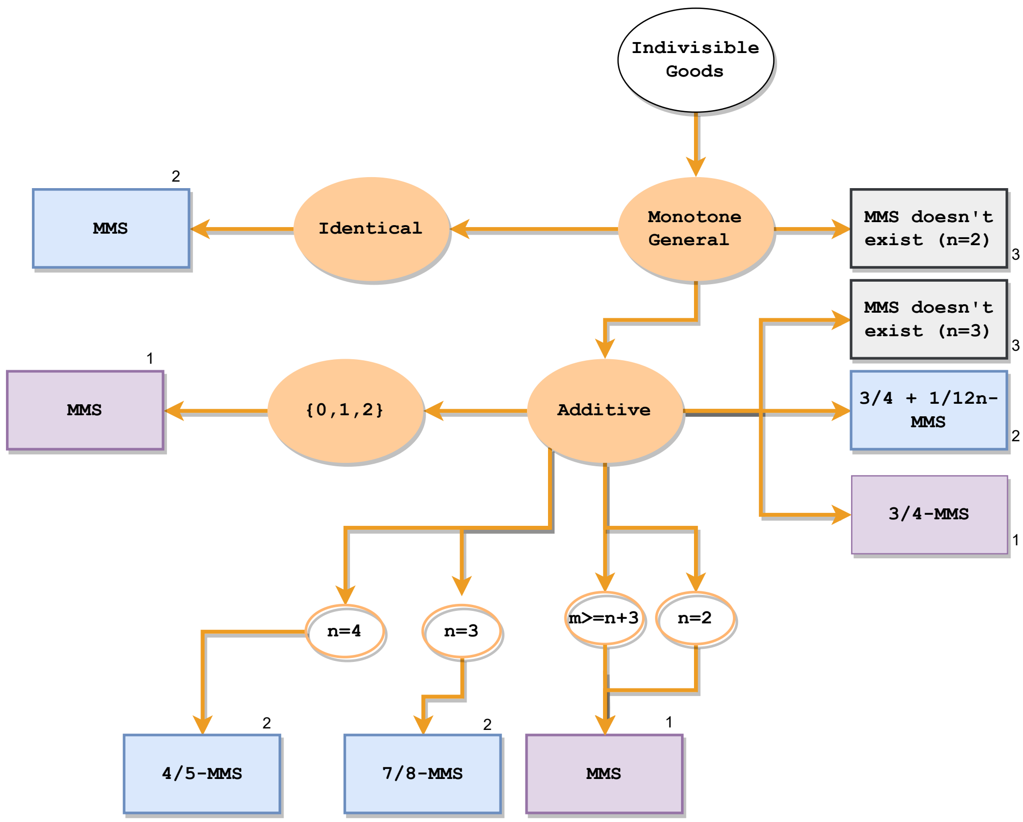

Figure 2.3: Summary of MMS fairness notion for indivisible goods. The Violet box (number 1) represents the polynomial-time algorithm. The blue box (number 2) represents exponential time. The grey box (number 3) represents such an allocation does not exist.

We describe the latest algorithm in detail, i.e., the polynomial-time algorithm for *3/4-MMS* allocation. Except for that, we simply skim over other algorithms briefly. We now talk about two sub-routines.

Normalized Valuations α-MMS is scale invariant, i.e., if A is α-MMS allocation for the valuation profile $\mathcal{V}$, and if we consider a scaled valuation profile $\mathcal{V}'$, i.e., $\forall i \in N, k \in M, v'_{ik} = c \cdot v_{ik}$, A is α-MMS for $\mathcal{V}'$. For simplicity, we scale the valuations such that $\forall i, \mu_i \leq 1$, i.e., $v_i(M) = |N|$.

Bag Filling. Ghodsi et al. 2018 [79] introduced bag filling to allocate low-value goods as an extension of the Moving Knife Procedure (allocates divisible goods proportionally). The algorithm begins by filling an empty bag with arbitrary goods until it satisfies some agent, that is until the agent calls for the bag; after that, the algorithm assigns the bundle to the agent and starts refilling an empty bag again, and repeats the process with the remaining agents. When agents value each item less than $\delta\mu_i$, i.e., $v_{ik} \leq \delta\mu_i$, the bag filling algorithm gives $(1-\delta)\mu_i$-MMS allocation. The intention is that after an agent receives a bundle, enough goods are left for the rest of the agents to meet their $(1-\delta)\mu_i$-MMS allocations. When assigning a bundle to an agent, the other agents value it less than $(1-\delta)\mu_i$ (since they didn't call for the bundle), so the valuation of the rest of the agents for remaining goods is at least $\frac{n-1}{n} \cdot v_i(M)$, which is proportionally more than before assigning the bundle. We can say that the agents who pick bundles early leave more goods than they take. For normalized valuations (i.e. $\forall i \in N$, $v_i(M) = n$), if $\forall i$ and $\forall k$, $v_{ik} \leq \delta$, the bag filling guarantees $(1-\delta)$-MMS allocation, where $0 \leq \delta \leq 1/2$.

Algorithm 5 *Bag Filling Algorithm*

```
Normalize valuations
Set ∀i, A_i = ∅, B = ∅
for k ← 1 to m do
    if ∃i ∈ N, v_i(B ∪ k) > (1 − δ) then
        A_i ← B
        B ← ∅
    end if
    B ← B ∪ k
end forreturn Allocation A
```

Valid Reduction. If we remove an agent i and an item k from an instance $\mathcal{I} = \langle M, N, \mathcal{V} \rangle$, it doesn't reduce MMS guarantees for the remaining agents. We mean to say that in $\mathcal{I}' = \langle M \setminus \{i\}, N \setminus \{k\}, \mathcal{V} \rangle$, $\mu_i' \geq \mu_i$, i.e., the MMS value in the new reduced instance is at least the MMS value in the original instance. For high valued items, i.e., $\exists i, k, v_{ik} \leq \alpha\mu_i$, we assign item k to agent i. After assigning this high-valued item, we still have enough goods, i.e., the MMS value of each agent in the reduced instance is at least the MMS value in the original instance. Our goal is to assign alpha-MMS, i.e., the idea is to assign the high-value items firstly, i.e., giving an agent an item if they value more than alpha mu. We reach the stage where the instance is irreducible, i.e., none of the remaining agents have any remaining high-value items. We say an instance is α-irreducible, if there $\nexists i \in N$, $k \nexists M$, such that $v_{ik} \geq \alpha\mu_i$.

The core concept lies in using these two algorithms to ensure α-MMS, i.e., firstly assign high-valued items using valid reduction and further assign low-valued and medium-valued items using a modified bag filling.

A polynomial-time 1/2-MMS We start by normalizing the valuations for all the agents. We assign an item to an agent if they value it more than 1/2. We break ties arbitrarily. Note that when we assign an item to an agent, in the reduced instance, the new MMS value of all the remaining agents is at least as much as in the original instance. Once the instance becomes 1/2-irreducible, i.e., if there are remaining agents, they all value the remaining items less than 1/2, we assign the rest of the goods using the bag filling algorithm.

Ordered Instances Further, there's another core concept used by these algorithms, which was first noticed by Bouveret et al. 2016 [45]. They showed that it is challenging to find an MMS allocation when agents have similar preferences, i.e., the more conflict, the more difficult it is to find an MMS allocation. The authors proved that if one can find an MMS allocation for an identical preference instance, i.e., an ordered instance, every permutation derived from that preference instance can also be allocated as an MMS allocation. Formally, we say an instance is an ordered instance *iff* there exists a total ordering over the set of goods, i.e., $\forall i \in N, v_{i1} \geq v_{i2} \geq \ldots \geq v_{im}$. Further, Bouveret et al., Barman et al. 2016, 2020 [45, 30] provided a reduction from any arbitrary instance $I = (N, M, \mathcal{V})$ to an ordered instance I', and showed that if A' satisfies α-MMS guarantees for I', we can find allocation A for I in polynomial time derived from A' in polynomial-time.

Given an instance $I = (N, M, \mathcal{V})$, we construct an ordered instance $I' = (N, M, \mathcal{V}')$ as follows,

Algorithm 6 *Converting into Ordered Instance*

```
for k ← 1 to m do
    for i ← 1 to n do
        k* ← agent i's k^th most valuable good
        v'_ik ← v_ik*
    end for
end forreturn Valuation V'
```

Now given an allocation A' which satisfies α-MMS for I', we construct A which satisfies the same for I as follows,

Algorithm 7 *α-MMS for Original Instance I*

```
A = (∅, ∅, ..., ∅), and R ← M
for k ← 1 to m do
    Pick the agent i who has item k in A'
    g ← argmax_{k∈R} v_ik
    A_i ← A_i ∪ {g} and R ← R \ {g}
end forreturn Valuation V'
```

It is easy to verify that this whole procedure is a polynomial-time algorithm. From now, we only need to address the setting of ordered instances, i.e., identical preferences, and using a polynomial time algorithm, we can compute α-MMS for any arbitrary additive instances. Observe that the MMS values of an agent in I and I' are the same because it neither depends on the order of the items nor on other agents' valuations.

3/4 MMS when we know the μ values

We now discuss the algorithm for computing 3/4-MMS allocation. We pre-process our instances such that they are normalized and ordered. We assume that we know the MMS value of each agent, and we scale the valuation such that the MMS value becomes 1 for all. Note that this is a crucial assumption; later, for the strongly polynomial time algorithm for exact 3/4, we will change the way we normalize the valuations.

For ease of exposition, we abuse notation and use M and N to denote the set of unallocated items and the set of agents who have not yet received any bundle. The basic structure of the algorithm closely resembles that of valid reduction and bag filling.

The algorithm first assigns high-valued items. In the algorithms, they make it a simple process of greedy assignment by leveraging the pigeonhole principle to make valid reductions. They define bundles $S_1 := \{1\}$, $S_2 := \{n, n+1\}(\emptyset, \mathit{if} m \leq n)$, $S_3 := \{2n-1, 2n, 2n+1\}(\emptyset, \mathit{if} m \leq 2n)$, and $S_4 := \{1, 2n+1\}$. S_1 has the highest valued item in M, S_2 has the n^{th} and $(n+1)^{th}$ highest valued items in M, and so on. If any of these bundles value 3/4 to an agent, then it is a valid reduction. We assign the lowest index bundle in $\{S_1, S_2, S_3, S_4\}$ to agent i such that it is valued at least 3/4. Note that after each valid reduction, we update M and N, and these bundles will change accordingly.

Once the instance becomes α-irreducible, i.e., for the remaining agents $i \in N$, $S \in \{S_1, S_2, S_3, S_4\}$, and $\forall i, S, v_i(S) < 3/4$, we proceed to the modified bag filling algorithm for the rest of the agents. Let $J_1 := \{1, 2, \ldots, n\}$ and $J_2 := \{n+1, \ldots, 2n\}$. Next, we initialize n bags as follows: $B = \{B_1, B_2, \ldots, B_n\}$; where $B_k = \{k, 2n-k+1\}, \forall k$. Each bag contains one item from J_1 and one item from J_2 such that from B_1 to B_n value of items from J_1 decreases and the value of items from J_2 increases. Now in each round k, it starts a new bundle T with $T \leftarrow B_k$. If there is an agent who values T to be at least 3/4, then assign T to such an agent. Otherwise, keep adding goods from $M \setminus \{J_1 \cup J_2\}$ to T one by one until an agent with no bundle assigned to her values T at least 3/4. The algorithm allocates T to that agent, and if there are multiple such agents, it chooses one arbitrarily. The complete details are given in the Algorithm 8.

However, note that this algorithm requires a detail of the exact MMS value of all the agents, which is NP-Hard to compute. The authors further modified the algorithm in such a that it is strongly polynomial in time and computes exact 3/4-MMS allocation with the same underlying concepts. The key modification is the way we normalize valuation, allocation of S_4 bundle, and use the average as an upper bound for the MMS value.

Algorithm 8 *Exact 3/4-MMS Algorithm*

Assume MMS values are known, alpha,

Normalize the Valuations such that $\forall i, \mu_i = 1$

Using Modified Valid Reduction now

$S_1 \leftarrow \{1\}$, $S_2 \leftarrow \{n, n+1\}, S_3 \leftarrow \{2n-1, 2n, 2n+1\}, S_4 \leftarrow \{1, 2n+1\}$

$\forall S \subseteq M$, we define $\tau(S) := \{i \in N : v_i(S) \geq \alpha\}$

while $(\tau(S_1) \cup \tau(S_2) \cup \tau(S_3) \cup \tau(S_4))) \neq \emptyset$ **do**

    $S \leftarrow$ lowest index in $\{S_1, S_2, S_3, S_4\}$ which satisfies α

    agent i in $\tau(S)$

    $A_i \leftarrow S$

    $M \leftarrow M \setminus S$; $N \leftarrow N \setminus \{i\}$

end while

Initialize bags B, where $B_k = \{k, 2n-k+1\}, \forall k$

$R \leftarrow M \setminus J$

for $k \leftarrow 1$ *to* n **do**

    $T \leftarrow B_k$

    while $v_i(T) < \alpha$ **do**

        $g \in R$

        $T \leftarrow T \cup \{j\}; R \leftarrow R \setminus \{j\}$

    end while

    $\exists i$, $v_i(T) \geq \alpha$, $A_i \leftarrow T$, $N \leftarrow N \setminus i$

end forreturn Valuation $\mathcal{V}'$

Garg et al. 2021 [78] proved that existence of $\frac{3}{4} + \frac{1}{12n}$ using Algorithm 8. They showed that this bound is tight. They showed that extending this approach for improved α would be challenging.

Apart from additive valuations for indivisible goods, MMS allocations have been studied extensively in various other settings. Farhadi et al. 2019 [69] studied MMS allocation for asymmetric agents. Seddighin et al. 2019 [124] explored MMS allocation in the presence of externalities. Amanatidis et al., Amanatidis et al., Aziz et al., Barman et al. 2017, 2016, 2019, 2019 [4, 6, 18, 29] studied MMS allocation for strategic agents. Barman et al., Ghodsi et al., Li et al., Gourvès et al., Biswas et al. 2020, 2018, 2021, 2019, 2018 [30, 79, 100, 83, 41] studied MMS allocation beyond additive valuations.

Next, we discuss fair and efficient algorithms for indivisible goods.

2.2.2 Fair and Efficient Allocation

For various reasons, we strive to seek fair, effective allocations. We want to ensure fairness, but we also want to ensure the items go to the most deserving; however, we encounter various difficulties when doing so. Within the PROP1 allocation set, finding an allocation that is utilitarian-maximum is NP-hard [17]. The problem of making Pareto improvements to PROP1 allocation is computationally challenging [12, 93]. It is possible that a Pareto improvement over a PROP1 allocation will not even satisfy PROP1. In this section, we summarize the literature on efficient fair allocation for indivisible goods.

Efficient EF1 allocations

Our aim in this section is to discuss the results regarding efficient allocation of EF1.

EF1 + PO Recall that an allocation is said to be Pareto optimal (PO) if it is not Pareto dominated by any other allocation, i.e., there is no other allocation in which every agent is better off and at least one agent is strictly better off. Under additive valuations, Caragiannis et al. [55] has shown that EF1 and PO allocations always exist. As shown in their study, the MNW allocation (i.e., an allocation that maximizes Nash social welfare) satisfies both EF1 and PO. This is challenging because the complexity of computing an allocation like this is high, i.e., even though finding EF1 is polynomial time, even for additive valuations, polynomial-time algorithms are not available to find EF1 and PO. Computing MNW allocation is APX-hard [98]. In addition,Barman, Krishnamurthy, and Vaish [32] proposed a *pseudo-polynomial time* algorithm for determining PO and EF1 allocations.

EF1 + USW Among a set of utilitarian maximal allocations, Aziz et al. [16] proved that it is strongly NP-hard to find an allocation that is PROP1. Furthermore, they showed that maximizing the utilitarian welfare among a set of fair allocations when the number of agents is variable is *strongly NP-Hard*, while it is NP-Hard when the number of agents is fixed and greater than two. There is a polynomial-time algorithm that determines whether an EF1 allocation exists or not among the utilitarian maximal allocations when there are two agents. Further, they designed *pseudo-polynomial time* algorithms for both problems with a fixed number of agents.

In light of these open questions in finding efficient EF1 allocations, researchers have explored efficient PROP1 allocations and obtained positive results.

Efficient PROP1 allocations

PROP1 + PO EF1 and PO are satisfied by MNW allocation, which implies that when agents have additive valuations, an allocation that meets PROP1 and PO exists. [61]. In public decision settings (a setting that is more general than indivisible goods allocation), Conitzer, Freeman, and Shah[61] proved that there is always a PROP1 and PO allocation. They left the complexity of computing such an allocation as an open question. For goods with additive valuations, Barman and Krishnamurthy[31] proposed a *strongly polynomial-time* algorithm for calculating PROP1 and fPO (stronger than PO) by leveraging *pure market equilibrium*. In contrast to PO, fPO requires that an allocation is not dominated by fractional (or integral) allocations; fPO is, therefore, a stronger solution concept. Because finding Pareto improvements while maintaining PROP1 is challenging, they deal with fractional allocations that are improving and convert them into integral allocations. They do so by leveraging fisher markets, i.e., such markets consist of a set of divisible goods, a set of agents with budgets and valuations over these goods. Fisher markets clear when goods are assigned prices; each agent spends its entire budget on only those most valuable goods per unit spent, and the market is in equilibrium. From the resource allocation perspective, market equilibrium is applicable since the first welfare theorem states that these equilibria are always Pareto efficient [107]. In Fisher markets with additive valuations, Eisenberg and Gale's convex program provides an efficient way to determine equilibrium allocations and prices: the primal and dual solutions of the convex program correspond to the equilibrium allocations and prices [67, 122]. Strongly polynomial-time algorithms exist for the same [113, 136]. The equilibrium allocation is fractionally Pareto efficient (fPO). However, equilibrium allocations are not necessarily integral, i.e., the allocation we get may not be applicable in the context of indivisible goods. Note that (in contrast to computing an arbitrary equilibrium) finding an integral equilibrium is computationally hard, i.e., determining whether a given Fisher market is pure is an NP-hard problem [31]. In general, agents are assigned fractional shares of goods in a market equilibrium. In other words, we cannot directly incorporate indivisible goods into the market framework.

The authors developed an efficient algorithm that first computes an equilibrium (A, p) of the market B and, then, round (fractional) allocation A to an integral allocation A' such that it is an integral equilibrium of the market B'. The algorithm doesn't change the price of the goods. It defines the new budgets (a bounded change in agents' budgets) to explicitly satisfy budget exhaustion with respect to the computed allocations and the unchanged prices at the end. Since A' is an equilibrium of the Fisher market B', via the first welfare theorem, A' is fPO and PROP1. This conversion to A' is in strongly polynomial time. Briefly speaking, given a market and its equilibrium (A, p), the spending graph $G(A, p)$ is a forest. The algorithm starts by identifying the root of each tree at some agent in this forest. Then assign child goods to root agent i (with no parents) until adding more child goods to i violates the budget constraint. The remaining child goods are then appropriately assigned to grandchildren agents. After each distribution, delete this parent agent i and all its allocated child goods. Iteratively assign remaining goods to agents until all the goods are allocated. Note that if a good is completely assigned to an agent, i.e., it is integrally assigned, then it will continue to be assigned to the same agent after conversion. This algorithm gives weighted-PROP1, which was later mentioned in [48]. This algorithm satisfies EF1[1], weighted-PROP1, and fPO.

PROP1 + USW Among a set of utilitarian maximal allocations, Aziz et al. [16] proved that it is *strongly NP-hard* to find an allocation that is PROP1. Furthermore, they showed that maximizing the utilitarian welfare among a set of fair allocations when the number of agents is variable is strongly NP-Hard, while it is NP-Hard when the number of agents is fixed and greater than two. There is a polynomial-time algorithm that determines whether a PROP1 allocation exists or not among the utilitarian maximal allocations when there are two agents. Further, they designed pseudo-polynomial time algorithms for both problems with a fixed number of agents.

PROPm + PO/USW Towards finding efficient PROPm allocations, this area of study remains unexplored, leaving the scope for future research.

Efficient EFX allocations

EFX + PO In the presence of zero marginal valuations, Plaut and Roughgarden [117] showed that there are additive valuations in which no EFX allocation is also PO. In the presence of zero marginal valuations, Plaut and Roughgarden [117] showed that there are general and identical valuations in which no EFX allocation is also PO. In the presence of nonzero marginal valuations, for any number of agents with general but identical valuations, and for two agents with (possibly distinct) additive valuations, the leximin++ solution is

EFX and PO. They also argued that the assumption of nonzero marginal utility is quite reasonable with two agents having additive valuations as one agent is indifferent to some good; we can assign the good to the other agent and exclude it from the division process completely.

Towards finding efficient, fair allocations, especially for general valuations, this area of study remains unexplored, leaving the scope for future research. Moreover, it would be interesting to explore efficient allocation when a fair allocation, i.e., if MMS allocation exists, what are the efficiency guarantees we can achieve? We summarize our results about the efficient, fair allocation of goods in Figure 2.4. Next, we move on to the discussion of chores.

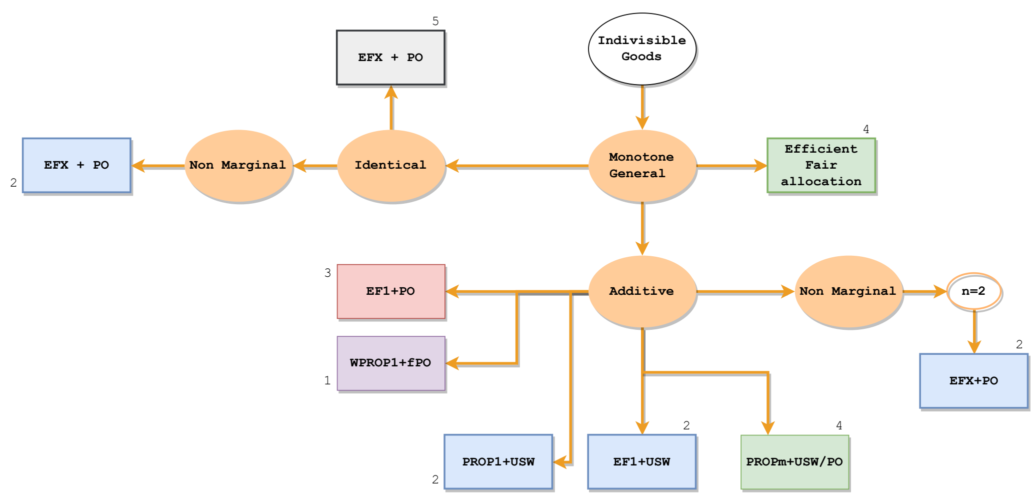

Figure 2.4: Summary of efficient fairness allocation for indivisible goods. The Violet box (number 1) represents the polynomial-time algorithm. The blue box (number 2) represents exponential time. The grey box (number 5) represents such an allocation does not exist. The red box (number 3) represents pseudo-polynomial time. The green box (number 4) represents open questions.

2.3 Indivisible Chores

2.3.1 Fair Allocation

Envy-Freeness:

Similar to indivisible goods, determining the existence of an envy-free allocation is NP-complete, even in the simple case when agents have binary additive valuations for chores [40].

Envy-Freeness up to one item (EF1):

Similarly to goods, a simple round robin gives an EF1 for chores in *polynomial time* – $\mathrm{O}(mn \log m)$. The chore division problem might appear to be the 'opposite' of the goods problem, and therefore intuitively, one might expect the natural adaptation of algorithms designed to compute an EF1 allocation for goods, to also work for chores. This, however, is not the case. The envy-cycle elimination algorithm [101] doesn't give EF1 allocation in the case of chores. Bhaskar, Sricharan, and Vaish [40] presented *Top-trading envy-cycle elimination algorithm*, a modified Envy-cycle algorithm for chores that gives EF1 for indivisible chores with anti-monotonic valuations. For indivisible goods, resolving arbitrary envy cycles (until the envy graph becomes acyclic) is known to preserve EF1. However, for indivisible chores, the choice of which envy cycle is resolved matters. EF1 for chores entails that any pairwise envy can be addressed by removing some chores from the envious agent's bundle. The algorithm works by assigning, at each step, an unassigned item with the largest cost to an agent who does not envy anyone else (i.e., a non-envious agent who is a "sink" node in the top-trading envy graph). If the top-trading envy graph does not have a sink, then it must have a cycle. Then resolving the top-trading envy cycles guarantees the existence of a sink agent in the top-trading envy graph.

Now we briefly describe the algorithm for EF1 for chores as follows. In the above algorithm, given a partial allocation A, we consider a subgraph of the envy-graph G_A that we call the top-trading envy graph T_A whose vertices denote the agents, and an edge (i, k) denotes that agent i's (weakly) most preferred bundle is A_k. If the envy graph does not have a sink, then the top-trading envy graph T_A has a cycle. Therefore, resolving top-trading envy cycles guarantees the existence of a sink agent in the envy graph, and it also preserves EF1. This is because every agent involved in the top-trading exchange receives their most preferred bundle after the swap, which means they do not envy anyone else in the next round.

Envy-Freeness up to any item (EFX):

An EFX allocation also exists for IDO instances for chores [99]. As described in [40], the Top Trading Envy Cycle algorithm allocates items in any order, and it meets the EF1 requirement. Modifying this algorithm to allocate the largest chores (minimum valued items) to sink agents guarantees EFX allocation for IDO instances. We can ensure that EFX allocation is made for IDO instances if we select the item with the highest cost at each step. This returns an EFX allocation in polynomial time for additive valuations for any IDO instance. For two agents or any agents with IDO valuation functions, EFX allocations always exist.

Approximate EFX allocations While good results have been obtained for the (approximation of) EFX allocations for goods, little is known about how EFX is allocated (approximately). It is not known even if constant approximations of EFX allocations exist for three agents with general additive valuations, let alone whether EFX allocations exist at all. For indivisible chores, the authors propose algorithms to compute approximate EFX allocations in polynomial time in [142]. When there are three agents, the algorithm produces 5-EFX, and when there are four or more agents, it produces $3n^2$, i.e.,$\mathcal{O}(n^2)$ as the approximation ratio. The authors observed that in doing so, i.e., the algorithm in order to allocate approximate EFX often would increase the cost of other agents instead of decreasing the cost of an envied agent, which can lead to inefficient allocation. Another interesting open question in approximate-EFX is, can we get a constant approximate ratio for any number of agents? Apart from approximate-EFX, in the case of goods, EFX with charity is well explored as described in Section, However, it is unknown whether similar results hold for the allocation of chores.

Algorithm 9 EF1 for Chores with General Valuations

1: Initialize $A \leftarrow (\emptyset, \emptyset, \ldots, \emptyset)$
2: **for** $k \leftarrow 1$ *to* m **do**
3:     **if** there is no sink agent in envy graph G_A **then**
4:         $C \leftarrow$ any cycle in T_A
5:         $A \leftarrow A^C$
6:     **end if**
7:     Chose a sink j in the envy graph G_A
8:     $A_j \leftarrow A_j \cup k$
9: **end forreturn** Allocation A

Proportionality (PROP):

Similar to indivisible goods, there may not be a proportional allocation for indivisible chores. Determining if a proportional allocation is possible, even for just two agents, is a problem that is NP-Complete [45].

Proportionality up to one item (PROP1):

When valuations are additive, EF1 and PROP1 are equivalent. This means that any algorithm that results in EF1 allocations also satisfies PROP1. According to a study by Aziz et al. ([14]), using the PROP1 concept allows for not only fairness but also connectivity

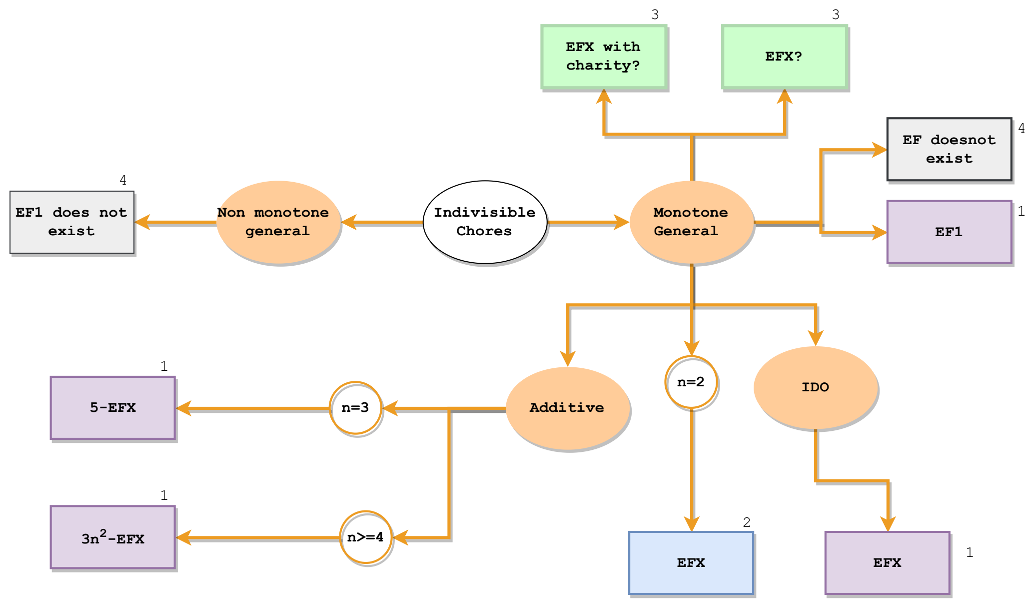

Figure 2.5: Summary of EF fairness notion for indivisible chores. The Violet box (number 1) represents the polynomial-time algorithm. The blue box (number 2) represents exponential time. The grey box (number 4) represents such an allocation does not exist. The green box (number 3) represents an open question.

in the allocation of mixed goods (including chores) in a contiguous and proportional manner and can be computed efficiently.

Proportionality up to any item (PROPX):

It is widely acknowledged that EFX (Envyfreeness up to any item) is stronger to PROPX (Proportionality up to any item) when it comes to the allocation of chores. This stands in contrast to the allocation of goods, where PROPX has been found to be more stronger. In their study, [99] examined the concept of weighted proportionality up to any item and established that a weighted PROPX allocation exists and can be calculated in a computationally efficient manner. Therefore, while PROPX may not be applicable to indivisible goods, it can be calculated efficiently for indivisible chores. The *"Bid-and-Take"* algorithm, a variation of the "Top trading envy cycle algorithm" [40], is presented by [99]. In this algorithm, they choose the largest tasks (items with the lowest value) and assign them to the active agent who has the lowest current social cost for the item. Initially, all agents are active. When an agent's cumulative cost exceeds their proportional share, they are deactivated. It is important to note that in [40], items were allocated in any order and were EF1. However, this modification guarantees that the allocation is EFX (and therefore PROPX) for IDO instances.

The authors proposed using a similar approach to algorithms for approximate MMS, where they reduce the instance to IDO to find the α-MMS and then reconvert it back to the original instance, while still maintaining its α-MMS status. They state that if a polynomial time algorithm exists that can compute an α-WPROPX allocation for any IDO instance, then there also exists a polynomial time algorithm that can compute an α-WPROPX allocation for any additive instance. This modified envy-cycle elimination algorithm ensures both weighted PROPX and 4/3 approximate MMS simultaneously. Additionally, the authors have also found that an EFX and a weighted EF1 allocation for indivisible chores exist if all agents have the same ordinal preference. It should be noted that while all PROPX allocations are PROP1, the reverse is not necessarily true. As demonstrated for indivisible chores, wPROPX allocations can be computed in a *polynomial time*, the emphasis is placed on wPROPX allocations, and any findings apply to PROP1.

Maximin Share Allocation (MMS):

The problem of allocating MMS allocation to chores is closely related to *bin packing problem/job scheduling problem*.

Aziz et al. [21] presented an example based on [118] with three agents with additive valuations where every allocation results in some agents having a lower MMS value than they should. There is no guarantee that an MMS allocation for chores exists, even with additive valuations, as demonstrated in [21, 70]. While it has been shown that MMS

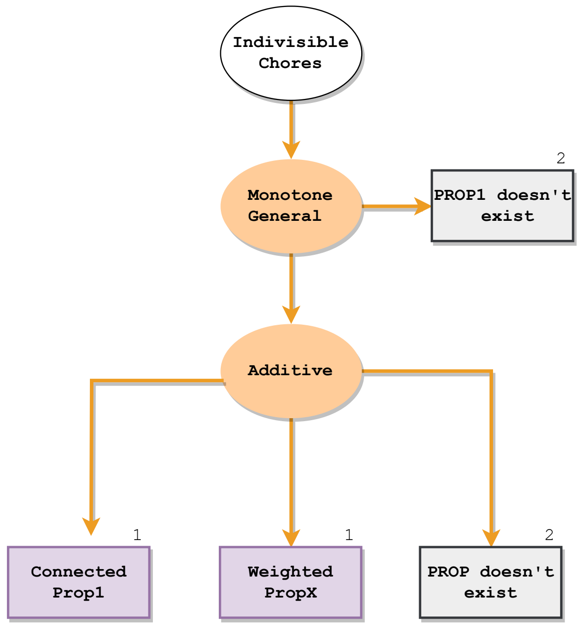

Figure 2.6: Summary of PROP fairness notion for indivisible chores. The Violet box (number 1) represents the polynomial-time algorithm. The grey box (number 2) represents such an allocation does not exist.

allocations are not guaranteed to exist for indivisible chores [21], there are many works that study its approximations [21, 30, 90]. The state-of-the-art approximation ratio for MMS allocation for indivisible chores is 11/9 [90]

Theorem 2.8. *[21] Given a chore allocation problem* $(N, M, \mathcal{V})$*, for any* $n \geq 3$*, there exists* M *and (additive) valuations that do not admit an MMS allocation.*

Approximately MMS Allocation (α-MMS):

Researchers have been working on finding efficient algorithms for approximating the maximum minimum share (MMS) allocation, particularly for situations where MMS does not always exist, such as in the case of chores. One such algorithm for 2-MMS has been demonstrated to be polynomial time by Aziz et al.[21]. This result has been improved upon by Barman and Krishnamurthy, who presented a polynomial time algorithm for 4/3-MMS. In the latest development, Huang and Lu showed that an *11/9-MMS* allocation could be achieved by converting the instances to identical ordinal instances (IDO) and then solving for approximate maximin allocation. The authors proposed a mechanism similar to the *First Fit Decreasing (FFD) algorithm* for bin packing problems to allocate chores. This involves starting with high-valued chores and adding them to the bundle in decreasing order until an agent's threshold is met, and then allocating that bundle to that agent. The algorithm repeats this process for all agents until all chores are allocated. The authors also provided a polynomial time approximation scheme (PTAS) and a polynomial time algorithm for 5/4-MMS allocation.

To compute the 11/9-MMS allocation, the algorithm first converts the instance to an IDO instance and orders the chores in decreasing order. Then, starting with high-valued chores, the algorithm adds chores to the bundle until an agent's threshold is met and allocates the bundle to that agent. This process is repeated for all agents until all chores are allocated. The authors proved that if the threshold value is set to 11/9, then the algorithm terminates with an 11/9-MMS allocation, with all chores being allocated. However, the best approximation achievable using this method is an open question.

In summary, researchers have been exploring ways to approximate the MMS allocation for situations where it does not always exist, such as in the case of chores. The latest development is an algorithm that can achieve an *11/9-MMS* allocation by converting instances to IDO instances and using a mechanism similar to the FFD algorithm for allocating chores. This algorithm has been shown to be polynomial time, and the authors also provided a PTAS and a polynomial time algorithm for *5/4-MMS* allocation. However, the best approximation achievable using this method is still unknown.

Algorithm 10 *Algorithm for* 11/9-*MMS*

Assume IDO instance and threshold values of each agent $(\alpha_1, \alpha_2, \ldots, \alpha_3)$

Assume $\forall i, \mu_i = 1$

for $i \leftarrow 1$ *to* n **do**

  $K \leftarrow \emptyset$

  for $k \leftarrow 1$ *to* m **do**

    if $j \exists N,\ v_j(A_j \cup M_k) \leq \alpha_j$ **then**

      $K \leftarrow K \cup M_k$

    end if

  end for

  $M \leftarrow M \setminus K$

  Let j be the agent who is allocated K, i.e., $v_j(K) \leq \alpha_j$

  $A_j \leftarrow K$ and $N \leftarrow N \setminus j$

end forreturn Allocation A

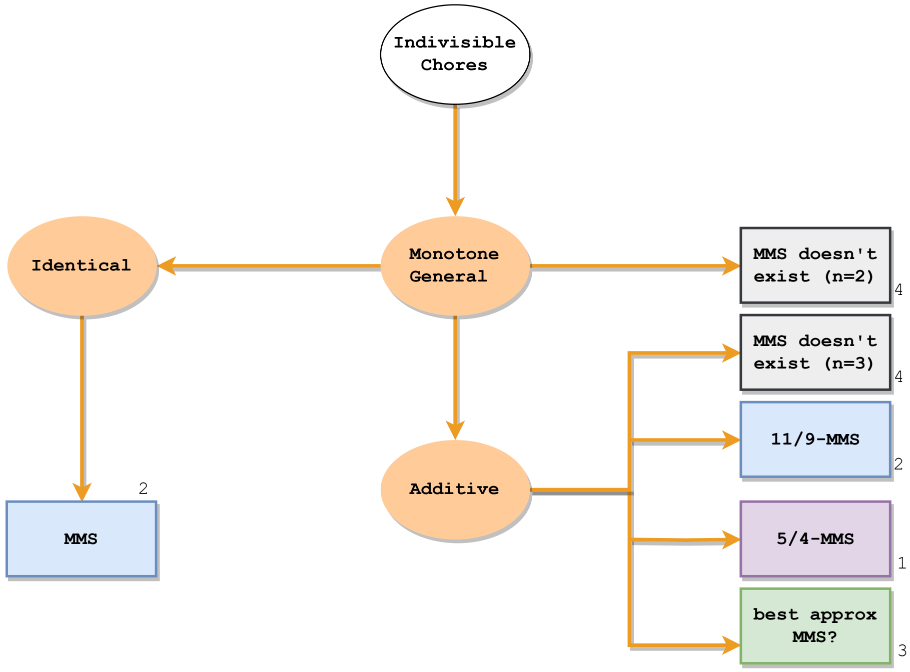

Figure 2.7: Summary of MMS fairness notion for indivisible chores. The Violet box (number 1) represents the polynomial-time algorithm. The blue box (number 2) represents the exponential-time algorithm. The green box (number 3) represents an open question. The grey box (number 4) represents such an allocation does not exist.

2.3.2 Fair and Efficient Allocation

Efficient EF1 allocations

When valuations are negative, the existence and complexity of an allocation that satisfies EF1 (envy-freeness up to one item) and PO (proportional) properties is still an *open problem*. However, alternative fairness criteria, such as *EF_1^1 (envy-freeness up to one item and one penny)* introduced in [48], do exist and can be efficiently computed along with fPO (fractional proportional) allocation. Additionally, the problem of maximizing utilitarian welfare while ensuring EF1 allocation is not fully explored in chore allocation.

In summary, despite the challenge posed by negative valuations, there exist fair allocation criteria that can be efficiently computed, and there is still room for further research in maximizing welfare while satisfying fairness constraints in the context of chore allocation.

Efficient PROP1 allocations

In a recent study, Branzei et al. [48] demonstrated the existence of a *strongly polynomial-time* algorithm for solving the chores problem. This algorithm guarantees a weighted PROP1 and PO allocation, provided the number of agents or items is fixed. The underlying algorithm used in their study is similar to the pure market equilibria approach proposed by Barman et al. [31].

The researchers also established that for any chore division problem, there exists an indivisible allocation A that is Pareto optimal in the divisible problem and satisfies weighted-EF1[1] and weighted-Prop1. Furthermore, they computed a competitive allocation A' and demonstrated that an integral competitive allocation A with a budget satisfying could be constructed from A'. This allocation is Pareto optimal by the first welfare theorem.

In summary, Branzei et al. proposed an efficient algorithm for solving the chores problem and provided theoretical guarantees for the Pareto optimality and efficiency of their proposed allocation. Their findings provide valuable insights into the design of allocation algorithms for resource allocation problems. It is important to note that the existence of PROPX and PO allocation for chore allocation problems remains an *open question*.

We summarize the results discussed in this section in the following table 2.1. The red line represents impossibilities. And brown line represents open questions in the literature. Even though this is a rich literature, there are still a lot of open areas that need to be resolved.

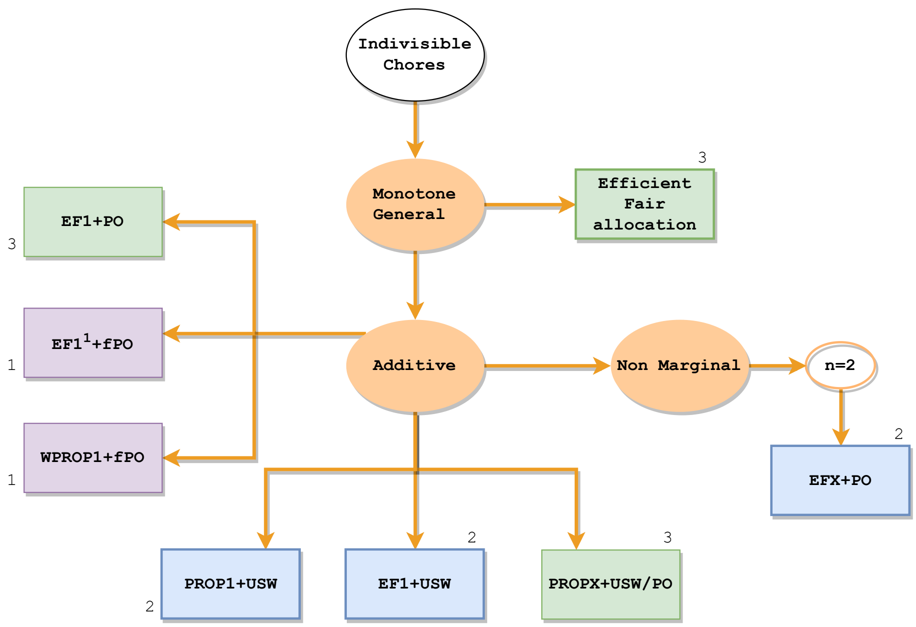

Figure 2.8: Summary of efficient fairness notion for indivisible chores. The Violet box (number 1) represents the polynomial-time algorithm. The blue box (number 2) represents exponential time. The green box (number 3) represents an open question.

Table 2.1: Summary of Fair Division for Indivisible Goods or/and Chores

Fairness	Efficiency	Agents	Items	Valuations	Existence	Computation	Algorithm	Paper
EF	any	any	any	any	no	NP-Hard	-	-
EF1	-	any	goods	additive	yes	polynomial	round robin	Caragiannis et al. [55]
			chores					Aziz et al. [14]
			comb				doubly round robin elimination	
EF1	-	any	goods	general	yes	polynomial	envy cycle elimination	Lipton et al. [101]
			chores				modified envy cycle elimination	Bhaskar, Sricharan, and Vaish [40]
			comb					
EF1	-	any	any	arbitrary	?	?	?	?
EF1	PO	any	any	general	?	?	?	?
			goods		yes	pseudo polynomial	market equilibria	Barman, Krishnamurthy, and Vaish [32]
EF1	PO	any	chores comb	additive	?	?	?	?
EF1	USW	any	any	any	yes	NP-Hard	?	?
EF1	USW	any	good	additive	yes	exponential	dynamic programming	Aziz et al. [16]
EFX	any	any	goods/ chores	general	?	?	?	?
EFX	any	any	goods/ chores	additive	?	?	?	?
EFX	-	any	comb	lexicographic sub domain of additive	no			Hosseini et al. [87]
			goods		yes	exponential	leximin++	Plaut and Roughgarden [117]
EFX	-	any	chores comb	IDO general	?	?	?	?
EFX	-	any	goods	IDO additive	yes	polynomial	envy cycle	Plaut and Roughgarden [117]
			chores				envy cycle	Li, Li, and Wu [99]
			comb		?	?	?	?
EFX	-	any	goods	2-value instances	yes	polynomial	match&freeze	Amanatidis et al. [5]
EFX			goods	interval valuation $[v_i - 2v_i]$	yes	polynomial	modified RR	Amanatidis et al. [5]
EFX	-	$n = 3$	goods	additive	yes	pseudo polynomial		Chaudhury, Garg, and Mehlhorn [57]
EFX	-	$n = 2$	goods	general	yes	exponential	cut-choose, leximin++	Plaut and Roughgarden [117]
	-		chores				cut-choose	Implied Zhou and Wu [142]
			comb				cut-choose	Implied
EFX	-	$n = 2$	goods	additive		polynomial	envy cycle	Plaut and Roughgarden [117]
1/2 EFX	-	any	goods	subadditive	yes	polynomial	envy cycle	Plaut and Roughgarden [117]
0.618 EFX	-	any	goods	additive	yes	polynomial		Amanatidis, Markakis, and Ntokos [8]
EFX with charity	-	any	goods	additive	yes	polynomial		Caragiannis, Gravin, and Huang [53]
approx EFX with charity	-	any	goods	additive	yes	polynomial		Chaudhury et al. [58]
approx EFX	any	any	chores/ comb	any	?	?	?	?
EFX with charity	any	any	chores/ comb	any	?	?	?	?

Continued on next page

Fairness	Efficiency	Agents	Items	Valuations	Existence	Computation	Algorithm	Paper
EFX	PO	$n = 2$	goods	general nonzero marginal utility		exponential	cut-choose, leximin++	Plaut and Roughgarden [117]
EFX	PO	any	goods	identical general non zero marginal		exponential	leximin++	Plaut and Roughgarden [117]
EFX	PO	any	goods	2-value instances		exponential	MNW	Amanatidis et al. [5]
EFX, PROPX, 4/3 MMS			chores	IDO additive		polynomial	envy cycle	Li, Li, and Wu [99]
PROP	any	any	any	any	no	NP-Hard	?	?
PROP1	-	any	goods	additive	yes	polynomial	EF1 implies PROP1 for additive	Caragiannis et al. [55] Aziz et al. [14]
			chores					
			comb					
connected PROP1	-	any	any	additive	yes	polynomial		Aziz et al. [14]
wPROP1	fPO	any	any	additive	yes	polynomial	market equilibria	Aziz, Moulin, and Sandomirskiy [20]
PROP1	USW	any	goods	additive	yes	exponential	dynamic programming	Aziz et al. [16]
			any			?	?	?
PROPX wPROPX	-	any	goods	additive	no	-	-	Aziz et al. [14]
			comb					Implied
			chores			polynomial	envy cycle, bid and take	Li, Li, and Wu [99]
PROPX	PO	any	chores	additive	?	?	?	?
PROPX	USW	any	chores	additive	yes	NP-Hard	?	?
PROPm	-	any	goods	additive	yes	polynomial	divide and conquer plus graph reassigning	Baklanov et al. [25]
PROPm			comb	additive	?	?	?	?
PROPm	PO	any	any	additive	?	?	?	?
PROP	USW	any	any	additive	yes	NP-Hard	?	?
MMS	-	any	any	additive	no	NP-Hard	-	Procaccia and Wang [118] Aziz et al. [21]
MMS	-	any	any	IDO	yes	NP-Hard		Bouveret and Lematre [45]
MMS	-	any	goods	0,1	yes	polynomial	RR with modification	Bouveret and Lematre [45]
MMS	-	any	goods	0,1,2	yes	polynomial	RR with modification	Amanatidis et al. [7]
MMS	-	$m \leq n+3$	goods	additive	yes			Bouveret and Lematre [45]
MMS	-	$n = 2$	goods	additive	yes	exponential	cut and choose	Bouveret and Lematre [45]
3/4 MMS	-	any	goods	additive	yes	polynomial	valid reduction, clustering, bag filling	Garg and Taki [78]
3/4+1/12n	-	any	goods	additive	yes	exponential		Garg and Taki [78]
11/9 MMS	-	any	chores	additive	yes	exponential	modified bag filling with threshold	Huang and Lu [90]
5/4 MMS	-	any	chores	additive	yes	polynomial		Huang and Lu [90]
α-MMS	any	any	comb	additive	no	-	-	Kulkarni, Mehta, and Taki [96]
α-MMS	PO	any	any	additive	?	?	?	?
α-MMS	USW	any	any	any	yes	?	?	?

2.4 Combination of Indivisible Goods and Chores

2.4.1 Fair Allocation

Envy-freeness up to one item (EF1):

Aziz et al. proposed a *double round-robin* algorithm for the combination of goods and chores when agents have additive valuations for computing EF1 allocation in polynomial time. The algorithm involves two phases and applies the round-robin method twice, once in a clockwise direction and once in an anticlockwise direction. During the first phase, the algorithm allocates chores to agents based on their non-positive utility using the round-robin algorithm. In the second phase, the algorithm allocates the remaining goods to agents using the reversed round-robin algorithm, starting with the agent who chose the last in the first phase.

For more general valuations, Bhaskar, Sricharan, and Vaish presented a modified version of the *envy cycle elimination*, which allocates goods and chores to agents in a doubly monotone instance where the items are indivisible. The algorithm runs in two phases. In the first phase, the algorithm allocates items that are goods for at least one agent using the envy-cycle elimination algorithm [101]. However, this is done only for the subgraph of agents who consider the item a good. In the second phase, the algorithm allocates items that are chores to the remaining agents using the top-trading envy cycle elimination algorithm. The paper's authors have proven that the modified algorithm returns an EF1 (envy-free up to one item) allocation. Additionally, they maintain the invariant that the partial allocation remains EF1 at every algorithm step. Moreover, the algorithm terminates in *polynomial time*.

Envy-freeness up to any item (EFX):

Hosseini et al. investigated the problem of finding EFX allocation in the presence of *lexicographic valuations*, a subclass of additive valuations, where items can be goods or chores. The authors demonstrated that an envy-free allocation existence is computationally hard for instances where only chores are considered under lexicographic preferences, in contrast to the polynomial-time solvability of the goods-only case. They also showed that even in the case of combinations of goods and chores, an EFX allocation may not exist. Despite these challenges, the authors identified a domain restriction where EFX allocations are guaranteed to exist and can be computed efficiently. Furthermore, they showed that a different notion of fairness, maximin share (MMS), always exists and can be computed efficiently for any mixed instance with lexicographic preferences.

Algorithm 11 *EF1 for Combination of Goods and Chores with General Valuations*

Initialize $A \leftarrow (\emptyset, \emptyset, \ldots, \emptyset)$
// For each good
for $k \leftarrow 1$ *to* m **do**
    $V^g \leftarrow i \in [n]; g \in [m]; |v_{ig} \geq 0$ // Agents who value the good positively
    Choose a source agent j from the envy graph G_A^g
    $A_j \leftarrow A_j \cup k$
    while G_A has a directed cycle C **do**
        $A \leftarrow A^C$ // Remove the cycle from the allocation
    end while
end for
// For each chore
for $k \leftarrow 1$ *to* m **do**
    if there is no sink agent in envy graph G_A **then**
        $C \leftarrow$ any cycle in T_A
        $A \leftarrow A^C$ // Remove the cycle from the allocation
    end if
    Choose a sink agent j from the envy graph G_A
    $A_j \leftarrow A_j \cup k$
end forreturn Allocation A

Proportionality up to one item (PROP1):

In the case of additive valuations, the EF1 allocation implies the PROP1 allocation. Consequently, any algorithm that gives EF1 allocations also guarantees PROP1. Aziz et al. [14] proved that a *contiguous PROP1* allocation of a combination of goods and chores could be computed in polynomial time for additive utilities. This exciting result paves the way for practical algorithms that can achieve fairness and connectivity in allocating indivisible goods and chores.

Proportionality up to any item (PROPX):

PROPX requires that each agent receives a utility that is at least proportional to their share, even after we add a small good allocated among the remaining agents to their bundle or remove the smallest chore (least negative) from the agent's bundle as mentioned in Definition 3.2. However, PROPX may not exist even for three agents when items are only goods [20]. Therefore, it can be concluded that PROPX may not exist in the case of a combination of goods and chores. However, this is a much less explored domain. Although we know that PROPm exists in polynomial time for goods and PROPX exists in polynomial time for chores, we have yet to explore a stronger notion than PROP1 but weaker for proportionality in the case of goods and chores.

Maximin Share (MMS):

When allocating indivisible goods or chores among a group of agents, MMS allocation is often used to ensure fairness. However, the MMS allocation may not exist when goods or chores are indivisible. Hence, in the case of the combination of goods and chores, MMS allocation may not exist.

To address this issue, researchers have explored the concept of approximate MMS (α-MMS) allocation, which aims to allocate goods or chores to approximate the ideal MMS allocation. The best-known approximate solution for α-MMS in goods is $3/4 + 1/12n$, where n is the number of agents.

In a recent study, Kulkarni, Mehta, and Taki introduced the problem of finding the (near) best $\alpha \in (0, 1]$ for which an α-MMS allocation exists. They showed that an α-MMS allocation *may not always exist* for any given $\alpha > 0$, making it challenging to solve the problem for a fixed α. To tackle this challenge, the authors developed an efficient algorithm to find an α-MMS and Pareto Optimal (PO) allocation for the maximum $\alpha \in (0, 1]$ for which it exists. This approach can provide a fair allocation of indivisible goods or chores, even when an exact MMS solution is not possible.

2.4.2 Fair and Efficient Allocation

Efficient EF1 allocations

The concept of allocating goods and chores fairly and efficiently has been studied recently. In Section 2.3, it was noted that when valuations are negative, it is still an unsolved issue to find an allocation that meets both EF1 (envy-freeness up to one item) and PO (proportional) properties. While there is a pseudo-polynomial time algorithm for determining PO and EF1 allocation in the case of goods, the existence and complexity of such an allocation for a combination of goods and chores remain unclear. However, Aleksandrov and Walsh has explored a weaker form of EF1 and PO for mixed sets of goods and chores.

Another area of interest is maximizing utilitarian welfare while ensuring EF1 allocation in mixed good and chore allocation. Unfortunately, this problem has not been fully explored yet.

Efficient PROP1 allocations

The paper by Aziz, Moulin, and Sandomirskiy [20] presents a *strongly polynomial time* algorithm for computing allocations that are Pareto optimal and PROP1 for both goods and chores. They demonstrate that an fPO and PROP1 allocation always exists, even for mixed utilities and any number of agents. To achieve this, they first compute a proportional fPO allocation with divisible items and then round it to preserve the fPO property and ensure PROP1. The paper by Brânzei and Sandomirskiy and Barman and Krishnamurthy use the same methodology but rely heavily on the concept of *competitive equilibrium with equal incomes (CEEI)*, which does not apply to economies with a mixture of goods and chores. Therefore, the authors of [20] construct a rounding procedure that does not rely on equilibrium prices and applies to any fPO proportional allocation of divisible items with an acyclic consumption graph. This is achieved by starting with an equal division and then finding a Pareto-dominating allocation through sequential cyclic trades, ensuring the consumption graph remains acyclic. The algorithm picks an agent who shares some items with other agents and rounds all their fractions to their advantage, breaking all other partial shares in the corresponding sub-tree. The acyclicity of the tree ensures that the algorithm terminates and returns a PROP1 allocation.

In conclusion, despite the challenges posed by the combination of goods and chores, there are fair allocation criteria that can be computed efficiently. However, there is still much room for further research, particularly in maximizing welfare while ensuring fairness constraints. We summarize the results of goods and chores in the following image.

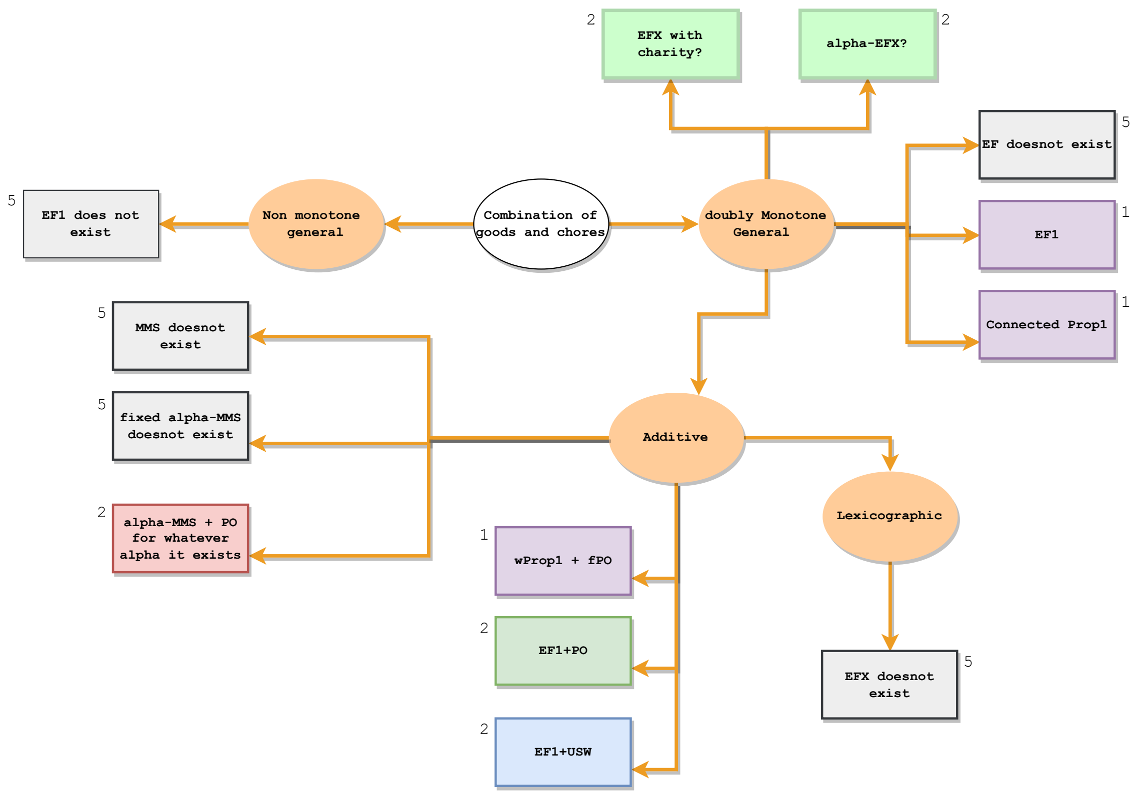

Figure 2.9: Summary of fairness and efficient notion for indivisible goods and chores. The Violet box (number 1) represents the polynomial-time algorithm. The blue box (number 2) represents exponential time. The red box (number 3) represents Pseudo-polynomial time. The green box (number 4) represents an open question. The grey box (number 5) represents such an allocation does not exist.

2.5 Mixed Indivisible and Divisible Goods and Chores

The world of fair division literature has long been fixated on the concept of items being either divisible or indivisible. But what happens when reality throws us a curveball, and we're suddenly faced with the challenge of allocating a mixed bag of divisible and indivisible items? As it turns out, this is a very real predicament many of us encounter daily, as evidenced by the classic example of room sharing and rent division. This scenario poses the intriguing possibility of some items being allocated fractionally while others must be claimed in their entirety. To break it down even further, let $D = k \in M \mid k$ is divisible item and $I = k \in M \mid k$ is indivisible item , with $D \cap I = \emptyset$ and $D \cup I = M$. So the question becomes: how do we navigate this complex terrain of mixed divisible and indivisible items to achieve a fair and equitable division for all involved?

The study of fair division with mixed types of resources was initiated by Bei et al. in their research, in which they explored envy-freeness for mixed goods, including both indivisible and divisible items. Further, Bhaskar, Sricharan, and Vaish delved into envy-freeness for mixed goods/chores, encompassing both indivisible goods/chores and divisible chores. While they also attempted to investigate envy-freeness for indivisible chores and divisible goods, their research yielded limited positive results, leaving room for further exploration.

Since envy-freeness cannot be achieved for indivisible items, relaxation techniques must be considered. However, EF1 (envy-freeness up to one item) can be too weak when only divisible resources are present. This raises the question of whether we can combine a relaxation notion that captures EF and EF1. For instance, imagine Agastya and Noorie are tasked with cleaning their house (a divisible chore) and dividing a bicycle between them (an indivisible good). What would be a fair allocation in this situation? If we use EF1 for indivisible items and EF for divisible items, both would clean the house equally, and Agastya would get the bicycle. However, is this allocation truly fair? A fairer allocation would be for Agastya to clean the house and receive the bicycle. This highlights the need to reconsider our notions of fairness and efficiency in such settings.

Bei et al. proposed a new concept called *EFM (envy-freeness for mixed goods)*, which applies to both indivisible and divisible goods. This notion was then extended by Bhaskar, Sricharan, and Vaish to include indivisible goods/chores and divisible goods/chores. In essence, EFM means that any agent who owns divisible goods should not be envied, any agent who owns divisible chores should not envy anyone else, and, subject to these conditions, any pairwise envy should be EF1. In summary, EFM provides a more comprehensive and nuanced approach to achieving fairness in allocating mixed types of resources. Next, we define EFM for various scenarios.

Divisible Good, Indivisible Good (D, I both consists of goods)[35] In the case of divisible and indivisible goods, an allocation is considered EFM if an agent's allocation contains a divisible good, then other agents compare their allocation to that agent using the EF criterion. However, if the allocation contains only indivisible goods, then the other agents compare their allocation to that agent's bundle using EF1. More formally, $\forall i, j \in N$,

$$\left.\begin{array}{ll} v_i(A_i) \geq v_i(A_j) & ;\ \exists k \in A_j \text{ s.t. } k \text{ is divisible good} \\ v_i(A_i) \geq v_i(A_j \setminus k) & ;\ \exists k \in A_j; A_j \subseteq I \end{array}\right\} \textit{is EFM}$$

Divisible Good, Indivisible Chore(D consists of goods, I consists of chores)[40]
In the case of divisible goods and indivisible chores, an allocation is EFM if an agent's allocation contains a divisible good, then other agents compare their allocation to that agent using the EF criterion. However, if the allocation contains any indivisible chore, the agent compares their allocation to other agents' allocation using EF1. More formally, $\forall i, j \in N$,

$$\left.\begin{array}{ll} v_i(A_i) \geq v_i(A_j) & ;\ \exists k \in A_j \text{ s.t. } k \text{ is divisible good} \\ v_i(A_i \setminus k) \geq v_i(A_j) & ;\ \exists k \in A_i; v_{ik} < 0 \end{array}\right\} \textit{is EFM}$$

Divisible Chore, Indivisible Good(D consists of goods, I consists of chores)[40] In the case of divisible chores and indivisible goods, an allocation is EFM if an agent's allocation contains a divisible chore, then the agent compares their allocation to other agents' allocation using the EF criterion. However, if the allocation contains an indivisible good, the other agents compare their allocation to that agent using EF1 More formally, $\forall i, j \in N$,

$$\left.\begin{array}{ll} v_i(A_i) \geq v_i(A_j) & ;\ \exists k \in A_i \text{ s.t. } k \text{ is divisible chore or} \\ v_i(A_i) \geq v_i(A_j \setminus k) & ;\ \exists k \in A_j; v_{ik} > 0 \end{array}\right\} \textit{is EFM}$$

Divisible Chore, Indivisible Chore (D consists of chores, I consists of chores) [40] Lastly, in the case of divisible and indivisible chores, an allocation is EFM if an agent's allocation contains a divisible chore, then other agents compare their allocation to that agent using the EF criterion. However, if the allocation contains an indivisible chore, the agent compares their allocation to other agents' allocation using EF1. More formally, $\forall i, j \in N$,

$$\left.\begin{array}{ll} v_i(A_i) \geq v_i(A_j) & ;\ \exists k \in A_i \text{ s.t. } k \text{ is divisible chore or} \\ v_i(A_i \setminus k) \geq v_i(A_j) & ;\ \exists k \in A_i; A_i \subseteq I \end{array}\right\} \textit{is EFM}$$

The paper by Bei et al.[35] examines the problem of fairly dividing a set of heterogeneous divisible goods and a collection of indivisible goods. The authors prove that an allocation that satisfies envy-freeness for mixed resources (EFM) for mixed goods with additive valuations for divisible goods and general monotone indivisible goods always exists. The algorithm used to achieve this requires an oracle for computing a perfect allocation in cake cutting but can compute an EFM allocation in a polynomial number of steps. The algorithm starts by obtaining an arbitrary EF1 allocation (using the *Envy-cycle elimination* algorithm) of the indivisible goods, and then allocates the divisible goods to obtain an EFM allocation. While the paper's algorithm is only proven for additive valuation for indivisible goods, [40] shows that it also works for general monotone valuation.

Additionally, the authors present two algorithms for computing an EFM allocation for special cases without using the perfect allocation oracle. The first algorithm is for two agents with general additive valuations in the Robertson-Webb model, and the second is for any number of agents with piece-wise linear valuation functions.

In the Robertson-Webb model, the authors present an algorithm that computes an *ϵ-EFM allocation* with a polynomial running time complexity that is dependent on the number of agents, the number of indivisible goods, and $1/\epsilon$. The authors note that it is unclear whether an EFM allocation can be computed in a finite number of steps in the Robertson-Webb model in general. The algorithm does not require a perfect allocation oracle, which is an appealing result due to its polynomial running time complexity. In contrast, a bounded exact EFM protocol, if it exists, is likely to require a large number of queries and cuts. For instance, in the special case when all resources are divisible, EFM reduces to EF in cake cutting, and the best-known protocol [19] for this problem has a very high query complexity (a tower of exponents of n).

Bhaskar, Sricharan, and Vaish[40] investigates a mixed model consisting of indivisible items (goods/chores) and a divisible heterogeneous chore. The authors extend the definition of EFM to this model and prove that an EFM allocation always exists. They use a similar algorithmic approach to that in [35], starting with an arbitrary EF1 allocation of the indivisible item and then allocating the cake to obtain an EFM allocation.

Finally, the authors of [40] also study a mixed model with *indivisible chores and divisible goods*, which proves to be more challenging. Unlike previous cases, it is not possible to start with an arbitrary EF1 allocation of the indivisible items and allocate the cake to obtain an EFM allocation. However, the authors show that an EFM allocation exists for two special cases in this model: when each agent has the same ranking over the chores, and when the number of chores is at most one more than the number of agents. In particular, for two agents, an EFM allocation always exists in this setting, and with n agents, m additive indivisible chores, and cake, where $m \leq n + 1$, an EFM allocation exists.

We have summarized the results for a mixed setting of divisible and indivisible item instances in Table 2.2. From the table, it can be observed that this domain has been relatively less explored but is highly relevant and has the potential to yield exciting findings.

Table 2.2: Summary of Fair Division for Mixed Model

Fairness	Agents	Divisible Item	Divisible Valuation	Indivisible Item	Indivisible Valuation	Paper	Computation	Details
EFM	any	good	additive	good	general	Bei et al. [35]	polynomial	With perfect allocation oracle
EFM	any	good	additive	chores	general	?	?	
EFM	any	chores	additive	good chores combination	general	Bhaskar, Sricharan, and Vaish [40]	polynomial	With perfect allocation orcale
		good		good	general	Bei et al. [35]	polynomial	
EFM	$n = 2$	good chore	general	combination	general	?	?	With perfect allocation orcale
		good		good			polynomial	
EFM	any		piece wise linear	combination	general	?	?	With perfect allocation orcale
		chore				?	?	
EFM	$n = 2$	good	additive	chore	general	Bhaskar, Sricharan, and Vaish [40]	polynomial	With perfect allocation orcale
	any	good	additive identical for n-1					
	$m \leq n+1$	good	additive					
		good		good		Bei et al. [35]	polynomial	
ϵ-EFM	any		additive	combination	general	?	?	Without perfect allocation orcale
		chores						

2.6 Summary of Complexity Results for Fair Division

The problems of computing allocations that maximize the egalitarian welfare (utility of the worst-off agent) or Nash welfare (geometric mean of the utilities of agents) have been shown to be NP-hard, as demonstrated in prior research by Nguyen and Rothe. Moreover, even computing an exact maximin share for an agent with additive valuations has been proven to be an NP-complete problem, as shown through a reduction from the NP-hard problem of Partition. Additionally, in the case of agents with binary additive valuations for goods, checking for the existence of an envy-free (EF) allocation has also been shown to be NP-complete, as demonstrated in studies by Aziz et al., Hosseini et al. Furthermore, the problem of making efficiency improvements in fair allocation has been identified as

NP-hard in research by Aziz et al., further highlighting the computational complexity of these allocation problems.

In addition to the computational complexity, it is important to consider the likelihood of finding a fair allocation in real-world scenarios. Research has explored the conditions under which a fair allocation is likely to exist or not, shedding light on the practical feasibility of achieving fairness in allocation mechanisms.

2.7 Other Important Results

In this section, we will provide a brief overview of other important aspects in the field of fair division, including the likelihood of finding a fair allocation, and the price of fairness.

2.7.1 Likelihood of Finding a Fair Allocation

Dickerson et al. was the first to address this question, under the assumption that utilities are additive and each agent's utilities for individual items are drawn from probability distributions. They established that an envy-free allocation is likely to exist with high probability when the number of items m is at least $\Omega(nlogn)$, but not when $m = n + o(n)$. Suksompong investigated the asymptotic existence of proportional allocations - which are weaker than envy-free allocations under the additivity assumption - and showed that such allocations occur with high probability provided that either m is a multiple of n or $m = w(n)$.

Manurangsi and Suksompong showed that EFX allocations, which satisfy envy-freeness up to any item, exist with high probability for any number of agents and items under the assumption that the valuations of the agents are drawn at random from a probability distribution. When agents' valuations for individual items are randomly drawn from a probability distribution, they show that the classical round-robin algorithm is likely to result in an envy-free allocation, particularly when the number of items m, satisfies the condition $m = \Omega(n \log n / \log \log n)$. Furthermore, they demonstrated that a proportional allocation is likely to exist with high probability as long as the number of items, m, is greater than or equal to the number of agents, n. Additionally, an allocation that satisfies envy-freeness up to any item (EFX) is likely to be present for any relationship between m and n. They also demonstrated the challenge of showing the existence of EFX allocations, as in instances with two agents there can be as few as two EFX allocations, while the number of EF1 allocations is always exponential in the number of items.

Amanatidis et al. undertook a probabilistic analysis and proved that in randomly generated instances, maximin share allocations exist with high probability, motivated by the apparent difficulty in establishing lower bounds in calculating α-MMS. Furthermore, Procaccia and Wang showed that the non-existence of MMW (maximin share with weights) allocation requires a number of items that is exponential in the number of players. In contrast, if the number of items is only slightly larger than the number of players, an MMS allocation is guaranteed to exist [45]. This raises the question of what is the largest number of items for which an MMS allocation is guaranteed to exist.

In conclusion, the likelihood of finding a fair allocation depends on various factors such as the specific fairness notion, the type of valuations, and the number of agents and items involved. Probabilistic analyses have been conducted to understand the conditions under which fair allocations are likely to exist, shedding light on the practical feasibility of achieving fairness in real-world scenarios.

2.7.2 Price of fairness

The concept of the price of fairness quantitatively measures the *tradeoff between fairness and social welfare*. It represents the ratio of the maximum welfare of any allocation to the maximum welfare of any allocation that satisfies the desired fairness notion, and it provides insights into the efficiency loss incurred in achieving fairness.

The authors, Caragiannis et al., investigated the effect of fairness on the efficiency of resource allocations. They examined three fairness criteria, namely proportionality, envy-freeness, and equitability, for allocations of divisible and indivisible goods and chores. They presented a set of findings on the price of fairness under each of these criteria that quantify the loss in efficiency in fair allocations compared to optimal ones.

For indivisible goods, the authors derived an exact bound of $n + 1/n$ on the price of proportionality, while they demonstrated that the price of envy-freeness is $\theta(n)$. In contrast, for indivisible chores, they established an exact bound of n on the price of proportionality, but both the price of envy-freeness and equitability are infinite. These results indicate that, in the case of indivisible chores, envy-freeness and equitability are typically incompatible with efficiency.

Their work provides insights into the trade-off between efficiency and fairness in resource allocation, considering both divisible and indivisible goods. However, a limitation of their study is that fairness notions may not always be satisfied for indivisible goods, which they addressed by ignoring such instances in their analysis of the price of fairness. This omission may not capture certain scenarios that can arise in practice. Furthermore, the assumption that envy-free allocations are always proportional may not necessarily hold true, as there are instances where proportional allocations exist but envy-free allocations do not.

In the study by [37], the focus shifts from classical fairness notions, such as envy-freeness, proportionality, and equitability, which may not always be achievable for indivisible goods, to notions with guaranteed existence, including envy-freeness up to one good (EF1), balancedness, maximum Nash welfare (MNW), and leximin. They also introduce the concept of the strong price of fairness, which captures the efficiency loss in the worst fair allocation. They provide tight or asymptotically tight bounds on the worst-case efficiency loss for allocations satisfying these notions, for both the price of fairness and the strong price of fairness.

Specifically, the study shows that for the price of EF1, a lower bound of $\Omega(\sqrt{n})$ and an upper bound of $O(n)$ are provided. Additionally, it is shown that two common methods for obtaining an EF1 allocation, namely the round-robin algorithm and MNW, have a price of fairness of linear order (with the exact price being n for round-robin), indicating that these methods cannot improve the upper bound for EF1. For MNW, MEW, and leximin, an asymptotically tight bound of $\theta(n)$ on the price of fairness is proven. The results suggest that round-robin is a promising allocation method, as it produces an EF1 allocation with high welfare, is simple and intuitive, and always produces a balanced allocation.

The authors of [27] address the impact of fairness guarantees on social welfare in the allocation of indivisible goods by resolving the price of two well-known fairness notions: envy-freeness up to one good (EF1) and approximate maximin share (MMS). They show that the price of fairness is $O(\sqrt{n})$ for both EF1 and 1/2-MMS, using different techniques. The $\Omega(\sqrt{n})$ lower bound due to Bei et al. is matched by their upper bound, which holds for the more general class of subadditive valuations, as opposed to additive valuations. Their work therefore resolves this open question for all valuation classes between additive and subadditive and also settles the price of proportionality up to one good (Prop1) as $\theta(\sqrt{n})$ for additive valuations.

For the 1/2-MMS fairness notion and additive valuations, the authors also show that the price of fairness is $\theta(\sqrt{n})$ using a different algorithm. They prove that for a fixed $\epsilon > 0$, a $(1/2-\epsilon)$-MMS allocation with welfare within $O(\sqrt{n})$ factor of the optimal can be computed in polynomial time. Overall, their main contribution is comprehensively settling the price of EF1 and 1/2-MMS fairness notions, and providing efficient algorithms for obtaining their upper bounds.

Li, Li, and Wu investigate the price of fairness (PoF), which measures the loss in social welfare when enforcing allocations to be (weighted) proportional (PROPX). They prove that the tight ratio for PoF is $\theta(n)$ for symmetric agents, indicating that the loss in social welfare can be significant. However, they also show that the PoF can be unbounded for asymmetric agents, suggesting that fairness constraints can have a drastic impact on social welfare in such cases.

2.7.3 Different Settings in Fair Division

After exploring different notions of fairness and their complexities, we will now provide a brief overview of various settings to highlight the intricacies of fair division. It's important to note that fairness is subjective and can vary depending on the setting. What may be considered fair in one context may not be viewed as fair in another. Therefore, it is necessary to outline all the possible settings before delving into a detailed analysis of a specific setting.

Different Types of Items The literature on fair division encompasses various items, including divisible, indivisible, or a combination of both. Divisible items can be divided into smaller portions, such as disk space in a CPU, cake for children, money for employees, land among siblings, or rent among housemates. Divisible items can be either homogeneous, where the value is uniform across the item or heterogeneous, where the values may differ. There is rich literature on ensuring envy-free allocation for divisible items, with numerous possibilities to explore.

On the other hand, indivisible items cannot be further divided and must be allocated entirely to one person or not at all. Indivisible items include books, cars, antiques, laptops, and trees. Ensuring fairness in allocating indivisible items is more challenging than divisible items. For instance, allocating one car among two people can never be truly fair, as giving it to one person would inherently be unfair to the other. Achieving fairness in allocating indivisible items often requires considering both efficiency and fairness measures. Indivisible items can also include goods or chores, such as distributing free books based on people's preferences, whereas allocating a book outside someone's preference can result in negative utility.

In practice, situations often arise where a mixed set of divisible and indivisible items must be allocated, for example, dividing rent and room allocation among friends sharing an apartment with different room preferences or dividing chores and a mobile phone among roommates. Allocating fairly in such mixed settings requires re-considering fairness and efficiency to ensure a more equitable allocation.

Next, we list various possibilities for agents to state their valuation/preferences over a bundle of items.

Different Types of Valuation functions

In this section, we consider different ways agents can express their preferences or valuations for bundles of items. First, in the combinatorial setting, each agent is asked to value 2^m possible bundles of m indivisible items that must be divided among n agents. Various types of valuations can be used for such bundles, such as sub-additive, super-additive, and xor. As an illustration, consider the items of a left shoe and a right shoe, which are examples of super-additive items due to their complementarity. This means that the combined value of the left and right shoes is greater than the sum of their individual values. On the other hand, items such as the Luxor Pen and the Pentonic Pen are sub-additive, as they are substitutes for each other. Agents would typically only desire one pen, thus, the value of each individual pen is either greater than or equal to the combined value of both pens. However, it can be challenging for agents and social planners to evaluate such a large number of valuations, even with learning-based preference elicitation techniques. In the case where 30 items are to be distributed, the system is requesting agents to evaluate over a billion item-bundle values, which can be an exceedingly complex task. There exists a plethora of literature on the intricacies involved in eliciting preferences [123].

Second, the literature mainly assumes that agents have additive valuations over bundles of items, which capture the essence of agents' valuations in a simple yet accurate manner. Additionally, many real-world scenarios involve agents who only express their preferences over the items they like or dislike, known as binary valuations, and there are several modifications to this class, such as dichotomous preferences, (0,1)-OXS, matroid rank valuation functions, single-mindedness, etc.

Third, agents may only report partial knowledge of their valuations, such as their top k items out of a large list of items, due to the difficulty of evaluating additive valuations for many items. This is known as the partial information setting. This scenario arises when thousands of items are to be divided among a group of interested agents, making it challenging to elicit additive valuation. Hence agents report only their top favorite items. For example, in a research institute, a department such as Security may not report their valuation of an item such as a microscope and instead focus on valuing items that are more important to them. The paper [85] redefines fairness notions applicable to this setting and provides results on EF1 and MMS fairness.

In another scenario, agents only report partial knowledge about their valuations, i.e., they report the top k valued item out of all the listed items. This scenario arises when thousands of items are to be divided among a group of interested agents, making it challenging to elicit additive valuation. Hence agents report only their top favorite items. For example, in a research institute, a department such as Security may not report their valuation of an item such as a microscope. Instead, they focus on value items that are more important to them. The paper [85] redefines fairness notions applicable to this setting and provides results on EF1 and MMS fairness.

Fourth, some agents may find it tedious to evaluate the exact value of a bundle of items, in which case ordinal valuations can be used. In this setting, each agent is equipped with a preference relation that expresses their preference for one set of items over another. Ordinal preferences are often used in practical applications where the algorithm cannot collect complete information on agent preferences. In this case, the algorithm can use partial information to compute approximately fair allocations [15, 44]. The literature has studied various notions of envy-freeness and proportionality in the ordinal setting. For example, people ranking their preferences in voting, schools, or university applications can be modeled using ordinal preferences. Finally, recent studies have shown that approximately fair allocations can be computed using ordinal preferences, even with partial information.

Lastly, Researches are also exploring the problem of allocating indivisible goods among strategic agents and focus on settings where monetary transfers are not available. Truthfulness is a key desideratum in this context, meaning mechanisms should ensure that participating agents cannot gain by misreporting their valuations. However, achieving the central objectives of fairness, economic efficiency, and truthfulness together in settings without monetary transfers is challenging, even under additive valuations [33].

Next, we list various possibilities in different settings arrive to model real-life settings introducing complexity in allocating items fairly.

Different Types of Setting

Online setting: Most of the literature on fair division deals with a fixed number of agents and items, which is not reflective of real-life settings where the number of agents or items may change over time [3, 138]. Examples of such settings include allocating parking slots to cars, donations to changing recipients, allocating resources to system processes, and enrolling students in Ph.D. programs. Papers have been written on online fair division in various settings, including fixed agents and changing items, changing agents and fixed items, and changing agents and items. Traditional offline fairness approaches may not be effective in these online settings.

Two-sided matching settings: Many real-life scenarios involve two-sided fairness, where fairness needs to be ensured for both parties involved [10]. Examples include online platforms like Amazon and Airbnb, matching students to schools or employees to employers, and allocating advisors to students in Ph.D. programs. Papers have been written on recent developments and guarantees for two-sided fairness, such as double envy-freeness up to one match (DEF1) and double maximin share guarantees (DMMS), which may not exist in certain cases.

General models with arbitrary shares: There is a growing literature on fair division where agents may have arbitrary and possibly unequal shares of items [116]. Examples include peer review in academia, where papers and reviewers have preferences for each other, and conference or course allocation settings, where conflicting items need to be fairly allocated. Papers have been written on these settings and additional constraints must be considered.

Fair division with unequal entitlements: In specific scenarios, agents may have unequal entitlements to items based on their membership status or employment type, and fairness needs to be ensured accordingly [91]. Examples include membership plans in a resort, where different membership levels have different entitlements, and resource allocation for full-time and part-time employees in a company. Papers have been written on the fair allocation of items with unequal entitlements and how fairness notions can be applied in such settings.

Public Goods: In fair resource allocation, most work focuses on private goods that can only be assigned to a single agent. However, some goods are public and can be enjoyed by multiple agents simultaneously. Public goods allocation problems can arise in various contexts, such as voting for the allocation of public resources like schools, libraries, and museums in a city. Naively tallying votes may not always result in a fair allocation of public goods, as it may ignore the preferences of a large minority. Fairness axioms like proportionality do not capture the idea of fairness in public goods allocation, as they focus on individual agents' perspectives. Public goods allocation problems require striking a balance between agents with different preferences, as opposed to private goods allocation where tension arises between agents with similar preferences. The literature has moved beyond the private goods model to study fair allocation for public goods in various applications such as participatory budgeting, committee selection, and shared memory allocation. Proportional fairness, which balances fairness and efficiency, is a commonly studied notion in allocating public goods. One limitation of the current literature is that it mainly considers "one-shot" settings where all agents' preferences are known beforehand. At the same time, real-world scenarios often involve dynamic and online allocation decisions. Recent works have focused on designing online algorithms for fair resource allocation to capture the dynamic nature of fair division problems in practice [68, 26] .

Randomized Allocation In the literature survey, the concept of randomized allocation for indivisible goods is discussed. The idea of constructing a fractional assignment and implementing it as a lottery over pure assignments was introduced Hylland and Zeckhauser. The key question addressed is whether ex-ante envy-freeness can be achieved in combination with ex-post envy-freeness up to one good. It is shown that if randomization is allowed, a simple allocation where the entire set of goods is allocated to one agent at random can achieve ex-ante envy-freeness, but induces a large amount of envy ex-post. This has led to significant research on fairness in deterministic allocations of indivisible goods, such as envy-freeness up to one good (EF1), which requires removing envy by eliminating at most one good from the envied agent's bundle. Lotteries and randomization are commonly used to break ties in allocating indivisible objects, where agents may have equal priority rights for some objects. Examples of such allocation problems include public housing associations assigning apartments to residents, school districts assigning seats to students, and childcare cooperatives assigning chores to members. Furthermore, the concept of achieving ex-ante and ex-post envy-freeness through randomized allocations is discussed. It is shown that achieving ex-ante envy-freeness does not guarantee fairness in ex-post envy, and different allocations may have different levels of fairness [73, 11].

Repeated Allocation The literature explores various allocation mechanisms to achieve fairness in the repeated allocation of indivisible goods without the use of monetary transfers. This is relevant in scenarios such as allocating server time, rooms to students, or food among food banks. Additionally, the problem of fair matching has been studied in traditional setups such as marriage matching or matching medical residents to hospitals. Still, modern online matching platforms require investigating the two-sided fairness of providers and customers in a platform performing repeated matchings of providers and customers over time. The fair distribution of income for providers in on-demand and marketplace platforms is crucial for ensuring long-term stable platform operation, and its study has received little attention so far despite the increasing dependence of people on the sharing economy to earn a living [133, 82].

Group Fairness In the literature, group fairness is explored in the context of resource allocation in settings such as corporate environments [62]. For example, a manager may need to allocate resources such as interns, conference rooms, or time slots for shared machines to employees. The manager may aim to ensure that no business team envies another team and that there is no envy between genders, locations, roles, and other groups. Envy-freeness alone may not be sufficient in this setting, as an allocation that is envy-free up to one good may still result in significant levels of inequality and envy between groups of players. Another example can be, consider a university that needs to allocate research funding to its faculty members. The university may want to ensure that no department envies another department, that junior faculty do not envy senior faculty, that female faculty do not envy male faculty, that faculty members from one research area do not envy those from another area, and so on. Simply ensuring envy-freeness may not be sufficient in this case, as it may still result in significant levels of inequality and envy between different groups of faculty members. Therefore, the university may need to explore alternative fairness criteria and allocation mechanisms to ensure a fair and efficient allocation of research funding.

Externalities The concept of externalities in allocation refers to situations where an agent's utility not only depends on their own bundle but also on the bundles allocated to other agents. This scenario arises frequently in the allocation of essential goods such as hospital beds, ventilators, and vaccines during emergencies like the COVID-19 pandemic. The allocation of critical assets to one group in the division of assets among conflicting groups can also cause negative externalities for the other group's functionality. On the other hand, certain goods generate positive consumption externalities, such as the increased value of a PlayStation to an agent if more of their friends also own one. Positive externalities are also observed in healthcare, where individuals who are vaccinated reduce the risk of contraction for others around them.

2.8 Summary

In conclusion, fair resource allocation is a challenging problem that has been studied extensively in the literature. In this chapter, we focused on the fair allocation of indivisible goods and chores, as well as mixed instances that include both indivisible and divisible items. We discussed various fairness definitions, including EF, EF1, EFX, PROP, PROP1, PROPX, PROPm, MMS, and α-MMS, and their corresponding fair and efficient allocations, such as EF1+PO, EF1+USW, PROP1+PO, PROP1+USW, PROPm+PO/USW, and EFX+PO. All the existing results are summarized in Table 2.1 and 2.2. We also touched on computational complexity and the likelihood of finding a fair allocation in different settings, such as different types of items, valuations, and settings. Finally, we considered the price of fairness and its impact on the efficiency of the allocation. Overall, this survey provides a comprehensive overview of the state-of-the-art approaches to fair resource allocation for indivisible goods and chores. It also highlights the need for further research in this area, especially in developing efficient algorithms that can handle more complex settings and achieving a better trade-off between fairness and efficiency. Equitably dividing resources poses numerous challenges, prompting researchers in recent literature to investigate data-driven approaches in game theory and mechanism design. In the forthcoming chapter, we will conduct an in-depth analysis for fair division in the presence of externalities.

Chapter 3

Fair Allocation with Special Externalities

Most of the existing algorithms for fair division do not consider externalities. Under externalities, the utility of an agent depends not only on its allocation but also on other agents' allocation. An agent has a positive (negative) value for the assigned goods (chores). In this work, we study a special case of externality which we refer to as 2-D. In 2-D, an agent receives a positive or negative value for unassigned items independent of who receives them. We propose a simple valuation transformation and show that we can adapt existing algorithms using it to retain some of the fairness and efficiency notions in 2-D. However, proportionality doesn't extend in 2-D. We redefine PROP and its relaxation and show that we can adapt existing algorithms. Further, we prove that maximin share (MMS) may not have any multiplicative approximation in this setting. Studying this domain is a stepping stone towards full externalities where ensuring fairness is much more challenging.

3.1 Introduction

We consider the problem of allocating m *indivisible* items fairly among n agents who report their valuations for the items. These scenarios often arise in the division of inheritance among family members, divorce settlements and distribution of tasks among workers [46, 108, 126, 131, 132]. Economists have proposed many fairness and efficiency notions widely applicable in such real-world settings. Researchers also explore the computational aspects of some widely accepted fairness notions [55, 32, 93, 74, 118]. Such endeavours have led to web-based applications like Spliddit, The Fair Proposals System, Coursematch, etc. However, most approaches do not consider agents with *externalities*, which we believe is restrictive.

In the absence of externality, the utility corresponding to an unallocated item is zero. *Externality* implies that the agent's utility depends not only on their bundle but also on the bundles allocated to other agents. Such a scenario is relatively common, mainly in allocating necessary commodities. For example, the COVID-19 pandemic resulted in a sudden and steep requirement for life-supporting resources like hospital beds, ventilators, and vaccines. There has been a heavy disparity in handling resources across the globe. Even though there was a decrease in GDP worldwide, low-income countries suffered more than high-income countries. We can categorize externality into positive and negative; i.e., if it affects the agent positively, we refer to it as a positive externality and vice versa. Getting a vaccination affects an agent positively. The agent values it positively, possibly less, even if others get vaccinated instead of it. However, not receiving a ventilator results in negative utility for the patient and family. While there has been an increase in demand for pharmaceuticals, we see a steep decrease in travel. Such a complex valuation structure is modeled via externalities.

Generally with externalities, the utility of not receiving an item depends on which other agent receives it. That is, each agent's valuation for an item is an n-dimensional vector. The j^{th} component corresponds to the value an agent obtains if the item is allocated to agent j. In this work, we consider a special case of externalities in which the agents incur a cost/benefit for not receiving an item. Yet, the cost/benefit is *independent* of which other agent receives the item. This setting is referred to as 2-D, i.e., value v for receiving an item and v' otherwise. When there are only two agents, the 2-D domain is equivalent to the domain with general externalities. We refer to the agent valuations without externalities as 1-D. For the 2-D domain, we consider both goods/chores with positive/negative externality for the following fairness notions.

Fairness Notions. Envy-freeness (EF) is the most common fairness notion. It ensures that no agent has higher utility for other agents' allocation [72]. Consider 1-D setting with two agents - $\{1, 2\}$ and two goods - $\{g_1, g_2\}$; agent 1 values g_1 at 6 and g_2 at 5, while agent 2 values g_1 at 5 and g_2 at 6. Allocating g_1 and g_2 to agent 1 and 2, respectively, is EF. However, if agent 1 receives a utility of -1 and -100 for not receiving g_1 and g_2. And agent 2 receives a utility of -100 and -1 for not receiving g_1 and g_2; this allocation is no longer EF.

Externalities introduce complexity so much that the definition of proportionality cannot be adapted to the 2-D domain. Proportionality (PROP) ensures that every agent receives at least $1/n$ of its complete bundle value [131]. In the above example, each agent should receive goods worth at least 11/2. Guaranteeing this amount is impossible in 2-D, as it does not consider the dis-utility of not receiving goods. Moreover, it is known that EF implies PROP in the presence of additive valuations. However, in the case of 2-D, it need not be true, i.e., assigning g_2 to agent 1 and g_1 to agent 2 is EF but not PROP.

We consider a relaxation of PROP, the maximin share (MMS) allocation. Imagine asking an agent to divide the items into n bundles and take the minimum-valued bundle. The agent would divide the bundles to maximize the minimum utility, i.e., the MMS share of the agent. An MMS allocation guarantees every agent its MMS share. Even for 1-D valuations, MMS allocation may not exist; hence researchers find multiplicative approximation α-MMS. An α-MMS allocation guarantees at least α fraction of MMS to every agent. [78] provides an algorithm that guarantees $3/4 + 1/12n$-MMS for goods and authors in [90] guarantees 11/9-MMS for chores. In contrast, we prove that for 2-D valuation, it is impossible to guarantee multiplicative approximation to MMS. Thus, in order to guarantee existence results, we propose relaxed multiplicative approximation and also explore additive approximations of MMS guarantees.

In general, it is challenging to ensure fairness in the settings with full externality, hence the special case of 2-D proves promising. Moreover, in real-world applications, the 2-D valuations helps model various situations (e.g., COVID-19 resource allocation mentioned above). Studying 2-D domain is especially significant for α-MMS. We prove that there cannot exist any multiplicative approximation for MMS in 2-D. Therefore, we define Shifted α-MMS that always exists in 2-D. In summary our approach and contributions are as follows,

Our Approach. There is extensive literature available for fair allocations, and we primarily focus on leveraging existing algorithms to 2-D. We demonstrate in Section 3.3 that existing algorithms cannot directly be applied to 2-D. Towards guaranteeing fairness notion in 2-D, we propose a property preserving transformation $\mathfrak{T}$ that converts 2-D valuations to 1-D; i.e., an allocation that satisfies a property in 2-D also satisfies it in transformed 1-D and vice-versa.

Contributions.

1. We demonstrate in Section 3.3 that studying fair allocation with externalities is non-trivial and propose $\mathfrak{T}$ to retain fairness notions such as EF, MMS, and its additive relaxations and efficiency notions such as MUW and PO (Theorem 3.1). Thus, we can adapt the existing algorithms for the same.

2. We introduce PROP-E for general valuations for full externalities (Section 5.3) and derive relation with existing PROP extensions (Section 3.4).

3. We prove that α-MMS may not exist in 2-D (Theorem 3.2). We propose Shifted α-MMS, a novel way of approximating MMS in 2-D (Section 3.5.3).

Related Work

While fair resource division has an extremely rich literature, externalities is less explored. Velez [137] extended EF in externalities. [47] generalized PROP and EF for divisible goods with positive externalities. Seddighin et al. [125] proposed average-share, an extension of PROP, and studied MMS for goods with positive externalities. Authors in [22] explored EF1/EFX for the specific setting of two and three agents and provided PROP extension. For two agents, their setting is equivalent to 2-D, hence existing algorithms [55, 117, 20] suffice. Beyond two agents, the setting is more general and they proved the non-existence of EFX for three agents. In contrast, EFX always exists for three agents in our setting.

Envy-freeness up to one item (EF1) [51, 101] and Envy-freeness up to any item(EFX) [55] are prominent relaxation of EF. EF may not exist for indivisible items. We consider two prominent relaxations of EF, Envy-freeness up to one item (EF1) [51, 101] and Envy-freeness up to any item(EFX) [55]. We have poly-time algorithms to find EF1 in general monotone valuations for goods [101] and chores [40]. For additive valuations, EF1 can be found using Round Robin [55] in goods or chores, and Double Round Robin [13] in combination. When the valuations are general, non-monotone, and non-identical, a poly-time algorithm is given by [38] for two agents. [117] present an algorithm to find EFX allocation under identical general valuations for goods. [57] proved that an EFX allocation exists for three agents. Researchers have also studied fair division in presence of strategic agents, i.e., designing truthful mechanisms [29, 34, 115]. A great deal of research has been done on mechanism design [76, 75]. PROP1 and PROPX are popular relaxation of PROP. For additive valuations, EF1 implies PROP1, and EFX implies PROPX. Unfortunately, in paper [20], the authors showed the PROPX for goods may not always exists. [99] explored (weighted) PROPX and showed it exists in polynomial time. MMS do not always exist [97, 118]. The papers [118, 7, 28, 77] showed that 2/3-MMS for goods always exists. Paper [79, 78] showed that 3/4-MMS for goods always exists. Authors in [78] provides an algorithm that guarantees $3/4 + 1/12n$-MMS for goods. Authors in [21] presented a polynomial-time algorithm for 2-MMS for chores. . The algorithm presented in [30] gives 4/3-MMS for chores. Authors in [90] showed that 11/9-MMS for chores always exists. [96] explored $\alpha-$MMS for a combination of goods and chores. In [55] showed that MNW is EF1 and PO for indivisible goods and [32] gave a pseudo-polynomial time algorithm. [13] presented algorithm to find EF1 and PO for two agents. [20] presented an algorithm to find PROP1 and fPO for combination. [16] proposed a pseudo-polynomial time algorithm for finding utilitarian maximizing among EF1 or PROP1 in goods.

3.2 Preliminaries

We consider a resource allocation problem $(N, M, \mathcal{V})$ for determining an allocation A of $M = [m]$ indivisible items among $N = [n]$ interested agents, $m, n \in \mathbb{N}$. We only allow complete allocation and no two agents can receive the same item. That is, $A = (A_1, \ldots, A_n)$, s.t., $\forall i, j \in N,\ i \neq j; A_i \cap A_j = \emptyset$ and $\bigcup_i A_i = M$. A_{-i} denotes the set $M \setminus A_i$.

2-D Valuations. The valuation function is denoted by $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$; $\forall i \in N$, $V_i : 2^M \to \mathbb{R}^2$, $\forall S \subseteq M$, $V_i(S) = (v_i(S), v'_i(S))$, where $v_i(S)$ denotes the value for receiving bundle S and $v'_i(S)$ for not receiving S. The value of an agent i for item k in 2-D is (v_{ik}, v'_{ik}). If k is a good (chore), then $v_{ik} \geq 0$ ($v_{ik} \leq 0$). For positive (negative) externality $v'_{ik} \geq 0$ ($v'_{ik} \leq 0$).

The utility an agent $i \in N$ obtains for a bundle $S \subseteq M$ is, $u_i(S) = v_i(S) + v'_i(M \setminus S)$ Also, $u_i(\emptyset) = 0 + v'_i(M)$ and utilities in 2-D are not normalized[1]. When agents have additive valuations, $u_i(S) = \sum_{k \in S} v_{ik} + \sum_{k \notin S} v'_{ik}$. We assume monotonicity of utility for goods, i.e., $\forall S \subseteq T \subseteq M$, $u_i(S) \leq u_i(T)$ and anti-monotonicity of utility for chores, i.e, $u_i(S) \geq u_i(T)$. We use the term *full externalities* to represent complete externalities, i.e., each agent has n-dimensional vector for its valuation for an item. We next define fairness and efficiency notions.

[1] Utility is normalized when $u_i(\emptyset) = 0, \forall i$

Definition 3.1 (*Envy-free (EF) and relaxations* [17, 51, 55, 72, 137]). *For the items (chores or goods) an allocation A that satisfies $\forall i, j \in N$,*[2]

$$u_i(A_i) \geq u_i(A_j) \textit{ is EF}$$

$$\left.\begin{array}{l} v_{ik} < 0, u_i(A_i \setminus \{k\}) \geq u_i(A_j); \forall k \in A_i \\ v_{ik} > 0, u_i(A_i) \geq u_i(A_j \setminus \{k\}); \forall k \in A_j \end{array}\right\} \text{is EFX}$$

$$u_i(A_i \setminus \{k\}) \geq u_i(A_j \setminus \{k\}); \exists k \in \{A_i \cup A_j\} \text{is EF1}$$

Definition 3.2 (*Proportionality (PROP)* [131]). *An allocation A is said to be proportional, if $\forall i \in N$, $u_i(A_i) \geq \frac{1}{n} \cdot u_i(M)$.*

For 2-D, achieving PROP is impossible as discussed in Section 5.1. To capture proportional under externalities, we introduce *Proportionality with externality* (PROP-E). Informally, while PROP guarantees a $1/n$ share of the entire bundle, PROP-E guarantees a $1/n$ share of the sum of utilities for all bundles. Note that, PROP-E is not limited to 2-D and applies to full externalities. Formally,

Definition 3.3 (*Proportionality with externality (PROP-E)*). *An allocation A satisfies PROP-E if, $\forall i \in N$,$u_i(A_i) \geq \frac{1}{n} \cdot \sum_{j \in N} u_i(A_j)$*

Definition 3.4 (*PROP-E relaxations*). *An allocation A $\forall i, \forall j \in N$, satisfies PROPX-E if it is PROP-E up to any item, i.e.,*

$$\left.\begin{array}{l} v_{ik} > 0, u_i(A_i \cup \{k\}) \geq \frac{1}{n} \sum_{j \in N} u_i(A_j); \forall\ k \in \{M \setminus A_i\} \\ v_{ik} < 0, u_i(A_i \setminus \{k\}) \geq \frac{1}{n} \sum_{j \in N} u_i(A_j); \forall\ k \in A_i \end{array}\right\}$$

Next, A satisfies PROP1-E if it is PROP-E up to an item, i.e.,

$$\left.\begin{array}{l} u_i(A_i \cup \{k\}) \geq \frac{1}{n} \sum_{j \in N} u_i(A_j); \exists\ k \in \{M \setminus A_i\} \textit{ or,} \\ u_i(A_i \setminus \{k\}) \geq \frac{1}{n} \sum_{j \in N} u_i(A_j); \exists\ k \in A_i \end{array}\right\}$$

[2] *Beyond 2-D, one must include the concept of swapping bundles in EF [22, 137].*

Finally, we state the definition of MMS and its multiplicative approximation.

Definition 3.5 (*Maxmin Share MMS [51]*)**.** *An allocation A is said to be MMS if $\forall i \in N, u_i(A_i) \geq \mu_i$, where*

$$\mu_i = \max_{(A_1,A_2,\ldots,A_n)\in\prod_n(M)} \min_{j\in N} u_i(A_j)$$

An allocation A is said to be α-MMS if it guarantees $u_i(A_i) \geq \alpha \cdot \mu_i$ for $\mu_i \geq 0$ and $u_i(A_i) \geq \frac{1}{\alpha} \cdot \mu_i$ when $\mu_i \leq 0$, where $\alpha \in (0, 1]$.

Definition 3.6 (*Pareto-Optimal (PO)*)**.** *An allocation A is PO if $\nexists\ A'$ s.t., $\forall i \in N$, $u_i(A'_i) \geq u_i(A_i)$ and $\exists i \in N$, $u_i(A'_i) > u_i(A_i)$.*

We also consider efficiency notions like Maximum Utilitarian Welfare (MUW), that maximizes the sum of agent utilities. Maximum Nash Welfare (MNW) maximizes the product of agent utilities, and Maximum Egalitarian Welfare (MEW) maximizes the minimum agent utility.

In the next section, we define a transformation from 2-D to 1-D that plays a major role in adaptation of existing algorithms for ensuring desirable properties.

3.3 Reduction from 2-D to 1-D

We define a transformation $\mathfrak{T} : \mathcal{V} \rightarrow \mathcal{W}$, where $\mathcal{V}$ is the valuations in 2-D, i.e., $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$ and $\mathcal{W}$ is the valuations in 1-D, i.e., $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$.

Definition 3.7 (*Transformation $\mathfrak{T}$*)**.** *Given a resource allocation problem $(N, M, \mathcal{V})$ we obtain the corresponding 1-D valuations denoted by $\mathcal{W} = \mathfrak{T}(\mathcal{V}(\cdot))$ as follows,*

$$\forall i \in N, w_i(A_i) = \mathfrak{T}(V_i(A_i)) = v_i(A_i) + v'_i(A_{-i}) - v'_i(M) \tag{3.1}$$

When valuations are additive, we obtain $w_i(A_i) = v_i(A_i) - v'_i(A_i)$. An agent's utility in 2-D is $u_i(A_i)$ and the corresponding utility in 1-D is $w_i(A_i)$.

Lemma 3.1. *For goods (chores), under monotonicity (anti-monotonicity) of $\mathcal{V}$, $\mathcal{W} = \mathfrak{T}(\mathcal{V}(\cdot))$ is normalized, monotonic (anti-monotonic), and non-negative (negative).*

Proof. We assume the monotonicity of utility for goods in 2-D. Therefore, $\forall S \subseteq M, w_i(S)$ is also monotone. And $w_i(\emptyset) = v_i(\emptyset) + v'_i(M) - v'_i(M) = 0$ is normalized. Since $w_i(\cdot)$ is monotone and normalized, it is non-negative for goods. Similarly, we can prove that $w_i(\cdot)$ is normalized, anti-monotonic, and non-negative for chores. □

Theorem 3.1. *An allocation A is $\mathfrak{F}$-Fair and $\mathfrak{E}$-Efficient in $\mathcal{V}$* iff *A is $\mathfrak{F}$-Fair and $\mathfrak{E}$-Efficient in the transformed 1-D, $\mathcal{W}$, where $\mathfrak{F} \in \{$EF, EF1, EFX, PROP-E, PROP1-E, PROPX-E, MMS$\}$ and $\mathfrak{E} \in \{$PO, MUW$\}$.*

Proof Sketch. We first consider $\mathfrak{F} =$ EF. Let allocation A be EF in $\mathcal{W}$ then,

$$\forall i,\ \forall j,\ w_i(A_i) \geq w_i(A_j)$$
$$v_i(A_i) + v'_i(A_{-i}) - v'_i(M) \geq v_i(A_j) + v'_i(A_{-j}) - v'_i(M)$$
$$u_i(A_i) \geq u_i(A_j)$$

We can similarly prove the rest.

From Lemma 3.1 and Theorem 3.1, we obtain the following.

Corollary 3.1. *To determine $\{$EF, EF1, EFX, MMS$\}$ fairness and $\{$PO, MUW$\}$ efficiency, we can apply existing algorithms to the transformed $\mathcal{W} = \mathfrak{T}(V(\cdot))$ for general valuations.*

Existing algorithms cannot be directly applied.

We state a few examples below. Modified *leximin* algorithm gives PROP1 and PO for chores for 3 or 4 agents in [60], but it is not PROP1-E (or PROP1) and PO in 2-D when applied on utilities. The following example demonstrates the same,

Example 3.1. Consider 3 agents $\{1, 2, 3\}$ and 4 chores $\{c_1, c_2, c_3, c_4\}$ with positive externality. The 2-D valuation profile is as follows, $V_{1c_1} = (-30, 1)$, $V_{1c_2} = (-20, 1)$,

$V_{1c_3} = (-30, 1)$, $V_{1c_4} = (-30, 1)$, $V_{3c_1} = (-1, 40)$, $V_{3c_2} = (-1, 40)$, $V_{3c_3} = (-1, 40)$, and $V_{3c_4} = (-1, 40)$. The valuation profile of agent 2 is identical to agent 1. Allocation $\{\emptyset, \emptyset, (c_1, c_2, c_3, c_4)\}$ is leximin allocation, which is not PROP1-E. However, allocation $\{c_3, (c_2, c_4), (c_1)\}$ is leximin on transformed valuations; it is PROP1 and PO in $\mathcal{W}$ and it is PROP1-E and PO in $\mathcal{V}$.

For chores, the authors in [99] showed that any PROPX allocation ensures 2-MMS for symmetric agents. This result also doesn't extend to 2-D. For e.g., consider two agents $\{1, 2\}$ having additive identical valuations for six chores $\{c_1, c_2, c_3, c_4, c_5, c_6\}$, given as $V_{1_{c1}} = (-9, 1)$, $V_{1_{c2}} = (-11, 1)$, $V_{1_{c3}} = (-12, 1)$, $V_{1_{c4}} = (-13, 1)$, $V_{1_{c5}} = (-9, 1)$, and $V_{1_{c6}} = (-1, 38)$. Allocation $A = \{(c_1, c_2, c_3, c_4, c_5), (c_6)\}$ is PROPX-E, but is not 2-MMS in $\mathcal{V}$. Further, adapting certain fairness or efficiency criteria to 2-D is not straightforward. E.g., MNW cannot be defined in 2-D because agents can have positive or negative utilities. Hence MNW implies EF1, and PO doesn't extend to 2-D. The authors proved that MNW allocation gives at least $\frac{2}{1+\sqrt{4n-3}}$-MMS value to each agent in [55], which doesn't imply for 2-D. Similarly, we show that approximation to MMS, α-MMS, does not exist in the presence of externalities (Section 3.5).

3.4 Proportionality in 2-D

We remark that PROP (Def. 3.2) is too strict in 2-D. As a result, we introduce PROP-E and its additive relaxations in Defs. 3.3 and 3.4 for general valuations.

Proposition 3.1. *For additive 2-D, we can adapt the existing algorithms of PROP and its relaxations to 2-D using $\mathfrak{T}$.*

Proof. In the absence of externalities, for additive valuations, PROP-E is equivalent to PROP. From Theorem 3.1, we know that $\mathfrak{T}$ retains PROP-E and its relaxations, and hence all existing algorithms of 1-D is applicable using $\mathfrak{T}$. $\square$

It is known that EF $\implies$ PROP for sub-additive valuation in 1-D. In the case of PROP-E, $\forall i, j \in N$, $u_i(A_i) \geq u_i(A_j) \implies u_i(A_i) \geq \frac{1}{n} \cdot \sum_{j=1}^{n} u_i(A_j)$.

Corollary 3.2. *EF $\implies$ PROP-E for arbitrary valuations with full externalities.*

We now compare PROP-E with existing PROP extensions for capturing externalities. We consider two definitions stated in literature from [125] (Average Share) and [22] (General Fair Share). Note that both these definitions are applicable when agents have additive valuations, while PROP-E applies for any general arbitrary valuations. In [22], the authors proved that Average Share $\implies$ General Fair Share, i.e., if an allocation guarantees all agents their average share value, it also guarantees general fair share value. With that, we state the definition of Average Share (in 2-D) and compare it with PROP-E.

Definition 3.8 (Average Share [125])**.** *In $\mathcal{V}$, the average value of item k for agent i, denoted by $avg[v_{ik}] = \frac{1}{n} \cdot [v_{ik} + (n-1)v'_{ik}]$. The average share of agent i, $\overline{v_i(M)} = \sum_{k \in M} avg[v_{ik}]$. An allocation A is said to ensure average share if $\forall i, u_i(A_i) \geq \overline{v_i(M)}$.*

Proposition 3.2. *PROP-E is equivalent to Average Share in 2-D, for additive valuations.*

Proof Sketch. $\forall i \in N$,

$$\begin{aligned} u_i(A_i) \geq \frac{1}{n} \cdot \sum_{j \in N} u_i(A_j) &= \frac{1}{n} \cdot \sum_{j \in N} v_i(A_j) - v'_i(M \setminus A_j) \\ &= \frac{1}{n} \cdot \sum_{k \in M} v_{ik} - \frac{1}{n} \cdot \sum_{k \in M} (n-1)v'_{ik} \end{aligned}$$

Next, we briefly state the relation of EF, PROP-E, and Average Share beyond 2-D and omit the details due to space constraints.

Remark 3.1. *In case of full externality, EF $\not\Longrightarrow$ Average Share [22].*

Proposition 3.3. *Beyond 2-D, PROP-E $\not\Longrightarrow$ Average Share and Average Share $\not\Longrightarrow$ PROP-E.*

To conclude this section, we state that for the special case of 2-D externalities with additive valuations, we can adapt existing algorithms to 2-D, and further analysis is required for the general setting.

We now provide analysis of MMS for 2-D valuations in the next section.

3.5 Approximate MMS in 2-D

From Theorem 3.1, we showed that transformation $\mathfrak{T}$ retains MMS property, i.e., an allocation A guarantees MMS in 1-D *iff* A guarantees MMS in 2-D. We draw attention to the point that,

$$\mu_i = \mu_i^{\mathcal{W}} + v_i'(M) \tag{3.2}$$

where $\mu_i^{\mathcal{W}}$ and μ_i are the MMS value of agent i in 1-D and 2-D, respectively. [118] proved that MMS allocation may not exist even for additive valuations, but α-MMS always exists in 1-D. The current best approximation results on MMS allocation are $3/4+1/(12n)$-MMS for goods [78] and 11/9-MMS for chores [90] for additive valuations. We are interested in finding multiplicative approximation to MMS in 2-D. Note that we only study α-MMS for complete goods or chores in 2-D, as [96] proved the non-existence of α-MMS for combination of goods and chores in 1-D. From Eq. 3.5 for $\alpha \in (0, 1]$, if μ_i is positive, we consider α-MMS, and if it is negative, then $1/\alpha$-MMS.

We categorize externalities in two ways for better analysis 1) Correlated Externality 2) Inverse Externality. In the correlated setting, we study goods with positive externality and chores with negative externality. And in the inverse externality, we study goods with negative externality and chores with positive externality. Next, we investigate α-MMS guarantees for correlated externality.

3.5.1 α-MMS for Correlated Externality

Proposition 3.4. *For correlated externality, if an allocation A is α-MMS in $\mathcal{W}$, A is α-MMS in $\mathcal{V}$, but need not vice versa.*

Proof. **Part-1.** Let A be α-MMS in $\mathcal{W}$,

$$\forall i \in N, w_i(A_i) \geq \alpha\mu_i^{\mathcal{W}} \quad \Longrightarrow \quad u_i(A_i) - v_i'(M) \geq -\alpha v_i'(M) + \alpha\mu_i \quad \textit{for goods}$$
$$\forall i \in N, w_i(A_i) \geq \frac{1}{\alpha}\mu_i^{\mathcal{W}} \quad \Longrightarrow \quad u_i(A_i) - v_i'(M) \geq -\frac{1}{\alpha}v_i'(M) + \frac{1}{\alpha}\mu_i \quad \textit{for chores}$$

For goods with positive externalities, $\mu_i > 0$, $\alpha \in (0,1]$, and $\forall S \subseteq M$, $v_i'(S) \geq 0$. We derive $v_i'(M) \geq \alpha v_i'(M)$, and hence we can say $u_i(A_i) \geq \alpha\mu_i$. For chores with negative externalities, $\mu_i < 0$, $1/\alpha \geq 1$, and $\forall S \subseteq M, v'(S) \leq 0$. Similarly to the previous point, we derive $v'(M) \geq \frac{1}{\alpha}v'(M)$ and thus $u_i(A_i) \geq \frac{1}{\alpha}\mu_i$.

Part-2. We now prove A is α-MMS in $\mathcal{V}$ but not in $\mathcal{W}$.

Example. Consider $N = \{1, 2\}$ both have additive identical valuations for goods $\{g_1, g_2, g_3, g_4, g_5, g_6\}$, $V_{ig_1} = (0.5, 0.1)$, $V_{ig_2} = (0.5, 0.1)$, $V_{ig_3} = (0.3, 0.1)$, $V_{ig_4} = (0.5, 0.1)$, $V_{ig_5} = (0.5, 0.1)$, and $V_{ig_6} = (0.5, 0.1)$. After transformation, we get $\mu_i^{\mathcal{W}} = 1$ and in 2-D $\mu_i = 1.6$. Allocation, $A = \{\{g_1\}, \{g_2, g_3, g_4, g_5, g_6\}\}$ is 1/2-MMS in $\mathcal{V}$, but not in $\mathcal{W}$. Similarly, it is easy to verify the same for chores. □

Corollary 3.3. *We can adapt the existing α-MMS algorithms using $\mathfrak{T}$ for correlated externality for* general valuations*.*

Corollary 3.4. *For correlated 2-D externality, we can always obtain* $3/4 + 1/(12n)$*-MMS for goods and* $11/9$*-MMS for chores for* additive*.*

3.5.2 α-MMS for Inverse Externality

Motivated by the example given in [118] for non-existence of MMS allocation for 1-D valuations, we adapted it to construct the following instance in 2-D to prove the impossibility of α-MMS in 2-D. We show that for any $\alpha \in (0, 1]$, an α-MMS or $1/\alpha$-MMS allocation may not exist for inverse externality in this section. We construct an instance V^g such that α-MMS exists in V^g only if MMS allocation exists in $W = \mathfrak{T}(V^g)$. Note that W is exactly the instance of the example in [118]. Hence the contradiction.

<u>*Non-existence of α-MMS in Goods.*</u>

Consider the following example.

Example 3.2. We consider a problem of allocating 12 goods among three agents and represent valuation profile as V^g. The valuation profile V^g is equivalent to $10^3 \times V$ given in Table 3.1. We set $\epsilon_1 = 10^{-4}$ and $\epsilon_2 = 10^{-3}$. We transform these valuations in 1-D using $\mathfrak{T}$, and the valuation profile $\mathfrak{T}(V^g)$ is the same as the instance in [118] that proves the non-existence of MMS for goods. Note that $\forall i, v_i'(M) = -4055000 + 10^3\epsilon_1$ The MMS value of every agent in $\mathfrak{T}(V^g)$ is 4055000 and from Eq. 3.2, the MMS value of every agent in V^g is $10^3\epsilon_1$.

Recall that $\mathfrak{T}$ retains MMS property (Theorem 3.1) and thus we can say that MMS allocation doesn't exist in V^g.

Lemma 3.2. *There is no α-MMS allocation for the valuation profile V^g of Example 3.2 for any $\alpha \in [0, 1]$.*

Proof. An allocation A is α-MMS for $\alpha \geq 0$ *iff* $\forall i, u_i(A_i) \geq \alpha\mu_i \geq 0$ when $\mu_i > 0$. Note that the transformed valuations $w_i(A_i) = \mathfrak{T}(V_i^g(A_i))$. From Eq. 3.1, $u_i(A_i) \geq 0$, *iff* $w_i(A_i) \geq -v_i'(M)$, which gives us $w_i(A_i) \geq 4055000 - 0.1$. For this to be true, we need $w_i(A_i) \geq 4055000$ since $\mathfrak{T}(V^g)$ has all integral values. We know that such an allocation doesn't exist [118]. Hence for any $\alpha \in [0, 1]$, α-MMS does not exist for V^g. $\square$

<u>*Non-existence of $1/\alpha$-MMS in Chores.*</u> Consider the following example.

Example 3.3. We consider a problem of allocating 12 chores among three agents. The valuation profile V^c is equivalent to -10^3V given in Table 3.1. We set $\epsilon_2 = -10^{-3}$. We transform these valuations in 1-D, and $\mathfrak{T}(V^c)$ is the same as the instance in [21] that proves

Table 3.1: Additive 2-D Valuation Profile (V)

Item	Agent 1 (v_1, v_1')	Agent 2 (v_2, v_2')	Agent 3 (v_3, v_3')
k_1	($3\epsilon_2$, -1017+$3\epsilon_1$-$3\epsilon_2$)	($3\epsilon_2$, -1017+$3\epsilon_1$-$3\epsilon_2$)	($3\epsilon_2$, -1017+$3\epsilon_1$-$3\epsilon_2$)
k_2	($2\epsilon_1$, -1025+$2\epsilon_1$+ϵ_2)	($2\epsilon_1$, -1025+$2\epsilon_1$+ϵ_2)	($1025 - \epsilon_1$, -ϵ_1)
k_3	($2\epsilon_1$, -1012+$2\epsilon_1$+ϵ_2)	($1012 - \epsilon_1$, -ϵ_1)	($2\epsilon_1$, -1012+$2\epsilon_1$+ϵ_2)
k_4	($2\epsilon_1$, -1001+$2\epsilon_1$+ϵ_2)	($1001 - \epsilon_1$, -ϵ_1)	($1001 - \epsilon_1$, -ϵ_1)
k_5	($1002 - \epsilon_1$, -ϵ_1)	($2\epsilon_1$, -1002+$2\epsilon_1$+ϵ_2)	($1002 - \epsilon_1$, -ϵ_1)
k_6	($1022 - \epsilon_1$, -ϵ_1)	($1022 - \epsilon_1$, -ϵ_1)	($1022 - \epsilon_1$, -ϵ_1)
k_7	($1003 - \epsilon_1$, -ϵ_1)	($1003 - \epsilon_1$, -ϵ_1)	($2\epsilon_1$, -1003+$2\epsilon_1$+ϵ_2)
k_8	($1028 - \epsilon_1$, -ϵ_1)	($1028 - \epsilon_1$, -ϵ_1)	($1028 - \epsilon_1$, -ϵ_1)
k_9	($1011 - \epsilon_1$, -ϵ_1)	($2\epsilon_1$, -1011+$2\epsilon_1$+ϵ_2)	($1011 - \epsilon_1$, -ϵ_1)
k_{10}	($1000 - \epsilon_1$, -ϵ_1)	($1000 - \epsilon_1$, -ϵ_1)	($1000 - \epsilon_1$, -ϵ_1)
k_{11}	($1021 - \epsilon_1$, -ϵ_1)	($1021 - \epsilon_1$, -ϵ_1)	($1021 - \epsilon_1$, -ϵ_1)
k_{12}	($1023 - \epsilon_1$, -ϵ_1)	($1023 - \epsilon_1$, -ϵ_1)	($2\epsilon_1$, -1023+$2\epsilon_1$+ϵ_2)

the non-existence of MMS for chores. Note that $v_i'(M) = 4055000 - 10^3\epsilon_1$. The MMS value of every agent in $\mathfrak{T}(V^c)$ and V^c is -4055000 and $-10^3\epsilon_1$, respectively.

Lemma 3.3. *There is no $1/\alpha$-MMS allocation for the valuation profile V^c of Example 3.3 with $\epsilon_1 \in (0, 10^{-4}]$ for any $\alpha > 0$.*

Proof. An allocation A is $1/\alpha$-MMS for $\alpha > 0$ *iff* $\forall i, u_i(A_i) \geq \frac{1}{\alpha}\mu_i$ when $\mu_i < 0$. We set $\epsilon_1 \leq 10^{-4}$ in V^c. When $\alpha \geq 10^3\epsilon_1$ $\forall i$ then $u_i(A_i) \geq -1$. From Eq. 3.1, $u_i(A_i) \geq -1$ *iff* $w_i(A_i) \geq -4055001 + 10^3\epsilon_1$. Note that $0 < 10^3\epsilon_1 \leq 0.1$ and since $w_i(A_i)$ has only integral values, we need $\forall i, w_i(A_i) \geq -4055000$. Such A does not exist [21]. As ϵ_1 decreases, $1/\epsilon_1$ increases, and even though approximation guarantees weakens, it still does not exist for V^c. $\square$

From Lemma 3.2 and 3.3 we conclude the following theorem,

Theorem 3.2. *There may not exist α-MMS for any $\alpha \in [0,1]$ for $\mu_i > 0$ or $1/\alpha$-MMS allocation for any $\alpha \in (0,1]$ for $\mu_i < 0$ in the presence of externalities.*

Interestingly, in 1-D, α-MMS's non-existence is known for α value close to 1 [118, 71], while in 2-D, it need not exist even for $\alpha = 0$. It follows because α-MMS could not lead to any relaxation in the presence of inverse externalities. Consider the situation of goods having negative externalities, where MMS share μ_i comprises of the positive value from the assigned bundle A_i and negative value from the unassigned bundles A_{-i}. We re-write μ_i as follows, $\mu_i = \mu_i^+ + \mu_i^-$ where μ_i^+ corresponds to utility from assigned goods/unassigned chores and μ_i^- corresponds to utility from unassigned goods/assigned chores. When $\mu_i \geq 0$, applying $\alpha\mu_i$ is not only relaxing positive value $\alpha\mu_i^+$, but also requires $\alpha\mu_i^-$ which is stricter than μ_i^- since $\mu_i^- < 0$. Hence, the impossibility of α-MMS in 2-D. Similar argument holds for chores. Next, we explore MMS relaxation such that it exists in 2-D.

3.5.3 Re-defining Approximate MMS

In this section, we define *Shifted α-MMS* that guarantees a fraction of MMS share shifted by certain value, such that it always exist in 2-D. We also considered intuitive ways of approximating MMS in 2-D. These ways are based on relaxing the positive value obtained from MMS allocation μ^+ and the negative value μ^-, $\mu = \mu^+ + \mu^-$. In other words, we look for allocations that guarantee $\alpha\mu^+$ and $(1+\alpha)$ or $1/\alpha$ of μ^-. Unfortunately, such approximations may not always exist. We skip the details due to space constraints.

Definition 3.9 (*Shifted α-MMS*)**.** *An allocation A guarantees shifted α-MMS if $\forall i \in N, \alpha \in (0,1]$*

$$\left.u_i(A_i) \geq \alpha\mu_i + (1-\alpha)v_i'(M)\right\} \qquad \textit{for goods}$$

$$\left.u_i(A_i) \geq \tfrac{1}{\alpha}\mu_i + \tfrac{\alpha-1}{\alpha}v_i'(M)\right\} \qquad \textit{for chores}$$

Proposition 3.5. *An allocation A is α-MMS in $\mathcal{W}$ iff A is shifted α-MMS in $\mathcal{V}$.*

Proof. For goods, if allocation A is shifted α-MMS, $\forall i, u_i(A_i) \geq \alpha\mu_i + (1-\alpha)v'_i(M)$. Applying $\mathfrak{T}$, we get $w_i(A_i) + v'_i(M) \geq \alpha\mu_i^{\mathcal{W}} + \alpha v'_i(M) + (1-\alpha)v'_i(M)$ which gives $w_i(A_i) \geq \alpha\mu_i^{\mathcal{W}}$. For chores, if A is shifted $1/\alpha$-MMS, $\forall i, u_i(A_i) \geq \frac{1}{\alpha}\mu_i + \frac{(\alpha-1)}{\alpha}v'_i(M)$, which gives $w_i(A_i) \geq \frac{1}{\alpha}\mu_i^{\mathcal{W}}$. Similarly we can prove vice versa. $\square$

Corollary 3.5. *We can adapt all the existing algorithms for α-MMS in $\mathcal{W}$ to get shifted α-MMS in $\mathcal{V}$.*

We use $\mathfrak{T}$ and apply the existing algorithms and obtain the corresponding shifted multiplicative approximations.

In the next section, we examine the additive relaxation of MMS since a multiplicative approximation need not exist in the presence of externalities.

3.5.3.1 Additive Relaxation of MMS

Definition 3.10 (MMS relaxations)**.** *An allocation A that satisfies, $\forall i, j \in N$,*

$$\left.\begin{aligned} &\forall\ k \in \{M \setminus A_i\}, v_{ik} > 0, u_i(A_i \cup \{k\}) \geq \mu_i \\ &\forall\ k \in A_i, v_{ik} < 0, u_i(A_i \setminus \{k\}) \geq \mu_i \end{aligned}\right\} \qquad \textit{MMSX, MMS upto any item}$$
$$\left.\begin{aligned} &\exists\ k \in \{M \setminus A_i\}, u_i(A_i \cup \{k\}) \geq \mu_i,\ \ \textit{or,} \\ &\exists\ k \in A_i, u_i(A_i \setminus \{k\}) \geq \mu_i \end{aligned}\right\} \qquad \textit{MMS1, MMS upto an item} \tag{3.3}$$

Proposition 3.6. *From Lemma 3.1 and Theorem 3.1, we conclude that MMS1 and MMSX are preserved after transformation.*

EF1 is a stronger fairness notion than MMS1 and can be computed in polynomial time. On the other hand, PROPX might not exist for goods [20]. Since PROPX implies MMSX, it is interesting to settle the existence of MMSX for goods. Note that MMSX and Shifted α-MMS are not related. It is interesting to study these relaxations further, even in full externalities.

3.6 Conclusion

In this chapter, we conducted a study on indivisible item allocation with special externalities – 2-D externalities. We proposed a simple yet compelling transformation from 2-D to 1-D to employ existing algorithms to ensure many fairness and efficiency notions. We can adapt existing fair division algorithms via the transformation in such settings. We proposed proportionality extension in the presence of externalities and studied its relation with other fairness notions. For MMS fairness, we proved the impossibility of multiplicative approximation of MMS in 2-D, and we proposed Shifted α-MMS instead. There are many exciting questions here which we leave for future works. (i) It might be impossible to have a fairness-preserving valuation transformation for general externalities. However, what are some interesting domains where such transformations exist? (ii) What are interesting approximations to MMS in 2-D as well as in general externalities? In the next two chapters, we will deep dive into leveraging computers in solving fair resource allocation problems; first, we present a brief survey on using computers for mechanism design and next, we present our approach to using neural networks for solving fair division problems.

Chapter 4

Background for Neural Network-based Approach in Mechanism Design

In this chapter, we focus on the role of computers, particularly machine learning and AI, specifically Neural Networks, in the realms of game theory and mechanism design. The four-color theorem, which states that any map can be colored with at most four colors without neighboring regions sharing the same color, serves as a motivation for the importance of computers in solving complex problems. We discuss various examples of mechanism design problems, including optimal auctions, multi-facility location mechanisms, redistribution mechanisms, and sponsored search auctions, highlighting their limitations and challenges. It introduces the concept of automated mechanism design (AMD), which leverages machine learning and deep Neural Networks to overcome theoretical and computational limitations in mechanism design. We provide an overview of how researchers have used learning-based algorithms to go beyond these limitations and improve the design of mechanisms. In the next chapter, we will present the first study that integrates deep learning and fair resource allocation. In particular, we propose a Neural Network-based approach on approximately finding fair allocation that maximize social welfare, an NP-Hard problem.

4.1 Introduction

4.1.1 Motivation

The very famous *four-color theorem*, conjectured by Guthrie in 1853, states that any map in a plane can be colored using at-most four colors so that regions sharing a common boundary (other than a single point) do not share the same color. And over the centuries, there have been several attempts which later proved to be erroneous. In 1976, mathematicians Appel and Haken found an extremely arduous proof using a complex computer program to categorize the four-colorable maps [9]. Eventually, computers become an integral part of solving many complex problems. Over the past decade, with the success of deep Neural Networks for training computers to learn as well as make decisions, it has become an integral part of our lives. E.g., most computers (Macbooks 2020 or later) have built-in AI/ML chips.

In this chapter, we focus on the role of computers, in particular, Neural Networks for *game theory and mechanism design*. The problem of designing incentives for a game with self-interested and rational agents such that it achieves an outcome with desired objections. Even after advancement over decades in this field, there are still open fundamental questions. Recently, researchers have initiated a concurrent line of work in the AI community – *automated mechanism design* (AMD). AMD focuses on analyzing the sample complexity, formulating computational approaches, using machine learning/deep networks to optimize, and using inference tools in mechanism design. The goal is to use the recent advancement in learning theory and optimization to design mechanisms. Given theoretical and computational limitations, we move towards learning-based algorithms. We briefly describe the settings with limitations in this Section and further provide details about how researchers leveraged learning tools to go beyond limitations in this Chapter.

4.1.2 Examples

4.1.2.1 *Optimal Auctions*

One of the prominent parts of economic theory is *auction design*. Each bidder has a private valuation function over items. The goal is to maximize revenue and incentivize the auction so that bidders report their valuation truthfully. This setting is referred to as; Optimal Revenue-Maximizing Auction.

Limitations Myerson resolved the optimal auction design problem when a single item was for sale [110]. After more than 40 years of intense research, the problem is not entirely settled, even for a simple setting with two bidders and two items. The goal is to design dominant-strategy incentive compatibility (DSIC), i.e., truthful auctions. Most related work applies to the weaker Bayesian incentive compatibility (BIC) notion. In this chapter, we draw attention to Neural Network-based approaches for the automated design of optimal auctions.

4.1.2.2 *Multi-Facility Location Mechanism Design*

Given a space of locations, the goal of locating a single facility to reduce social costs is a single facility location problem without any external money. For example, locating a public school that is optimal for the town is a single-facility location problem. More complicated settings involve locating multiple facilities with minimized social cost, and such mechanisms are known as Multi-Facility Location Mechanism Design.

Limitations In the special case of a single facility location problem, where agents have single-peaked preferences, the problem reduces to choosing medians of the agents' peak (Moulin's generalized median rule), which is a strategyproof mechanism[109]. In contrast, the current literature doesn't fully characterize strategy-proof mechanisms, and recent work has shown negative worst-case results for the social cost.

4.1.2.3 *Redistribution Mechanism Design*

Consider the problem of allocating public resources among interested agents, where the goal is not to make a profit but to assign the resources to the one who values them the most. To find the agent with the highest valuation truthfully, one standard solution is to hold the VCG mechanism. However, it cannot ensure budget balance in the system while ensuring allocative efficiency (AE) and dominant strategy incentive compatibility (DSIC). That is, the net transfer of money cannot be zero. We design a Redistribution Mechanism to ensure a minimum surplus of money and AE and DSIC. The objective could be to minimize surplus in expectation or, in the worst case, these resources could be homogeneous or heterogeneous.

Limitations In the last decade, much research focused on designing an optimal redistribution mechanism (RM) that ensures the maximum possible total rebate, especially with homogeneous resources. However, designing an optimal redistribution mechanism, especially in a heterogeneous setting, is challenging and elusive.

4.1.2.4 *Sponsored Search Auctions*

Internet Advertising is a proven profitable model. In search engines, when the user hits a search query and the result page, we have a few ads related to search keywords. For a keyword, advertisers place their bids for these ad slots, and search engines compute the price for slots and allocate them accordingly. Bidders modify their bids based on the bidder's advertisement feedback, i.e., the number of impressions, click-through rates, price of slots, etc. So throughout, bidders' behavior and the number of bidders keep changing.

Limitations The sponsored search auction follows Generalized Second Price Auctions with reserved prices for each bidder (based on Myerson). Here we assume that bids depend on the type of bidders (private information), all bidders are rational, and there's common knowledge. In reality, bids depend on many parameters and are quite complex; the bidding setting is dynamic. So instead of static reserve prices for each bidder, we need to model the setting with dynamic reserve prices. Designing sponsored search auctions with dynamic reserve prices to maximize revenue is challenging.

4.1.2.5 *Neural Networks for AMD*

The literature consists of various approaches, such as analyzing sample complexity in designing revenue optimal auctions, using computation approaches towards Automated Mechanism Design, etc. In this thesis, we aim to survey the papers using a Neural Network-based approach to solving mechanism design. It is widely known that Neural Networks outperform existing approaches in finding an optimal mapping between the given input and output data. Neural Networks can learn algorithms ([94] learns Viterbi algorithm for decoding), mechanisms ([66] learns optimal auctions), or solve Mixed Integer Programs ([111]). This success has resulted even in the EconCS community using NNs to find allocation rules in optimal auctions. For example, [66] proposes learning optimal auctions via deep learning. [81] learns strategy-proof, multi-facility mechanisms that minimize expected social cost via NN. [50] integrate machine learning in a combinatorial auction to improve preference elicitation (which value queries to ask in every iteration). [140] instead use a NN to improve preference elicitation and reformulate WDP into a mixed-integer program. [129] learn to design a revenue-optimal, exact incentive-compatible mechanism while [103, 135, 102] learns optimal redistribution mechanisms and MAB through NN. In [139], the authors use NN to maximize the expected number of consumers and social welfare for public projects. Another line of work is Reinforcement Mechanism Design, such as learning dynamic price in sponsored search auctions [52, 128]. Additionally, [111] solves MIP on large-scale real-world application datasets and MIPLIB using a Neural Network that performs significantly better than the MIP solver. Given enough data, hyper-parameter tuning, and proper training, the networks are adept at learning effective transformations. First, we formally define all the required standard notations.

4.2 Preliminaries

We assume $\mathcal{F} = F_1 \times F_2, \ldots, \times F_n$ to be a known prior distribution for agents' valuations, where the distribution F_1 is over V_i. We denote $V = \Pi_{i=1}^{n} V_i$. Agent, i's valuation, is drawn independently from a distribution F_i, $v_i \sim F_i$. The social planner/auctioneer knows the distributions $\mathcal{F} = F_1 \times F_2, \ldots, \times F_n$, but does not know the agents/bidders' realized valuation v_i. We represent Neural Network as A^w, where w corresponds to network weights. The general idea is to model the mechanism design as a learning problem, where in the place of a loss function that measures error against a target label, we adopt our optimization loss function on valuations drawn from $\mathcal{F}$. We are given a parametric class of mechanism, $^w \in \mathcal{M}$, for parameters $w \in \mathbb{R}^d$ for some $d \in \mathbb{N}$, and a sample of agents' valuation profiles $S = \{v^{(1)}, \ldots, v^{(L)}\}$ drawn i.i.d. from $\mathcal{F}$. The goal is to find a mechanism that minimizes the optimization loss function among all parametric mechanisms in $\mathcal{M}$ that satisfy the respective constraints.

4.3 RegretNet: Learning Optimal Auctions using deep learning

4.3.1 Contributions

[66] provided a Neural Network approach for solving the multi-item auction design problem. They used multi-layer Neural Networks to encode auction mechanisms (allocation and payment rules), with bidder valuations being the input and allocation and payment being the output. They trained the networks using samples from the value distributions to maximize expected revenue subject to constraints for DSIC (i.e., truthful mechanism).

- Single-Bidder, Multi-items: They leveraged characterization results for IC mechanisms (Rochet's characterization) [120] and constructed the network architecture accordingly and further trained the network to minimize loss function (negated revenue). It is exactly DSIC, given Rochet's characterization.

- Multi-Bidder, Multi-items: There is no prior domain knowledge. They encode allocation and payment as networks and train over the loss function of negated expected revenue subject to minimizing expected ex-post regret (DSIC constraint) over the training valuation profiles.

They showed that using extensive experiments that Neural Network could learn auctions close to the optimal auctions in terms of revenue, allocation, and payment rule with quite a low regret. They also provided generalization bounds on expected revenue and ex-post regret based.

4.3.2 Neural Network Architecture

The bidders report their valuations (perhaps untruthfully), and an auction decides on allocating items to the bidders and charges payment to them. We denote an auction $(g; p)$ as a pair of allocation rules $g_i : V \to 2^M$ and payment rules $pi : V \to R_{\geq 0}$ (these rules can be randomized). Given bids $b = (b1, \ldots, bn) \in V$, the auction computes an allocation $g(b)$ and payments $p(b)$.

An auction is dominant strategy incentive compatible (DSIC) if each bidder's utility is maximized by reporting truthfully no matter what the other bidders report. A bidder with valuation v_i receives a utility $u_i(v_i, b) = v_i(g_i(b)) - p_i(b)$ for a report of bid profile b, $u_i(v_i, (v_i, b_{-i})) \geq u_i(v_i, (b_i, b_{-i}))$. An auction is ex-post individually rational (IR) if each bidder receives a non-zero utility, i.e. $u_i(v_i, (v_i, b_{-i})) \geq 0 \forall i \in N, v_i \in V_i, and b_{-i} \in V_{-i}$

They pose the optimal auction design problem as a learning problem, where they adopt the negated, expected revenue on sample valuations. We are given a parametric class of auctions, $A^w = (g^w, p^w) \in \mathcal{M}$. The goal is to find an auction that minimizes the negated, expected revenue $-\mathbb{E}[\sum_{i=1}^{n} p_i^w(v)]$, among all auctions in M that satisfy incentive compatibility.

Single Bidder, Multi-item. (Characterization-Based Approach)

- They construct a single-bidder, multi-item settings based on Rochet [120], i.e., they encode this characterization in Neural Network architecture. It acquires exactly DSIC, and they constrain the search space, i.e., to achieve IC. They refer to this network as RochetNet. It works for additive as well as unit demand valuations.

- Based on Rochet Characterization, the induced utility function u should satisfy the non-negative, monotone, convex Lipschitz utility function.

  $$u^{\alpha,\beta}(v) = \max\{\max_{j \in \mathcal{J}}\{\alpha_j \cdot v + \beta_j\}, 0\}$$

  , where $\forall j \in \mathcal{J}, \alpha_j \in [0,1]^m$ and $\beta_j \in \mathbb{R}$. The allocation rule for the valuation v is the gradient of this induced utility $= \delta u(v)$, i.e., α_j and the corresponding payment rule is $\delta u(v) \cdot v - u(v)$, i.e., $-\beta_j$. We aim to find the mechanism that maximizes the revenue among these classes of utilities $\mathcal{J}$.

- They model a single layer Neural Network as a utility function, where each $h_j = \alpha_j \cdot b + \beta_j$ with network parameters $w = (\alpha, \beta)$, as shown in Figure 4.1. The network is trained over the empirical loss function of negated expected utility given the training valuation samples and mechanism (g^w, p^w) is obtained.

- The authors provide an interpretation of the RochetNet architecture that the network consists of a menu of randomized allocations and prices and chooses the option from the menu that maximizes the bidder's utility based on the bid.

Multi Bidder, Multi-item.(Characterization-Free Approach)

- In this setting, the Neural Network encodes the mechanism, i.e., both the allocation and payment rules. These two components are distinct and trained simultaneously,

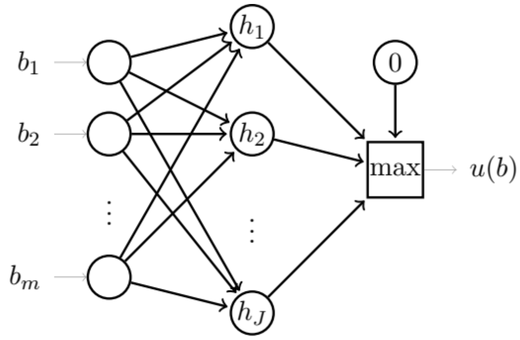

Figure 4.1: RochetNet Architecture [Image Credits: Duetting et al. [66]]

i.e., the allocation received from the allocation network uses the payment network to compute the loss function further. The goal is to maximize the revenue while ensuring IR and DSIC. The authors quantify DSIC in terms of ex-post regret, i.e., regret is the utility gain of an agent when bidding untruthfully. Exactly DSIC means ex-post regret is zero. The network searches the mechanism in large parametric space that maximizes revenue along with IR and DSIC; However, exact DSIC cannot be ensured, i.e., the network can learn approximate DSIC. This network is referred to as RegretNet and doesn't rely on any known results. Using the augmented Lagrangian method, the loss function consists of negated expected revenue and ex-post regret (DSIC) constraints.

- Fixing the bids of others, the ex-post regret for a bidder is the maximum increase in her utility, considering all possible non-truthful bids. The idea is to learn the untruthful bid that can maximize the regret and further learn the mechanism to minimize the regret and maximize revenue.

- Given this, they re-formulate the learning problem as minimizing expected negated revenue subject to the expected ex-post regret being zero for each bidder:

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{\tilde{v}\mathcal{F}}[-\sum_{i \in N} p_i^w(v)] \qquad s.t.\ \ rgt_i(w) = 0, \forall i \in N$$

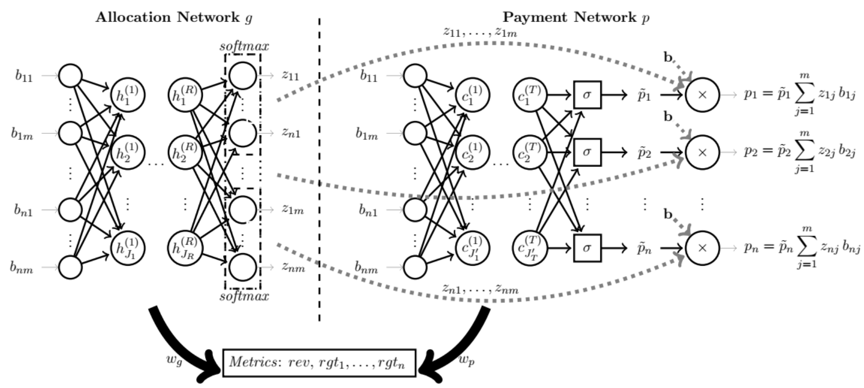

Figure 4.2: RegretNet Architecture [Image Credits: Duetting et al. [66]]

Given a sample of $\mathcal{L}$ valuation profiles from $\mathcal{F}$, they estimated the empirical ex post regret for bidder i as:

$$\hat{rgt_i}(w) = 1/L \sum_{l=1}^{L} [\max_{v_i' \in V_i} u_i^w (v_i^{(l);})]$$

and seek to minimize the empirical loss (negated revenue) subject to the empirical regret being zero for all bidders. To ensure IR, they restrict the search space to a class of parametrized auctions that charge no bidder more than her valuation for an allocation.

- In the case of additive valuation, RegretNet architecture is shown in Figure 4.2. The allocation network encodes a randomized allocation rule $g^w : R^{nm} \to [0,1]^{nm}$ and the payment network encodes a payment rule $pw : R^{nm} \to R^n_{\geq 0}$, both of which are modeled as feed-forward fully-connected networks with a tanh activation function in each of the hidden nodes. The input layer of the networks consists of bids $b_{ij} \geq 0$ representing the valuation of bidder i for item j. The allocation network outputs a vector of allocation probabilities $z_{1j} = g_{1j}(b), \ldots, z_{nj} = g_{nj}(b)$, for each item $j \in [m]$. To ensure feasibility, i.e., that the probability of an item being allocated is at most one, the allocations are computed using a softmax activation function so that for all items j, they have $\sum_{i=1} nz_{ij} \leq 1$. To accommodate the possibility of an item not being assigned, they include a dummy node in the softmax computation to hold the residual allocation probability. The payment network outputs a payment for each bidder that denotes the amount the bidder should pay in expectation for a particular bid profile. To ensure that the auction satisfies IR, i.e., does not charge a bidder more than her expected value for the allocation, the network first computes a normalized payment $\tilde{p}_i \in [0, 1]$ for each bidder i using a sigmoidal unit, and then outputs a payment $p_i = \tilde{p}_i(\sum_{j=1} mz_{ij}b_{ij})$, where the z_{ij}'s are the outputs from the allocation network.

- The authors also provide the details about network in case of Unit demand and general valuations.

4.3.3 Results

The paper showed that the Neural Network learns mechanisms with quite a low regret and achieves close to optimal auctions compared to theoretically known solutions by extensive experiments. It learns mechanisms for new settings without any analytical solution. The learned mechanisms are not only close to optimal auctions in terms of revenue but also in terms of allocation and payment rules. The auctions learned by RochetNet are exactly DSIC and match the optimal revenue precisely, with sharp decision boundaries in the allocation and payment rule. The revenue achieved by RegretNet matches the optimal revenue up to a $< 1\%$ error term, and the regret it incurs is < 0.001. For Rochet Net, they showed experiments on the setting: (A) - Single Bidder with two items having i.i.d. valuations $\sim \mathcal{U}[0, 1]$, (B) - Single Bidder with two items having i.i.d. valuations $\sim \mathcal{U}[2, 3]$, (C) - Single Bidder with m items having i.i.d. valuations $\sim \mathcal{U}[0, 1]$, where the only analytical solution available beyond $m \geq 6$ is a conjecture, and the paper showed result till $m = 10$ in the same direction as conjecture, and (D) - Single Bidder with two items having correlated valuation, where no analytical solution is available. For Regret-Net, they showed experiments on the setting: (E) - three additive bidders and ten items having i.i.d. valuations $\sim \mathcal{U}[0,1]$, and (F) - five additive bidders and ten items having i.i.d. valuations $\sim \mathcal{U}[0,1]$. There is no known solution for these two settings. They also compare their learned auctions with the LP solver solution for the setting of two bidders and three items having i.i.d. valuations $\sim \mathcal{U}[0,1]$; the LP solver took 69 hours, while their training took only 9 hours. Thus, providing strong results that Neural Networks can learn close to optimal auctions.

4.4 Multi-facility location mechanism using Deep Learning

4.4.1 Contribution

Golowich, Narasimhan, and Parkes [81] provided a Neural Network approach for solving the multi-facility location mechanism, i.e., strategic proof mechanisms that minimize the social cost. They used Neural Networks to encode multi-facility location mechanisms with a peak of agents as input and the facility's location as output. It models existing characterization results as MoulinNet and further where characterization results are unavailable as RegretNet-nm. They also show the flexibility of learning mechanisms in the case of weighted social cost and when agents having utility distribution is non-independent. They showed that their network with quite low ex-post regret, i.e., essentially strategy-proof with low or comparable social costs with the baselines.

4.4.2 Neural Network Architecture

There are two architectures proposed by [81], i.e., MoulinNet is based on the characterization of single facility location, which is extended for multi-facility location, and RegretNet-nm is characterization free with learns via constraint attached to its loss function. We first briefly explain the setting of the problem and then explain both the networks in this Section.

Let Ω be the set of locations, i.e., [0,1] and K facilities are to be located in Ω. Let $\tau(u_i)$ be the peak of the value function v_i. They assume the value of an agent as $v(x) = -\mathrm{mod}\ x - a$, where $a = \tau(u)$ is the peak. A mechanism $f : V \to \Omega$ takes input agents valuations $v = (v_1, v_2, \ldots, v_n)$, i.e., it takes agents' peak as input and outputs the locations for K facilities in Ω. They assume that the agent valuations are sampled from a joint distribution $\mathcal{D}$ over V, with full support on V. The objective is to find a mechanism f among the set of strategy-proof mechanisms for a multi-facility location that minimizes expected social cost, i.e., $-\mathbb{E}_{u\sim\mathcal{D}}[1/n \cdot \sum_{i=1}^{n} u_i(f(u))]$.

MoulinNet (Characterization-based Approach) As stated in the paper, A unanimous mechanism $f : U \to \Omega$ is strategy-proof if and only if it is a generalized median rule, i.e., for each $S \subseteq \{1, \ldots, n\}$, there exists some $a_S \in \Omega$ s.t. for all $(v_1, v_2, \ldots, v_n) \in V$,

$$f(u) = \min_{S\subseteq\{1,2,\ldots,n\}} \max\left\{\max_{i\in S}\{\tau(v_i)\}, a_S\right\}$$

The authors encode these characteristics in MoulinNet. Each a_S is parametrized using a monotone feed-forward Neural Network $h : \{-1, 1\}^n \to \mathbb{R}$ as shown in Figure 4.3. The Neural Network h maps a n-dimensional binary encoding of set S, i.e., $e(S) \in \{-1, 1\}^n$ to a real value. Hence $a_S = h(e(S))$. The Neural Network A^w, i.e., the strategy-proof single-facility mechanism, is built upon the network h as shown in Figure 4.4 that takes peaks $\tau(u_1), \ldots, \tau(u_n)$ as inputs and outputs the location of the facility. In the case of $K \geq 2$ facilities, they use K replicas of the network, and each network outputs the location of a facility. This network is strategyproof. Next, to find optimal network parameters, they train it over the loss function of negated expected social cost given the sample valuation profiles.

RegretNet-nm (Characterization-free Approach) Next, they present a general approach to learning mechanisms without characterization with low social cost and ex-post regret (strategy-proof). They optimize a feed-forward Neural Network to minimize the negated expected social cost with attached ex-post regret constraints. The idea is an ex-

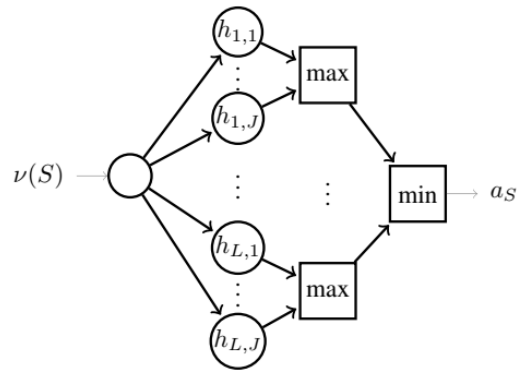

Figure 4.3: MoulinNet Architecture [Image Credits: Golowich, Narasimhan, and Parkes [81]]

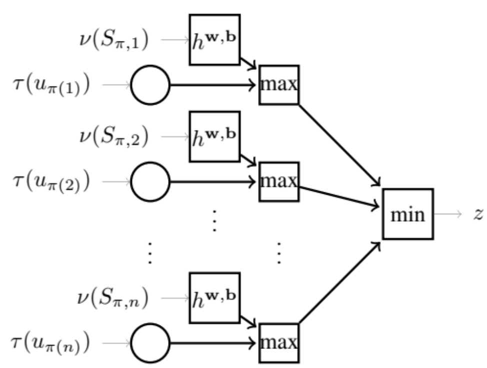

Figure 4.4: MoulinNet for single facility [Image Credits: Golowich, Narasimhan, and Parkes [81]]

tension of the paper [66], i.e., ex-post regret is defined as the maximum utility an agent can gain from misreporting their preferences.

As shown in Figure 4.5 is fully connected, takes agent peaks $\tau(u_1), \ldots, \tau(u_n)$ as input and outputs the locations of the K facilities $z_1, \ldots, z_K$ with ReLU as non-linear activation function in hidden layers. The augmented Lagrangian loss function is the expected social cost or a weighted variant subject to the expected ex-post regret being zero for all agents.

Since the optimization problem is non-convex, and the solver works with estimates of the loss and regret from training samples, the learned mechanism may have a low regret. Through extensive experiments, the authors showed that the learned mechanisms have low social costs along with negligible regret.

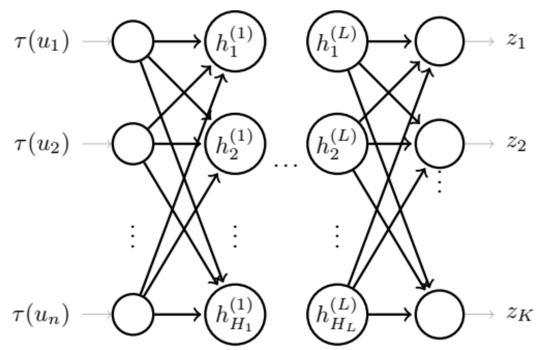

Figure 4.5: RegretNet-nm for facility location [Image Credits: Golowich, Narasimhan, and Parkes [81]]

4.4.3 Results

They performed experiments on five agents with $K = 1, 2, 3, 4, 5$ with three sets of varied settings (A) - peaks are distributed i.i.d. uniformly on [0; 1], (B) - weighted social cost and (C) peaks are sampled as non-independent variables. They compared their results across the baselines across the best percentile rule, the best dictatorial rule, and the best constant rule. For $K = 1$ both MoulinNet and RegretNet-nm nearly always place the facility very close to the median of the agents' peaks. The RegretNet-nm mechanisms have quite a low regret, i.e., they are essentially strategy-proof. They showed that Neural Networks can learn near-optimal, low ex-post regret mechanisms for the multi-facility location problem. They also showed the flexibility of the networks by experimenting with various settings to learn optimal mechanisms. RegretNet-nm and MoulinNet yield significantly smaller social costs than the baseline mechanisms. The flexibility of RegretNet-nm allows it to outperform all the benchmarks.

4.5 Summary

Neural Networks offer a promising alternative when traditional methods fail to solve complex problems. An example of this is the four-color theorem, which was solved in 1976 using a computer program to categorize four-colorable maps. Since then, computers have become essential tools for solving a wide range of complex problems. With the development of deep Neural Networks, computers can now learn and make decisions, making them an integral part of our lives. This chapter focuses on the role of Neural Networks in game theory and mechanism design, a field that has faced fundamental questions despite decades of research. To overcome these challenges, researchers have turned to automated mechanism design, which utilizes machine learning and deep networks to optimize and design mechanisms. By leveraging learning tools, researchers can go beyond theoretical and computational limitations to design mechanisms that achieve desired objectives. One challenging domain is resource allocation, and this computational approach can be used to solve complex problems in this field. Given the success of Neural Networks and theoretical limitations, we believe that combining deep learning and resource allocation is a promising area for further exploration, particularly with more complex objective functions. In the next chapter, we will discuss our approach to amalgamating deep learning and fair division.

Chapter 5

EEF1-NN: Efficient and EF1 allocations through Neural Networks

In this chapter, we will focus on allocating items to maximize efficiency while ensuring fairness. Since Envy-freeness may not always exist, we consider the relaxed notion, Envy-freeness up to one item (EF1) that is guaranteed to exist. We add the further constraint of maximizing efficiency, utilitarian social welfare (USW) among fair allocations. In general, finding USW allocations among EF1, i.e., EEF1, is an NP-Hard problem even for additive valuations. Neural networks (NNs) have shown state-of-the-art performance in designing optimal auctions as well as in learning algorithms. We design a NN inspired by U-Net for learning EEE1 allocations which we refer to as EEF1-NN. EEF1-NN is generic and scales to any number of agents and items once trained. We empirically demonstrate that EEF1-NN finds allocation with higher USW and ensures EF1 with a high probability for different distributions over input valuations.

5.1 Introduction

Consider a situation where a social planner needs to allocate a set of indivisible items (goods or/and chores) among interested agents. Agents have valuations for the items, i.e., an item might be a *good* – positive valuation for one while it might be a *chore* – negative valuation for the other. The agents reveal their valuations upfront to the *social planner*. The social planner is responsible for the fair and efficient allocation of these items among the agents. For example, a Government needs to distribute resources and delegate tasks amongst its subdivisions. The subdivisions should not feel mistreated in the system. While ensuring this, the Government would like to optimally allocate items for the system's growth.

Fair division is well-explored in literature [43, 106, 119]. Economists have proposed various fairness and efficiency notions applicable in real-world settings, such as division of investments and inheritance, vaccines, tasks, etc. There are web-based applications such as Spliddit, The Fair Proposals System, Coursematch, Divide Your Rent Fairly, etc., used for credit assignment, land allocation, division of property, course allocation, and even task allotment. All these applications assure certain fairness and efficiency guarantees.

One of the most popular fairness criteria is envy-freeness (EF) [72]. An allocation is envy-free if each agent values its share at least as much as they value any other agent's share. EF is also trivially satisfied by allocating empty bundles to every agent. Hence we must also have efficiency guarantees. When we consider a complete allocation of indivisible items, EF may not exist (two agents, one good). Finding whether EF exists or not is known to be Δ_2^pcomplete [42], let alone finding an efficient allocation among EF. To overcome this limitation, we consider a prominent relaxation of EF - EF1 (Envy-freeness up to one item) [51]. Unlike EF, EF1 always exists and can be computed in polynomial time [101].

In this work, we focus on *utilitarian* social welfare (USW), i.e., the sum of utilities of individual agents. When valuations are additive, finding allocation that maximizes USW (MUW) is polynomial-time solvable. While finding EF1 or MUW allocations are polynomial-time solvable, maximizing USW within EF1 allocations, i.e., *EEF1; efficient and envy-free up to one item*, is an NP-hard problem, even when valuations are additive for two agents [16, 29]. There is no known approximation algorithm for EEF1. With these theoretical limitations, we propose a data-driven learning approach, i.e., given the agents' valuations, we aim to learn EEF1. It is widely known that *Neural Networks (NNs)* outperform existing approaches in finding an optimal mapping between the given input and output data. Neural Networks can learn algorithms [94], mechanisms [66] or solve Mixed Integer Programs [111]. Motivated by the success of Neural Networks, we aim to learn algorithms for EEF1 using Neural Networks. We list our major challenges as follows,

Challenges. (i) In the existing integration of Neural Networks and mechanism design, payments are at their disposal. In our work, there are no payments, and we learn discrete allocations, i.e., our solution space is binary. Whereas the output of Neural Networks is real numbers, it can easily learn optimal fractional allocations. If we convert fractional solutions to integral, fairness guarantees no longer hold. (ii) Further, we aim to design a generalized network for any number of agents or items, even for configurations not seen during training. Most of the existing Neural Network-based approaches in EconCS train the models separately for each configuration [66, 103]. We overcome the above challenges as described below.

Contributions. To the best of our knowledge, this is the first study that integrates deep learning and fair resource allocation. In particular,

- We propose a Neural Network *EEF1-NN* inspired by U-Net to learn EEF1.
- We transform our valuations and augment them with additional channels to enhance the network's performance.
- We use a series of convolutional and up-convolutional layers to learn EEF1; EEF1-NN is generalized for any number of agents and items.

- We sample valuations from various distributions and report the expected fairness and efficiency achieved. Even for large instances, our network performs well. Moreover, [1] the network quickly computes the output; hence we can improvise this approach to be adept in practical real-time applications.
- We show that, for our setting, bagging of networks improves performance.

5.2 Related Work.

Fair resource allocation is well studied in the literature across various fairness and efficient notions [43, 106, 119]. When a definition of fairness is too strong or may not exist, we always look for its relaxation/approximation. There is existing work that provides approximate efficiency and fairness guarantees in [7, 32, 49, 118]. In this paper, we majorly focus on EEF1. Authors in [49] presents a framework to compute ϵ-Efficient and $\mathcal{F}$-Fair allocation, using parametric integer linear programming, which is double exponential in terms of n and m. They explored group Pareto Efficiency, which is equivalent to USW. Authors in [16] provides a pseudopolynomial-time algorithm to find MUW within EF1, which is exponential in n and polynomial in m and V, where V bounds the valuation per item.

EF1 allocations always exist and can be found in polynomial time even for general valuations [13, 55, 101]. Finding MUW allocations is also polynomial-time solvable for additive valuations, i.e., we iterate over items assign the item to the agent who values it the most. However, finding MUW allocation amongst EF1 allocations is NP-hard even for two agents with additive valuations. Also finding a truthful way for allocating EF1 is also challenging [115].

There is always a trade-off between fairness and efficiency, corresponding to the study of the price of fairness ([27, 36]). Researchers have also studied how likely a fairness notion

[1] We evaluated IP solver to solve maximizing USW w.r.t. EF1 constraints, and for 10x100, it was taking several minutes.

will not exist ([65, 104, 105]). When agents' valuations are additive and drawn randomly from a uniform distribution, EF exists with a high probability when m is at least $\Omega(n \log n)$ and can be obtained by MUW allocations proved by [65]. However, the hidden constants might be high[2], and it leaves scope to explore. [104] show that RR is envy-free when $m \geq \Omega(n \log n / \log \log n)$.

Recently the EconCS community has been interested in learning mechanisms/algorithms using Neural Network, especially in a setting of theoretical limitations [66, 81, 103, 94, 130, 135, 139]. Researchers have studied mechanism design widely [76, 75]. [66] and [81] learns optimal auctions and multi-facility mechanism using Neural Network. [140] uses Neural Network in the combinatorial auction for preference elicitation. [103, 135, 102] learns optimal redistribution mechanisms and MAB through Neural Network. [139] uses Neural Network to maximize the expected number of consumers and the expected social welfare for public projects. Additionally, [111] solves MIP on large-scale real-world application datasets and MIPLIB using a Neural Network that performs significantly better than the MIP solver. [114] proposed a Neural Network-based solution to achieve fairness in classification. Given enough data, hyper-parameter tuning, and proper training, the networks are adept at learning effective transformations. Another line of work is Reinforcement Mechanism Design, such as learning dynamic price in sponsored search auctions [52, 128].

5.3 Preliminaries

We consider the problem of allocating $M = [m]$ indivisible items among $N = [n]$ interested agents, where $m, n \in \mathbb{N}$. We only allow complete allocation and no two agents can receive the same item. That is, $A = (A_1, \ldots, A_n)$, $A \in \Pi_n(M)$ s.t., $\forall i, j \in N$, $i \neq j; A_i \cap A_j = \emptyset$ and $\bigcup_i A_i = M$. Each agent $i \in N$ has a valuation function $v_i : 2^M \to \mathbb{R}$ and $v_i(S)$ is its valuation for a $S \subseteq M$ s.t. $v_i(\emptyset) = 0$. We represent valuation profile

[2] Our experiments show that in the case of uniform distribution, even for 10 agents, 150 items, the probability of MUW allocation being EF1 is less than 0.5

$v = (v_1, v_2, \ldots, v_n)$. We only consider additive valuations. The valuation of an agent $i \in N$ for bundle A_i is $v_i(A_i) = \sum_{j \in A_i} v_i(\{j\})$. For an agent i, an item $j \in M$ is a *good* if, $v_i(\{j\}) \geq 0$, and a *chore* if, $v_i(\{j\}) < 0$. We consider three settings - pure goods, pure chores, and a combination of goods and chores. With this notation, we now define fairness and efficiency properties as follows.

Definition 5.1 (*Envy-free (EF) and relaxations*). *An allocation A that satisfies $\forall i, j \in N$,*

$$v_i(A_i) \geq v_i(A_j) \text{ is EF}$$
$$v_i(A_i \setminus \{k\}) \geq v_i(A_j \setminus \{k\}); \exists k \in \{A_i \cup A_j\} \text{ is EF1}$$

Definition 5.2 (*Maximum Utilitarian Welfare (MUW)*). *An allocation A^* is said to be* efficient *or MUW if it maximizes the USW, $sw(A, v) = \sum_{i \in N} v_i(A_i)$*

$$A^* \in \underset{A \in \Pi_n(M)}{arg\,max}\ sw(A, v)$$

Definition 5.3 (*EEF1 Allocation*). *We say an allocation is* EEF1 *if it satisfies EF1 fairness and maximizes USW amongst EF1 allocations.*

Given agents' valuation profile $v = (v_1, v_2, \ldots, v_n)$, we learn EEF1 allocations using a data-driven approach. We randomly draw $v_i \sim \mathcal{F}_i$ and assume $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2, \ldots, \times \mathcal{F}_n$ to be a known prior distribution over agents' valuations. We use the notation $n \otimes m$ to represent a setting with n agents and m items.

5.4 Our Approach: EEF1-NN

We construct the optimization problem for EEF1 5.4.1; we then formulate the Lagrangian loss function 5.4.2 and provide the detail of EEF1-NN 5.4.3 and 5.4.4.

5.4.1 Optimization Problem.

We are given a set of valuation profile $v = (v_1, v_2, \ldots, v_n)$, where v_i is drawn randomly from a distribution $\mathcal{F}_i$. Among all possible allocations, we need to find an optimal A^* that maximizes USW $sw(A, v)$ and satisfies a fairness constraint. We formulate two (generalized) fairness constraints - EF and EF1 as follows.

$$ef_i(A, v) = \sum_{k \in N} \max \left\{ 0, (v_i(A_k) - v_i(A_i)) \right\} \tag{5.1}$$

$$ef1_i(A, v) = \sum_{k \in N} \max \left\{ 0, (v_i(A_k) - v_i(A_i)) + \quad \min \left\{ - \max_{j \in A_k} v_i(\{j\}), \min_{j \in A_i} v_i(\{j\}) \right\} \right\} \tag{5.2}$$

Our goal is to maximize the expected welfare w.r.t. to the expected fairness.

$$\begin{aligned} \text{minimize} \quad & - \mathbb{E}_v \left[sw(A, v) \right] = \mathbb{E}_v [\sum_{i \in N} v_i(A_i)] \\ \text{subject to} \quad & \mathbb{E}_v \left[\sum_{i \in N} ef_i(A, v) \right] = 0 \text{ or, } \mathbb{E}_v \left[\sum_{i \in N} ef1_i(A, v) \right] = 0 \end{aligned} \tag{5.3}$$

In the above optimization problem, we have 'OR' among fairness constraints, which we elaborate more on this in the Ablation Study in Section 5.5.1.

5.4.2 EEF1-NN: Lagrangian Loss Function.

EEF1-NN represents a mapping from valuation to allocation space, i.e., $\mathcal{A}^w : \mathbb{R}^{\{n \times m\}} \to \{0, 1\}^{n \times m}$, where w represents the network's parameters. To learn w, we formulate our problem to optimize welfare w.r.t. to fairness constraints in Eq. 5.3 and formulate Lagrangian loss function ($\lambda \in \mathbb{R}_{\geq 0}$). Given $\mathcal{L}$ samples of valuation profiles $(v^1, \dots, v^{\mathcal{L}})$ drawn from $\mathcal{F}$, the loss per sample (I_v^l) is,

$$Loss(I_v^l, w, \lambda) = \left[- sw(\mathcal{A}^w(I_v^l), v^l) + \lambda \frac{\sum_{i \in N} envy_i(\mathcal{A}^w(I_v^l), v^l)}{n} \right] \tag{5.4}$$

We minimize the following loss w.r.t w, $\mathbf{L}_{EEF1}(I_v^l, w, \lambda) = \frac{1}{\mathcal{L}} \sum_l Loss(v^l, w)$.

5.4.3 Network Details

We describe EEF1-NNś various components, including the input, architecture, and other training details in this section. EEF1-NN is a fully convolutional network (FCN) and processes input of varied sizes (i.e., height $\times$ width).

EEF1-NN: Input. We transform our valuations and augment with additional channels to enhance performance. We construct an input tensor of size $n \times m \times 6$, i.e., $I_v \in \mathbb{R}^{n\times m\times 6}$. The first channel is an $n \times m$ matrix of given valuations, i.e., $\forall i, j; I_v[i, j, 1] = v_i(\{j\})$. We take a matrix X of size $n \times m$ that contains valuation for items only corresponding to the agent who values it the most, and the rest are zero. We break ties arbitrarily and expand X into five channels.

$$\forall j \in M;\ X[i.j, 1] = \begin{cases} v_i(\{j\}) & \text{if } i \in argmax_i v_i(\{j\}) \\ 0 & \text{otherwise} \end{cases}$$

The next channel contains information about items indexed as $0, 5, \ldots, \lfloor m/5 \rfloor$,

$$I[i.j, 2] = \begin{cases} X[i, j, 1] & \text{if } j \in \{0, 5, 10, \ldots, \lfloor m/5 \rfloor\} \\ 0 & \text{otherwise} \end{cases}$$

The next channel contains data from the previous channel along with items indexed as $1, 6, \ldots, 1 + \lfloor m/5 \rfloor$. And so on. The last channel $I_v[i, j, 6]$ is X. We observe that single-channeled input performs sub-optimal. We study the effect of input complexity on the performance in Section 5.5.1.

EEF1-NN: Architecture. Our architecture is inspired by *U-Net architecture* [121]. U-Net is a fully convolutional network built to segment bio-medical images; it also requires assigning labels to image patches and not just classifying the image as a whole. While we are working on valuation profiles rather than images, one of the primary motivations to use U-Net is to process arbitrary size images. If we use a feed-forward fully functional Neural Network to learn fair and efficient allocations, we need a different network for each

$n \otimes m$. Moreover, just using a feed-forward functional network (multi-layer perceptron) learns EEF1 allocations for smaller values of n, but cannot learn as n increases.

EEF1-NN contains series of convolution and up-convolution layers, as given by Figure 5.1. EEF1-NN has three series of Conv-UpConv layers. The convolutional layers consist of 4 repeated 3x3 convolution, each followed by a non-linear activation function, tanh. The up-convolution layers consist of 4 repeated 3x3 up-convolution, each followed by a tanh activation. Note that we are not using maxpool or skip connections as we found that they degraded the performance. We apply softmax activation function across all agents for every item to ensure each item is allocated exactly once, i.e., $\forall j \in M \ \sum_{i \in N} \mathcal{A}_i^w(\{j\}) = 1$. The final output represents the probability with which agent i receives item j. Using an FCN, we have a generalized network for $n \otimes m$; however, learning EEF1 is not easy. We need to learn integral variables, while Neural Networks are known for learning continuous output. We describe these challenges next.

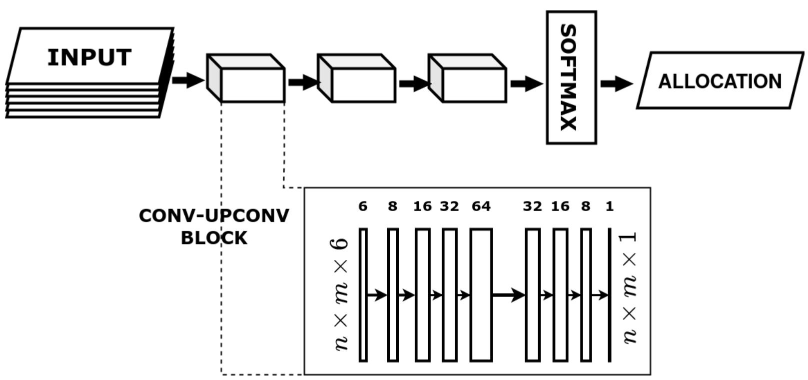

Figure 5.1: EEF1-NN Architecture

5.4.4 Training Details

Integral Allocations. The global optima of the optimization problem in Eq. 5.3 might lie in a continuous allocation setting, i.e., similar to allocating divisible items. If a network learns to distribute an item equally among all agents, then the gradient vanishes. Assigning an equal partition of each item is indeed an optimum. Converting these non-integral allocation to integral is non-trivial. Hence we set a *temperature* parameter T in the softmax layer of the network to prevent getting stuck at such optima. Let $o_j = \{o_{j_1}, \ldots, o_{j_n}\}$ denote the output of our network before the final layer. The final allocation for agent i is given by, $\mathcal{A}_i^w(\{j\}) = \text{softmax}(o_{j_i}) = \frac{e^{o_{j_i}/T}}{\sum_{k=1}^{n} e^{o_{j_k}/T}}$ which represents the probability of assigning item j to all the agents. It is common to start with a large T for initial exploration and gradually reduce T to reach the global optima. While training, when we set T to 1, we get fractional allocations. As we decrease the value of T, the network outputs allocation close to discrete. The approach we want is while training, allocation output is almost discrete, but not exactly discrete. When we keep the value too low, the output is exact discrete allocations, and there is no learning because of the vanishing gradients. We appropriately choose T based on our experiments. Once the network learns, we set the parameter low enough to ensure discrete allocations.

Inefficient Local Optima. Due to the low T value, the training of EEF1-NN is highly unstable and often gets stuck at inefficient local optima. To overcome this, we use the technique of *Bootstrap Aggregation* or Bagging. It combines the predictions from multiple classifiers to produce a single classifier. We train multiple weak networks with varied hyper-parameters on the same data set, capturing different sets of local optima. While testing, the final allocation is aggregated from these networks. We pass a test sample through all networks and select the allocation that is EF1 with maximum USW. In total, we bag seven networks with varied $\lambda \in [0.1, 2]$ for increased performance. We further analyze how Bagging affects our results in the ablation study.

We implement EEF1-NN using PyTorch. We initialize the network weights using Xavier Initialization [80]. To train, we use Adam Optimizer [95] with learning rate 0.001 for 1000 epochs with $T = 0.01$. We use a batch size of 256 samples. We sample valuations from $U[0, 1]$ (goods), $U[-1, 0]$ (chores) and $U[-1, 1]$ (combination). We sample $150k$ training data for both $10 \otimes 20$ and $13 \otimes 26$ for goods, chores, and combinations, so in total, we have $300k$ training samples, and we sample $10k$ testing samples for each setting. We train seven networks with varied $\lambda \in [0.1, 2]$ and bag them for enhanced performance. The training process takes 5-6 hours to train a single network using GPU. We are training the network for $10 \otimes 20$ and $13 \otimes 26$. however, we show our test results for various $n \otimes m$. We test for network performance for $n \in [7, 15]$. We also train an individual network over different distributions such as Gaussian, Log-normal, and Exponential. We validate EEF1-NN efficacy in the next section.

5.5 Experiments and Results

In this section, we conduct an ablation study to set appropriate hyper-parameters and 3 types of experiments showcasing performances across different item types, distributions, and scalability. To report the network performance, we define the following two metrics: the measure of fairness (probability of an allocation to be EF1) and the other of efficiency (how close our social welfare is to optimal).

Evaluation Metrics.

1. α_{EF1}^{ALG} - It measures the empirical probability with which an algorithm ALG outputs EF1 allocation. That is,

$$\alpha_{EF1}^{ALG} = \frac{\text{\#samples that ALG outputs as EF1}}{\text{\#samples outputs from ALG}}$$

   α_{EF1} is the ratio of the number of samples that are EF1 to the total number of samples.

2. β_{SW}^{ALG} - It measures the ratio of expected USW of an algorithm ALG by expected USW of MUW allocation.

$$\beta_{SW}^{ALG} = \frac{\text{Expected social welfare from ALG that outputs as EF1}}{\text{Expect social welfare of MUW}}$$

Note that $\beta_{SW}^{ALG} \in [0, 1]$ for goods, $\beta_{SW}^{ALG} \geq 1$ for chores, and will depend on the overall social welfare (positive/negative) for a combination of goods and chores. We will use that notation $(\alpha_{EF1}, \beta_{sw})$ to report performance.

5.5.1 Ablation Study

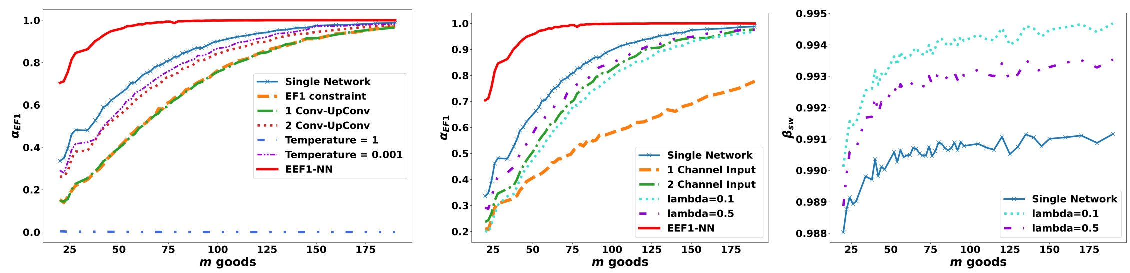

Figure 5.2: Ablation Study over varied hyper-parameters

We illustrate the effect of specific hyper-parameters in the performance of EEF1-NN in Figure 5.2. We set $n = 10$ goods for all the experiments. In the plots, the red line with the label EEF1-NN denotes the α_{EF1} for optimal parameters. Corresponding to EEF1-NN, a single network from this bagged network is labeled as *Single Network*. This *Single Network* trained with six-channeled input, $\lambda = 1$, and $T = 0.01$ is the baseline to compare across this ablation study. Only one parameter is changed w.r.t. the *Single Network* for the study.

(i) Effect of Temperature T. In Figure 5.2 (left), when $T = 1$, it converges to fractional allocation represented by the blue line at the bottom of the plot. When $T = 0.001$ (violet line), it is too low, and performs sub-optimally compared to single network. We also noticed that the performance for $T = 0.01$ and $T = 0.1$ are close to each other. We set $T = 0.01$ for all the bagged networks in EEF1-NN.

(ii) Effect of Series of Conv-UpConv layers. We select three series of Conv-UpConv for EEF1-NN as illustrated in Figure 5.2(left). As seen from Figure 5.2(left), a performance increase between 1-series (green dashed line) and 2-series (red dotted line) is significant compared to 2-series and 3-series (single network). The complexity of the network having 4-series is far more than the performance improvement.

(iii) Effect of Loss Function We empirically analyze how different envy definitions (Eq. 5.3) affect the training of EEF1-NN. As shown in Figure 5.2(left), when we train our

network using EF, i.e., Eq. 5.1 (*Single Network*, the network performs significantly better than when trained using EF1, i.e., Eq.5.2 (orange dashed line). For example, for $10 \otimes 20$, the performance of *Single Network* is (0.3358,17.9611), whereas the performance of the EF1 trained network is (0.1530,17.8708).

iv) Number of Input Channels. To enhance our network performance, we experimented with different channel inputs. For 2-channeled input, we set the first channel of input tensor as the valuation and the second to X. Like 6-channeled input, we expand X to 11 channels. As shown in Figure 5.2(right), the performance of a 1-channeled network is $(0.2113, 17.8976)$, 2-channeled network is $(0.2365, 17.8991)$, *Single Network* is $(0.3358, 17.9611)$, and 11-channeled network is $(0.3925, 17.9395)$. The network cannot be generalized for 11-channeled.

v) Effect of Bagging. We bag different combinations of networks, each trained for varied λ in Figure 5.2(right). The more the λ, the more penalty is given to envy. When λ is too small, the network learns a more efficient but less fair allocation. As we increase λ up to a certain value, the network learns less efficient but more fair allocations. We observed that varying λ results in converging to the different optimum. We bagged seven networks trained with $\lambda \in [0.1, 2]$. Bagging (EEF1-NN) outperforms the performance of a single network.

5.5.2 Experiment Details and Observations

We conduct three types of experiments, Exp1: Different kinds of resources, Exp2: Input distributions, and Exp3: Scalability. Since approaches in [16, 49] are exponential, we cannot report results on our experimentation configuration. We compare EEF1-NN with the following existing methods,

- MUW- We compare our results with MUW since we don't have EEF1. Note that EEF1-NN welfare is close to MUW; we can say it is also close to EEF1.

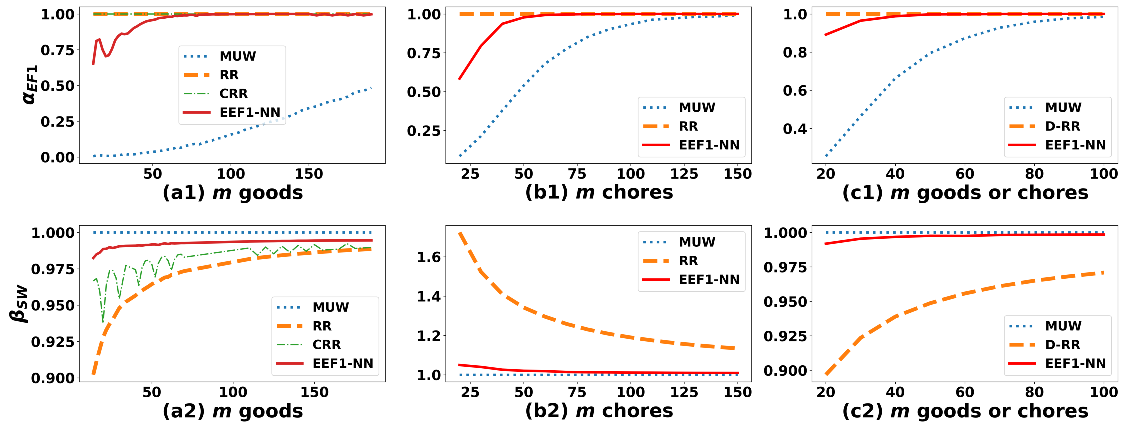

Figure 5.3: Exp1 ($n = 10$, Uniform Distribution)

- Round Robin (RR)- [55] finds EF1 for goods or chores. *Double Round Robin* (D-RR) [13] finds EF1 for the combination of goods and chores.
- Constrained Round Robin (CRR)- We implement CRR [17] to find RB sequences to increase efficiency. An RB sequence for goods.

EXP1: Performance across differed resources for Uniform Distribution. For $n = 10$, we compare α_{EF1} in Figure 5.3 (a1, b1, c1) and β_{SW} in Figure 5.3 (a2, b2, c2). As m increases, all the approaches move closer to EEF1. We observe that MUW (blue dotted line) converges towards EEF1 much faster for chores or combinations than goods. While RR converges to EEF1 much faster in goods compared to chores or combinations. We discuss this convergence in detail in Table 5.1. We observe that EEF1-NN consistently has better α_{EF1} than MUW and β_{sw} than RR/CRR. We observe that $\alpha_{EF1}^{EEF1-NN}$ is close to 1 after a certain m. At the same time, EEF1-NN is far more efficient than *RR*. (Fig 5.3 (a2,b2,c2)). Note that the CRR is only for goods. We observe that compared to CRR, EEF1-NN obtains marginally better β_{SW}, in Figure 5.3(a2).

EXP2: Performance across different distributions. We provide the performance of EEF1-NN when the valuations are sampled from different distributions such as Gaussian

(μ=0.5,σ=1) in Fig 5.4(a1, a2), Log-normal ($\mu = 0.5$,σ=1) in Fig 5.4(b1, b2), and Exponential ($\lambda = 1$) in Fig 5.4(c1, c2). We observe that in all three $\alpha_{EF1}^{EEF1-NN} > 0.99$ and $\beta_{SW}^{EEF1-NN} > 0.99$ for $m \geq 40$ in Figure 5.4.

EXP3: Scalability to larger number of agents. EEF1-NN is trained only for $10 \otimes 20$ and $13 \otimes 26$. As we have seen in the previous results and in Figure 5.5, the performance scales across varying m and n seamlessly. We provide the performance of EEF1-NN when $n = 7, 12, 14$ in Figure 5.5[3]. We observe that $\alpha_{EF1}^{EEF1-NN}$ also scales appropriately, is close to optimal, and is way better than α_{EF1}^{MUW}. We observe that there is a drop in $\alpha_{EF1}^{EEF1-NN}$ values, especially for smaller values of m as compared to the performance with $n = 10$ (Figure 5.3). It is still close to optimal and is way better than α_{EF1}^{MUW}. We observe that the $\beta_{SW}^{EEF1-NN}$ in Figure 5.5(a2, b2 c2) for goods is comparable to the performance for $n = 10$. But the efficiency drops for chores Fig 5.5(b2) although it performs better than β_{SW}^{RR}.

However, for smaller n, we can similarly train a network with fewer layers. Similarly, we observe the results for $n = 13$ for Uniform distribution. MUW allocations are likely to be EF1 when $m > 650$ for goods, i.e., $\alpha_{EF1}^{MUW} = 0.945$ for $m = 650$, when $m \geq 190$ for chores, i.e. α_{EF1}^{MUW} is more than 0.99. For goods, in Figure 5.5(a1), we observe that we get $\alpha_{EF1}^{EEF1-NN}$ almost 0.99 for $m \geq 100$. In Figure 5.5(a2), we observe that β_{EF1}^{RR} is almost 0.99 for $m \geq 200$, and β_{EF1}^{RR} is almost 0.99 for $m \geq 120$ while $\beta_{EF1}^{EEF1-NN}$ is almost 0.99 for $m \geq 30$. For chores, in Figure 5.5(b1) we observe that we get $\alpha_{EF1}^{EEF1-NN}$ almost 0.99 for $m \geq 60$. In Figure 5.5(b2), we observe β_{EF1}^{RR} is almost 1.01 for $m > 200$ while $\beta_{EF1}^{EEF1-NN}$ is almost 1.01 for $m > 100$. Similarly, we plot results for $n = 14$ goods in Figure 5.5(c1) and 5.5(c2), $\alpha_{EF1}^{EEF1-NN}$ is almost 0.99 for $m \geq 120$ and $\beta_{sw}^{EEF1-NN}$ is almost 0.99 for $m \geq 30$. For combination, in Figure 5.5(31) we observe that we get $\alpha_{EF1}^{EEF1-NN}$ almost 0.99 for $m \geq 60$.In Figure 5.5(32), we observe that β_{EF1}^{D-RR} is 0.96 for $m = 100$ while $\beta_{EF1}^{EEF1-NN}$ is almost 0.99 for $m \geq 25$.

[3] To report performance for $n \in [7, 9]$, we reduce a Conv-UpConv layer and train accordingly with $7 \otimes 14$ and $10 \otimes 20$ valuation profiles.

Analysis of Convergence to EEF1 Allocations (Uniform Distribution)

Definition 5.4 ($m^\star(n)$)**.** *For a given n, we say an algorithm converges to EEF1 allocation at $m^\star(n)$ if $\forall m > m^\star(n)$,*

(i) For goods*: $\alpha_{EF1}^{ALG} \geq 0.99$ and $\beta_{sw}^{ALG} \geq 0.99$.*

(ii) For chores*: $\alpha_{EF1}^{ALG} \geq 0.99$, and $\beta_{sw}^{ALG} \leq 1.02$.*

We empirically study $m^\star(n)$ value after which EEF1-NN, RR, and MUW start converging towards EEF1 for uniform distribution in Table 5.1. We don't report CRR in this; as we see fluctuations in β_{sw}, it doesn't increase smoothly in Figure [5.35.5]; For goods, EEF1-NN reaches close to EEF1 faster than MUW and RR, and RR reaches close to EEF1 faster than MUW. EEF1-NN converges first, then MUW, and finally RR for chores in Table 5.1. For chores, we report the value of $m^\star$ for RR when $\beta_{sw} \leq 1.064$ since m is significantly higher than MUW and RR, concluding that RR converges after a considerably larger m. As m increases, $\alpha_{EF1}, \alpha_{EFX}$, and α_{EF} of MUW gets closer. Note that we do not experiment with all possible m; the actual value of $m^\star(n)$ may be slightly different from Table 5.1. We aim to observe a pattern among approaches to achieve EEF1. For example, for 9×530 goods uniform distribution, $\alpha_{EF1} = 0.989$,$\alpha_{EFX} = 0.9834$, and $\alpha_{EF} = 0.9834$; while for 9×200 goods uniform distribution, $\alpha_{EF1} = 0.6436$, $\alpha_{EFX} = 0.5086$, and $\alpha_{EF} = 0.5032$. Note that as we do not perform experiment for all possible m, the actual value of $m^\star(n)$ may be slightly different from the exact point of convergence mentioned in the Table 5.1. Our goal is to observe a pattern among approaches to compare the different approaches to achieve EEF1.

Discussion $\alpha_{EF1}^{EEF1-NN}$ reaches 1 much faster than α_{EF1}^{MUW}, and $\beta_{sw}^{EEF1-NN}$ reaches close to β_{sw}^{MUW} much faster than RR, D-RR, CRR. EEF1-NN shows a better trade-off between EF1 and efficiency than the existing approaches for different input distributions. We also observe that even with training only on specific n and m along with goods or/and chores, the performance scales for any m and a large n. We trained our network with fixed $n \otimes m$ for goods or/and chores, it is interesting that the performance scales for any m and a

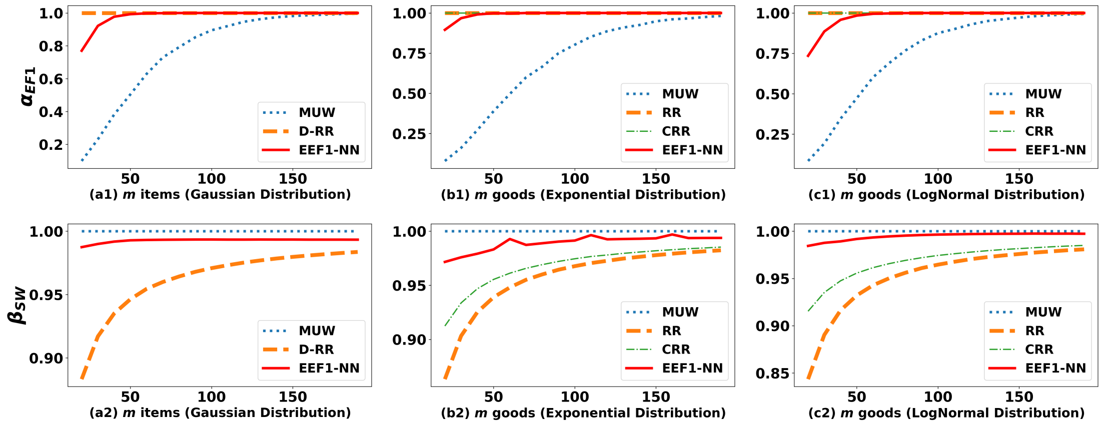

Figure 5.4: EXP2 ($n = 10$, different distributions)

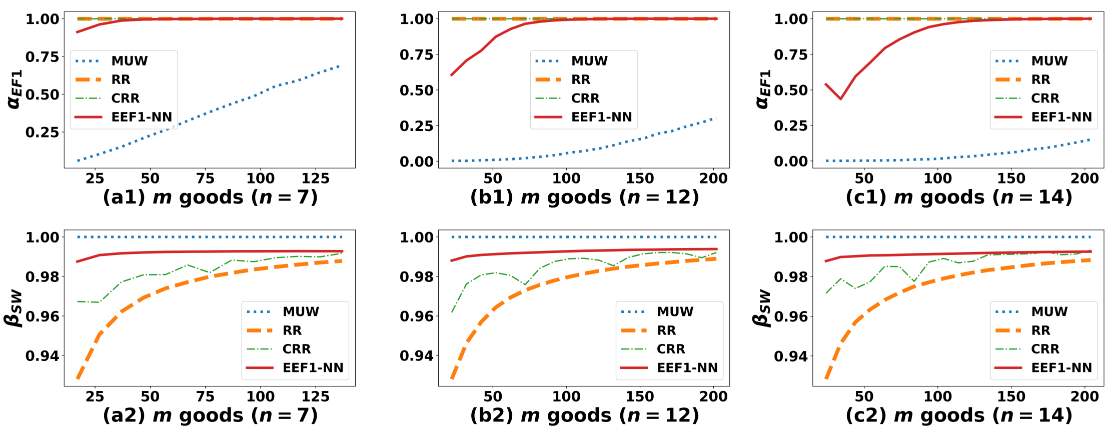

Figure 5.5: EXP3 ($n = 7, 12, 14$ goods, Uniform Distribution)

large n. For smaller n and m, one can use integer programming or any pseudo-polynomial approach. When $m >> n$, we observed that MUW converges towards EEF1 faster than RR in goods, while in chores, it's the other way around. MUW is EF1 with very high probability, which EEF1-NN also mimics it. We conclude that EEF1-NN effectively learns and provides a better trade-off when m is not too large or small compared to n but is in a specific range.

Table 5.1: Value of $m^\star(n)$ as different approaches converge to EEF1 allocations

n	(m) Goods			(m) Chores		
	EEF1-NN	RR	MUW	EEF1-NN	RR	MUW
7	38	159	380	44	195	112
8	46	172	450	44	240	120
9	57	186	530	53	295	130
10	70	196	610	60	340	148
11	82	206	660	68	400	160
12	94	214	740	75	455	167
13	110	220	840	83	505	180
14	134	228	940	87	565	190

5.6 Conclusion

In this chapter, we proposed a Neural Network EEF1-NN to find EEF1, an NP-hard problem. We designed architecture and input representation combined with other training heuristics to learn approximate EEF1 on average. We studied the effect of each proposed constituent on performance. Our experiments demonstrated the efficacy of EEF1-NN for different input distributions across various n and m over existing approaches. With theoretical limitations and the success of Neural Networks, we believe that the path of amalgamating deep learning and resource allocation is worth exploring further with more complex objective functions.

Chapter 6

Conclusion and Future Work

The distribution of goods and chores among multiple agents is a complicated issue that has become more challenging due to technological advancements and the emergence of autonomous agents. The primary objective of this thesis was to explore the difficulties involved in the fair and efficient allocation of goods and chores, particularly with indivisible items.

We began by examining the concept of fairness and its contextual nature. We explored various fairness criteria and studied their computational complexity. We observed that finding an exact fair allocation is NP-hard, but relaxing the fairness notion leads to the possibility of polynomial-time algorithms. We conducted a detailed survey of the current literature on fair division for goods or chores and the existence and computational complexity results. We summarized the essential results in Tables 2.1 and 2.2. These clearly indicate that there are still gaps that need to be addressed. Given the theoretical limitations, we presented a data-driven approach that utilizes neural networks to find an approximate fair and efficient allocation, which is an NP-hard problem. Though, in recent times, there have been few attempts to learn algorithms through neural networks, the problem we addressed is a mixed integer program, and very little work had done towards learning algorithms for the same through a neural network. We presented a neural network architecture and training ideas to determine EF1 and efficient allocation approximately. Our analysis showed the efficacy of the approach.

We also explored the problem of externalities, which occur when an agent's utility depends not only on the bundle they obtain but also on the bundles allocated to other agents. We highlighted the significance of externalities in allocating necessary commodities, particularly during the COVID-19 pandemic, when there was a strong demand for life-supporting resources. We noted that ensuring fairness in the presence of externalities is particularly challenging. We studied fair and efficient algorithms in the presence of externalities and provided solutions for the same.

Future Work There are many exciting directions that one could extend the thesis. We suggest exploring new fairness criteria and efficient algorithms suitable for different contexts and scenarios. Another direction is designing efficient algorithms for computing approximately fair allocations for a combination of indivisible goods and chores. Additionally, we suggest studying the fairness of allocation in multi-agent systems and the interaction between agents in decision-making. We propose investigating the impact of different types of externalities on allocation and developing mechanisms that take them into account. We also suggest exploring other machine learning techniques, such as reinforcement learning and deep learning, for resource allocation problems. Lastly, we propose studying the allocation of resources in multi-agent systems with heterogeneous preferences and goals.

Overall, the fair allocation of resources is a critical issue that affects many areas of our lives. Addressing this problem requires interdisciplinary efforts from computer science, economics, mathematics, and other related fields. The results of this thesis provide a stepping stone for future research to build upon and make progress toward achieving fair and efficient allocation in practice.

Bibliography

[1] Martin Aleksandrov and Toby Walsh. "Greedy algorithms for fair division of mixed manna". In: *arXiv preprint arXiv:1911.11005* (2019).

[2] Martin Aleksandrov and Toby Walsh. "Greedy algorithms for fair division of mixed manna". In: *arXiv preprint arXiv:1911.11005* (2019).

[3] Martin Aleksandrov and Toby Walsh. "Online fair division: A survey". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 09. 2020, pp. 13557–13562.

[4] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. "Truthful allocation mechanisms without payments: Characterization and implications on fairness". In: *Proceedings of the 2017 ACM Conference on Economics and Computation*. 2017, pp. 545–562.

[5] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A Voudouris. "Maximum Nash welfare and other stories about EFX". In: *Theoretical Computer Science* 863 (2021), pp. 69–85.

[6] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. "On truthful mechanisms for maximin share allocations". In: *arXiv preprint arXiv:1605.04026* (2016).

[7] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. "Approximation algorithms for computing maximin share allocations". In: *ACM Transactions on Algorithms (TALG)* 13.4 (2017), pp. 1–28.

[8] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. "Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination". In: *Theoretical Computer Science* 841 (2020), pp. 94–109.

[9] Kenneth I Appel and Wolfgang Haken. *Every planar map is four colorable*. Vol. 98. American Mathematical Soc., 1989.

[10] Haris Aziz. "Developments in multi-agent fair allocation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 09. 2020, pp. 13563–13568.

[11] Haris Aziz. "Simultaneously achieving ex-ante and ex-post fairness". In: *International Conference on Web and Internet Economics*. Springer. 2020, pp. 341–355.

[12] Haris Aziz, Péter Biró, Jérôme Lang, Julien Lesca, and Jérôme Monnot. "Optimal reallocation under additive and ordinal preferences". In: (2016).

[13] Haris Aziz, Ioannis Caragiannis, and Ayumi Igarashi. "Fair allocation of combinations of indivisible goods and chores". In: (2018). URL: http://arxiv.org/abs/1807.10684.

[14] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. "Fair allocation of indivisible goods and chores". In: *Autonomous Agents and Multi-Agent Systems* 36.1 (2022), pp. 1–21.

[15] Haris Aziz, Serge Gaspers, Simon Mackenzie, and Toby Walsh. "Fair assignment of indivisible objects under ordinal preferences". In: *Artificial Intelligence* 227 (2015), pp. 71–92.

[16] Haris Aziz, Xin Huang, Nicholas Mattei, and Erel Segal-Halevi. *Computing Fair Utilitarian Allocations of Indivisible Goods*. Dec. 2020.

[17] Haris Aziz, Xin Huang, Nicholas Mattei, and Erel Segal-Halevi. "The constrained round robin algorithm for fair and efficient allocation". In: *arXiv preprint arXiv:1908.00161* (2019).

[18] Haris Aziz, Bo Li, and Xiaowei Wu. "Strategyproof and approximately maxmin fair share allocation of chores". In: *arXiv preprint arXiv:1905.08925* (2019).

[19] Haris Aziz and Simon Mackenzie. "A discrete and bounded envy-free cake cutting protocol for any number of agents". In: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2016, pp. 416–427.

[20] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. "A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation". In: *Operations Research Letters* 48.5 (2020), pp. 573–578.

[21] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. "Algorithms for Max-Min Share Fair Allocation of Indivisible Chores". In: AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 335–341.

[22] Haris Aziz, Warut Suksompong, Zhaohong Sun, and Toby Walsh. *Fairness Concepts for Indivisible Items with Externalities*. 2021. arXiv: 2110.09066 [cs.GT].

[23] Moshe Babaioff, Tomer Ezra, and Uriel Feige. "Fair and truthful mechanisms for dichotomous valuations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 6. 2021, pp. 5119–5126.

[24] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. "PROPm Allocations of Indivisible Goods to Multiple Agents". In: *arXiv preprint arXiv:2105.11348* (2021).

[25] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. "PROPm Allocations of Indivisible Goods to Multiple Agents". In: *arXiv preprint arXiv:2105.11348* (2021).

[26] SIDDHARTHA BANERJEE, VASILIS GKATZELIS, SAFWAN HOSSAIN, BILLY JIN, EVI MICHA, and NISARG SHAH. "Proportionally Fair Online Allocation of Public Goods". In: ().

[27] Siddharth Barman, Umang Bhaskar, and Nisarg Shah. "Optimal bounds on the price of fairness for indivisible goods". In: *Web and Internet Economics: 16th International Conference, WINE 2020, Beijing, China, December 7–11, 2020, Proceedings.* Springer. 2020, pp. 356–369.

[28] Siddharth Barman, Arpita Biswas, Sanath Krishnamurthy, and Yadati Narahari. "Groupwise maximin fair allocation of indivisible goods". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 32. 1. 2018.

[29] Siddharth Barman, Ganesh Ghalme, Shweta Jain, Pooja Kulkarni, and Shivika Narang. "Fair division of indivisible goods among strategic agents". In: *arXiv preprint arXiv:1901.09427* (2019).

[30] Siddharth Barman and Sanath Kumar Krishnamurthy. "Approximation Algorithms for Maximin Fair Division". In: *ACM Trans. Econ. Comput.* 8.1 (Mar. 2020). ISSN: 2167-8375. DOI: 10.1145/3381525. URL: https://doi.org/10.1145/3381525.

[31] Siddharth Barman and Sanath Kumar Krishnamurthy. "On the proximity of markets with integral equilibria". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 33. 01. 2019, pp. 1748–1755.

[32] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. "Finding fair and efficient allocations". In: *Proceedings of the 2018 ACM Conference on Economics and Computation.* 2018, pp. 557–574.

[33] Siddharth Barman and Paritosh Verma. "Truthful and Fair Mechanisms for Matroid-Rank Valuations". In: *arXiv preprint arXiv:2109.05810* (2021).

[34] Xiaohui Bei, Guangda Huzhang, and Warut Suksompong. "Truthful fair division without free disposal". In: *Social Choice and Welfare* 55.3 (2020), pp. 523–545.

[35] Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. "Fair division of mixed divisible and indivisible goods". In: *Artificial Intelligence* 293 (2021), p. 103436.

[36] Xiaohui Bei, Xinhang Lu, Pasin Manurangsi, and Warut Suksompong. "The Price of Fairness for Indivisible Goods". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. IJCAI'19. Macao, China: AAAI Press, 2019, pp. 81–87. ISBN: 9780999241141.

[37] Xiaohui Bei, Xinhang Lu, Pasin Manurangsi, and Warut Suksompong. "The price of fairness for indivisible goods". In: *Theory of Computing Systems* 65 (2021), pp. 1069–1093.

[38] Kristóf Bérczi, Erika R Bérczi-Kovács, Endre Boros, Fekadu Tolessa Gedefa, Naoyuki Kamiyama, Telikepalli Kavitha, Yusuke Kobayashi, and Kazuhisa Makino. "Envy-free relaxations for goods, chores, and mixed items". In: *arXiv preprint arXiv:2006.04428* (2020).

[39] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. "(Almost Full) EFX Exists for Four Agents (and Beyond)". In: *arXiv preprint arXiv:2102.10654* (2021).

[40] Umang Bhaskar, AR Sricharan, and Rohit Vaish. "On approximate envy-freeness for indivisible chores and mixed resources". In: *arXiv preprint arXiv:2012.06788* (2020).

[41] Arpita Biswas and Siddharth Barman. "Fair Division Under Cardinality Constraints." In: *IJCAI*. 2018, pp. 91–97.

[42] Sylvain Bouveret. "Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity". In: *Journal of Artificial Intelligence Research* 32 (May 2008).

[43] Sylvain Bouveret, Yann Chevaleyre, Nicolas Maudet, and Hervé Moulin. "Fair Allocation of Indivisible Goods". In: *Handbook of Computational Social Choice*. Ed. by Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D.Editors Procaccia. Cambridge University Press, 2016, pp. 284–310.

[44] Sylvain Bouveret, Ulle Endriss, Jérôme Lang, et al. "Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods." In: *ECAI*. 2010, pp. 387–392.

[45] Sylvain Bouveret and Michel Lematre. "Characterizing conflicts in fair division of indivisible goods using a scale of criteria". In: *Autonomous Agents and Multi-Agent Systems* 30.2 (2016), pp. 259–290.

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

[47] Simina Brânzei, Ariel Procaccia, and Jie Zhang. "Externalities in cake cutting". In: *Twenty-Third International Joint Conference on Artificial Intelligence*. 2013.

[48] Simina Brânzei and Fedor Sandomirskiy. "Algorithms for competitive division of chores". In: *arXiv preprint arXiv:1907.01766* (2019).

[49] Robert Bredereck, Andrzej Kaczmarczyk, Dušan Knop, and Rolf Niedermeier. "High-Multiplicity Fair Allocation: Lenstra Empowered by N-Fold Integer Programming". In: *Proceedings of the 2019 ACM Conference on Economics and Computation*. EC '19. Phoenix, AZ, USA, pp. 505–523. ISBN: 9781450367929.

[50] Gianluca Brero, Benjamin Lubin, and Sven Seuken. *Machine Learning-powered Iterative Combinatorial Auctions*. 2020. arXiv: 1911.08042 [cs.GT].

[51] Eric Budish. "The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes". In: *Journal of Political Economy* 119.6 (2011), pp. 1061–1103.

[52] Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. "Reinforcement Mechanism Design for E-Commerce". In: *Proceedings of the 2018 World Wide Web Conference*. WWW '18. Lyon, France, 2018, pp. 1339–1348. ISBN: 9781450356398.

[53] Ioannis Caragiannis, Nick Gravin, and Xin Huang. "Envy-freeness up to any item with high Nash welfare: The virtue of donating items". In: *Proceedings of the 2019 ACM Conference on Economics and Computation*. 2019, pp. 527–545.

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

[55] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. "The unreasonable fairness of maximum Nash welfare". In: *ACM Transactions on Economics and Computation (TEAC)* 7.3 (2019), pp. 1–32.

[56] Mithun Chakraborty, Ayumi Igarashi, Warut Suksompong, and Yair Zick. "Weighted envy-freeness in indivisible item allocation". In: *ACM Transactions on Economics and Computation (TEAC)* 9.3 (2021), pp. 1–39.

[57] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. "EFX exists for three agents". In: *Proceedings of the 21st ACM Conference on Economics and Computation*. 2020, pp. 1–19.

[58] Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, Ruta Mehta, and Pranabendu Misra. "Improving EFX guarantees through rainbow cycle number". In: *Proceedings of the 22nd ACM Conference on Economics and Computation*. 2021, pp. 310–311.

[59] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. "A little charity guarantees almost envy-freeness". In: *SIAM Journal on Computing* 50.4 (2021), pp. 1336–1358.

[60] Xingyu Chen and Zijie Liu. "The fairness of leximin in allocation of indivisible chores". In: *arXiv preprint arXiv:2005.04864* (2020).

[61] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. "Fair public decision making". In: *Proceedings of the 2017 ACM Conference on Economics and Computation*. 2017, pp. 629–646.

[62] Vincent Conitzer, Rupert Freeman, Nisarg Shah, and Jennifer Wortman Vaughan. "Group fairness for the allocation of indivisible goods". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 1853–1860.

[63] Marco Dall'Aglio. "The Dubins–Spanier optimization problem in fair division theory". In: *Journal of Computational and Applied Mathematics* 130.1-2 (2001), pp. 17–40.

[64] Benoit Decerf, Francisco H.G. Ferreira, Daniel G. Mahler, and Olivier Sterck. "Lives and livelihoods: Estimates of the global mortality and poverty effects of the Covid-19 pandemic". In: *World Development* 146 (2021), p. 105561. ISSN: 0305-750X. DOI: https://doi.org/10.1016/j.worlddev.2021.105561. URL: https://www.sciencedirect.com/science/article/pii/S0305750X21001765.

[65] John Dickerson, Jonathan Goldman, Jeremy Karp, Ariel Procaccia, and Tuomas Sandholm. "The computational rise and fall of fairness". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28. 1. 2014.

[66] Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. "Optimal Auctions through Deep Learning". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 1706–1715.

[67] Edmund Eisenberg and David Gale. “Consensus of subjective probabilities: The pari-mutuel method”. In: *The Annals of Mathematical Statistics* 30.1 (1959), pp. 165–168.

[68] Brandon Fain, Kamesh Munagala, and Nisarg Shah. “Fair allocation of indivisible public goods”. In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. 2018, pp. 575–592.

[69] Alireza Farhadi, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Sebastien Lahaie, David Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. “Fair allocation of indivisible goods to asymmetric agents”. In: *Journal of Artificial Intelligence Research* 64 (2019), pp. 1–20.

[70] Uriel Feige, Ariel Sapir, and Laliv Tauber. “A tight negative example for MMS fair allocations”. In: *arXiv preprint arXiv:2104.04977* (2021).

[71] Uriel Feige, Ariel Sapir, and Laliv Tauber. “A tight negative example for MMS fair allocations”. In: *arXiv preprint arXiv:2104.04977* (2021).

[72] Duncan K. Foley. “Resource allocation and the public sector”. In: 1967.

[73] Rupert Freeman, Nisarg Shah, and Rohit Vaish. “Best of both worlds: Ex-ante and ex-post fairness in resource allocation”. In: *Proceedings of the 21st ACM Conference on Economics and Computation*. 2020, pp. 21–22.

[74] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. “Equitable Allocations of Indivisible Goods”. In: *IJCAI*. 2019.

[75] Dinesh Garg, Y Narahari, and Sujit Gujar. “Foundations of mechanism design: A tutorial Part 2-Advanced concepts and results”. In: *Sadhana* 33.2 (2008), pp. 131–174.

[76] Dinesh Garg, Yadati Narahari, and Sujit Gujar. “Foundations of mechanism design: A tutorial part 1-key concepts and classical results”. In: *Sadhana* 33.2 (2008), pp. 83–130.

[77] Jugal Garg, Peter McGlaughlin, and Setareh Taki. "Approximating maximin share allocations". English (US). In: *2nd Symposium on Simplicity in Algorithms, SOSA 2019 - Co-located with the 30th ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*. Ed. by Jeremy T. Fineman and Michael Mitzenmacher. OpenAccess Series in Informatics. Funding Information: Work on this paper partly supported by NSF CRII Award 1755619. Publisher Copyright: © Jugal Garg, Peter McGlaughlin, and Setareh Taki.; 2nd Symposium on Simplicity in Algorithms, SOSA 2019 ; Conference date: 08-01-2019 Through 09-01-2019. Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, Jan. 2019. DOI: 10.4230/OASIcs.SOSA.2019.20.

[78] Jugal Garg and Setareh Taki. "An improved approximation algorithm for maximin shares". In: *Artificial Intelligence* 300 (2021), p. 103547.

[79] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. "Fair allocation of indivisible goods: Improvements and generalizations". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. 2018, pp. 539–556.

[80] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: JMLR Workshop and Conference Proceedings, 13–15 May 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

[81] Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. "Deep Learning for Multi-Facility Location Mechanism Design". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 2018, pp. 261–267. ISBN: 9780999241127.

[82] Artur Gorokh, Siddhartha Banerjee, and Krishnamurthy Iyer. “The Remarkable Robustness of the Repeated Fisher Market”. In: *Available at SSRN 3411444* (2019).

[83] Laurent Gourvès and Jérôme Monnot. “On maximin share allocations in matroids”. In: *Theoretical Computer Science* 754 (2019), pp. 50–64.

[84] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. “Near Fairness in Matroids.” In: *ECAI*. 2014, pp. 393–398.

[85] Daniel Halpern and Nisarg Shah. “Fair and efficient resource allocation with partial information”. In: *arXiv preprint arXiv:2105.10064* (2021).

[86] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, Hejun Wang, and Lirong Xia. “Fair division through information withholding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 02. 2020, pp. 2014–2021.

[87] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. “Fairly Dividing Mixtures of Goods and Chores under Lexicographic Preferences”. In: *arXiv preprint arXiv:2203.07279* (2022).

[88] *https://en.dopl3r.com/memes/dank/how-my-brother-cut-the-pizza-how-it-looked-to-me/1483307*.

[89] *https://www.exportersindia.com/robotek-mobile/fexmon-immunity-boost-special-pizza-topping-5616952.htm*.

[90] Xin Huang and Pinyan Lu. “An algorithmic framework for approximating maximin share allocation of chores”. In: *Proceedings of the 22nd ACM Conference on Economics and Computation*. 2021, pp. 630–631.

[91] Halvard Hummel and Magnus Lie Hetland. “Fair Allocation of Conflicting Items”. In: *Autonomous Agents and Multi-Agent Systems* 36.1 (Apr. 2022). ISSN: 1387-2532. DOI: 10.1007/s10458-021-09537-3. URL: https://doi.org/10.1007/s10458-021-09537-3.

[92] Aanund Hylland and Richard Zeckhauser. "The efficient allocation of individuals to positions". In: *Journal of Political economy* 87.2 (1979), pp. 293–314.

[93] Bart de Keijzer, Sylvain Bouveret, Tomas Klos, and Yingqian Zhang. "On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences". In: *International Conference on Algorithmic Decision Theory*. Springer. 2009, pp. 98–110.

[94] Hyeji Kim, Yihan Jiang, Ranvir B. Rana, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. "Communication Algorithms via Deep Learning". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=ryazCMbR-.

[95] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014).

[96] Rucha Kulkarni, Ruta Mehta, and Setareh Taki. "Indivisible mixed manna: On the computability of MMS+ PO allocations". In: *Proceedings of the 22nd ACM Conference on Economics and Computation*. 2021, pp. 683–684.

[97] David Kurokawa, Ariel D Procaccia, and Junxing Wang. "When can the maximin share guarantee be guaranteed?" In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[98] Euiwoong Lee. "APX-hardness of maximizing Nash social welfare with indivisible items". In: *Information Processing Letters* 122 (2017), pp. 17–20.

[99] Bo Li, Yingkai Li, and Xiaowei Wu. "Almost (Weighted) Proportional Allocations for Indivisible Chores". In: *arXiv preprint arXiv:2103.11849* (2021).

[100] Zhentao Li and Adrian Vetta. "The fair division of hereditary set systems". In: *ACM Transactions on Economics and Computation (TEAC)* 9.2 (2021), pp. 1–19.

[101] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. "On approximately fair allocations of indivisible goods". In: *Proceedings of the 5th ACM Conference on Electronic Commerce*. 2004, pp. 125–131.

[102] Padala Manisha and Sujit Gujar. "Thompson Sampling Based Multi-Armed-Bandit Mechanism Using Neural Networks". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '19. Montreal QC, Canada, 2019, pp. 2111–2113. ISBN: 9781450363099.

[103] Padala Manisha, C. V. Jawahar, and Sujit Gujar. "Learning Optimal Redistribution Mechanisms Through Neural Networks". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. 2018, pp. 345–353.

[104] Pasin Manurangsi and Warut Suksompong. "Closing gaps in asymptotic fair division". In: *SIAM Journal on Discrete Mathematics* 35.2 (2021), pp. 668–706.

[105] Pasin Manurangsi and Warut Suksompong. "When Do Envy-Free Allocations Exist?" In: *SIAM Journal on Discrete Mathematics* 34 (Jan. 2020), pp. 1505–1521.

[106] Evangelos Markakis. "Approximation Algorithms and Hardness Results for Fair Division with Indivisible Goods". In: *Trends in Computational Social Choice*. Ed. by Ulle Endriss. AI Access, 2017. Chap. 12, pp. 231–247.

[107] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*. Vol. 1. Oxford university press New York, 1995.

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

[109] Hervé Moulin. "On strategy-proofness and single peakedness". In: *Public Choice* 35.4 (1980), pp. 437–455.

[110] Roger B Myerson. "Optimal auction design". In: *Mathematics of operations research* 6.1 (1981), pp. 58–73.

[111] Vinod Nair, Sergey Bartunov, Felix Gimeno, Ingrid von Glehn, Pawel Lichocki, Ivan Lobov, Brendan O'Donoghue, Nicolas Sonnerat, Christian Tjandraatmadja, Pengming Wang, Ravichandra Addanki, Tharindi Hapuarachchi, Thomas Keck, James Keeling, Pushmeet Kohli, Ira Ktena, Yujia Li, Oriol Vinyals, and Yori Zwols. "Solving Mixed Integer Programs Using Neural Networks". In: *ArXiv* abs/2012.13349 (2020).

[112] Trung Thanh Nguyen and Jörg Rothe. "Minimizing envy and maximizing average Nash social welfare in the allocation of indivisible goods". In: *Discrete Applied Mathematics* 179 (2014), pp. 54–68.

[113] James B Orlin. "Improved algorithms for computing Fisher's market clearing prices: Computing Fisher's market clearing prices". In: *Proceedings of the forty-second ACM symposium on Theory of computing*. 2010, pp. 291–300.

[114] Manisha Padala and Sujit Gujar. "Fnnc: Achieving fairness through neural networks". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI-20}, International Joint Conferences on Artificial Intelligence Organization*. 2020.

[115] Manisha Padala and Sujit Gujar. "Mechanism Design without Money for Fair Allocations". In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. 2021, pp. 382–389.

[116] Justin Payan and Yair Zick. "I Will Have Order! Optimizing Orders for Fair Reviewer Assignment". In: *arXiv preprint arXiv:2108.02126* (2021).

[117] Benjamin Plaut and Tim Roughgarden. "Almost envy-freeness with general valuations". In: *SIAM Journal on Discrete Mathematics* 34.2 (2020), pp. 1039–1068.

[118] Ariel D Procaccia and Junxing Wang. "Fair enough: Guaranteeing approximate maximin shares". In: *Proceedings of the fifteenth ACM conference on Economics and computation*. 2014, pp. 675–692.

[119] Ariel D. Procaccia and Hervé Moulin. "Cake Cutting Algorithms". In: *Handbook of Computational Social Choice*. Ed. by Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D.Editors Procaccia. Cambridge University Press, 2016, pp. 311–330.

[120] Jean-Charles Rochet. "A necessary and sufficient condition for rationalizability in a quasi-linear context". In: *Journal of mathematical Economics* 16.2 (1987), pp. 191–200.

[121] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: vol. 9351. Oct. 2015, pp. 234–241. ISBN: 978-3-319-24573-7.

[122] Tim Roughgarden. "Algorithmic game theory". In: *Communications of the ACM* 53.7 (2010), pp. 78–86.

[123] Tuomas Sandholm and Craig Boutilier. "Preference elicitation in combinatorial auctions". In: *Combinatorial auctions* 10 (2006).

[124] Masoud Seddighin, Hamed Saleh, and Mohammad Ghodsi. "Externalities and fairness". In: *The World Wide Web Conference*. 2019, pp. 538–548.

[125] Masoud Seddighin, Hamed Saleh, and Mohammad Ghodsi. "Externalities and fairness". In: *The World Wide Web Conference*. 2019, pp. 538–548.

[126] Erel Segal-Halevi. "Cake-cutting with different entitlements: How many cuts are needed?" In: *Journal of Mathematical Analysis and Applications* 480.1 (2019), p. 123382.

[127] Sujit Gujar Shaily Mishra Padala Manisha. "Fair Allocation Of Goods And Chores – Tutorial And Survey of Recent Results". In: *arXiv preprint arXiv:2307.10985* (2021).

[128] Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. "Reinforcement Mecha-

nism Design, with Applications to Dynamic Pricing in Sponsored Search Auctions". In: (2017). arXiv: 1711.10279. URL: http://arxiv.org/abs/1711.10279.

[129] Weiran Shen, Pingzhong Tang, and Song Zuo. "Automated Mechanism Design via Neural Networks". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '19. Montreal QC, Canada: International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 215–223. ISBN: 9781450363099.

[130] Weiran Shen, Pingzhong Tang, and Song Zuo. "Automated mechanism design via neural networks". In: *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*. 2019, pp. 215–223.

[131] Hugo Steihaus. "The problem of fair division". In: *Econometrica* 16 (1948), pp. 101–104.

[132] Francis Edward Su. "Cake-Cutting Algorithms: Be Fair If You Can. By Jack Robertson and William Webb". In: *The American Mathematical Monthly* 107.2 (2000), pp. 185–188.

[133] Tom Sühr, Asia J. Biega, Meike Zehlike, Krishna P. Gummadi, and Abhijnan Chakraborty. "Two-Sided Fairness for Repeated Matchings in Two-Sided Markets: A Case Study of a Ride-Hailing Platform". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 3082–3092. ISBN: 9781450362016. DOI: 10.1145/3292500.3330793. URL: https://doi.org/10.1145/3292500.3330793.

[134] Warut Suksompong. "Asymptotic existence of proportionally fair allocations". In: *Mathematical Social Sciences* 81 (2016), pp. 62–65.

[135] Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. "A Neural Architecture for Designing Truthful and Efficient Auctions". In: (2019). URL: http://arxiv.org/abs/1907.05181.

[136] László A Végh. "A strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives". In: *SIAM Journal on Computing* 45.5 (2016), pp. 1729–1761.

[137] Rodrigo A Velez. "Fairness and externalities". In: *Theoretical Economics* 11.1 (2016), pp. 381–410.

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

[139] Guanhua Wang, Runqi Guo, Yuko Sakurai, Muhammad Ali Babar, and Mingyu Guo. "Mechanism Design for Public Projects via Neural Networks". In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '21. Virtual Event, United Kingdom: International Foundation for Autonomous Agents and Multiagent Systems, 2021, pp. 1380–1388. ISBN: 9781450383073.

[140] Jakob Weissteiner and Sven Seuken. "Deep Learning—Powered Iterative Combinatorial Auctions". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.02 (Apr. 2020), pp. 2284–2293. ISSN: 2159-5399. DOI: 10.1609/aaai.v34i02.5606. URL: http://dx.doi.org/10.1609/aaai.v34i02.5606.

[141] Gerhard J Woeginger. "A polynomial-time approximation scheme for maximizing the minimum machine completion time". In: *Operations Research Letters* 20.4 (1997), pp. 149–154.

[142] Shengwei Zhou and Xiaowei Wu. "Approximately EFX allocations for indivisible chores". In: *arXiv preprint arXiv:2109.07313* (2021).