New Algorithms for the Fair and Efficient Allocation of Indivisible Chores*

Jugal Garg, **Aniket Murhekar**, **John Qin**
University of Illinois, Urbana-Champaign
{jugal, aniket2, johnqin2}@illinois.edu

Abstract

We study the problem of fairly and efficiently allocating indivisible chores among agents with additive disutility functions. We consider the widely-used envy-based fairness properties of EF1 and EFX, in conjunction with the efficiency property of fractional Pareto-optimality (fPO). Existence (and computation) of an allocation that is simultaneously EF1/EFX and fPO are challenging open problems, and we make progress on both of them. We show existence of an allocation that is

- EF1+fPO, when there are three agents,
- EF1+fPO, when there are at most two disutility functions,
- EFX+fPO, for three agents with bivalued disutility functions.

These results are constructive, based on strongly polynomial-time algorithms. We also investigate non-existence and show that an allocation that is EFX+fPO need not exist, even for two agents.

1 Introduction

Discrete fair division has recently received significant attention due to its applications in a wide variety of multi-agent settings; see recent surveys [Aziz, 2020; Walsh, 2020; Aziz *et al.*, 2022a]. Given a set of indivisible items and a set of n agents with diverse preferences, the goal is to find an allocation that is *fair* (i.e., acceptable by all agents) and *efficient* (i.e., non-wasteful). We assume that agents have additive valuations. The standard economic efficiency notions are Pareto-optimality (PO) and its strengthening fractional Pareto-optimality (fPO). Fairness notions based on *envy* [Foley, 1967] are most popular, where an allocation is said to be *envy-free* (EF) if every agent weakly prefers her bundle of items to any other agent's bundle of items. Since EF allocations need not exist, (e.g., dividing one item among two agents) its relaxations *envy-free up to any item* (EFX) [Caragiannis *et al.*, 2019] and *envy-free up to one item* (EF1) [Lipton *et al.*, 2004; Budish, 2011] are most widely used, where EF $\Rightarrow$ EFX $\Rightarrow$ EF1.

Achieving both fairness and efficiency is utmost desirable since just an efficient allocation can be highly unfair and similarly, just a fair allocation can be highly inefficient[1]. However, attaining both may not even be possible because these are often conflicting requirements, which put hard constraints on the set of all feasible allocations. Moreover, the landscape of known existence and tractability results varies depending on the nature of the items. The items to be divided can be either goods (which provide utility) or chores (which provide disutility).

For the case of goods, a series of works provided many remarkable results showing existence of EF1+(f)PO allocations. There are two broad approaches. The first uses the concept of *Nash welfare*, which is the geometric mean of agent utilities. [Caragiannis *et al.*, 2019] showed that the allocation with the maximum Nash welfare (MNW) is both EF1 and PO. However, computing an MNW allocation is APX-hard [Lee, 2015; Garg *et al.*, 2017], thus rendering this approach ineffective for the fast computation of an EF1+PO allocation in general. A notable special case is when agents have binary valuations; the MNW allocation can be computed in polynomial time for this case [Darmann and Schauer, 2014]. The efficiency notion of PO does not seem appropriate for fast computation because even checking if an allocation is PO is a coNP-hard problem [de Keijzer *et al.*, 2009]. Therefore, for fast computation, we need to work with the stronger notion of fPO, which admits polynomial time verification.

The second approach achieves fairness while maintaining efficiency through a *competitive equilibrium* [Barman *et al.*, 2018a; Barman and Krishnamurthy, 2019; Garg and Murhekar, 2021b; Garg and Murhekar, 2021a] of a Fisher market. In a Fisher market, agents are endowed with some monetary budget which they use to 'buy' goods to maximize their utility. A competitive equilibrium is an allocation along with prices for the goods in which each agent only owns goods that give them maximum 'bang-for-buck', i.e., goods with the highest utility to price ratio. This property ensures that the resulting allocation is fPO. The idea is then to endow agents with fictitious budgets and maintain an allocation that is the outcome of a Fisher market, while perform-

*Work on this paper is supported by NSF Grant CCF-1942321.

[1]Consider 2 agents and 2 items where agent 1 prefers item 1 to 2 while agent 2 prefers item 2 to 1. Giving all items to one agent is an efficient but unfair; while giving item 2 to agent 1 and item 1 to agent 2 is a fair (according to EFX/EF1) but inefficient.

Instance type	EF1+fPO		EFX+fPO	
	Goods	Chores	Goods	Chores
General additive	✓ BKV [2018a], GM [2021b]†	?	✗ GM [2021a]	✗ (Thm. 2)
$n = 3$ agents	✓GM [2021b]	✓(Thm. 1)	✗ (Thm. 2)	✗ (Thm. 2)
Two-type	✓(Rem. 1)	✓(Thm. 3)	✗ (Thm. 2)	✗ (Thm. 2)
Bivalued	✓GM [2021b; 2021a]	✓GM [2022a], EPS [2022]	✓GM [2021a]	✓$n = 3$ (Thm. 4)
2-ary	✓GM [2021b]	✓$k_i \geq m$ (Lem. 8)	✗ (Thm. 2)	✗ (Thm. 2)

Table 1: State-of-the-art for EF1/EFX+fPO allocation of indivisible items. ✓denotes existence/polynomial-time algorithm, ✗ denotes non-existence, ? denotes (non-)existence is unknown, † denotes no polynomial-time algorithm is known. Colored cells highlight our results.

ing changes to the allocation, prices, and budgets to achieve fairness. [Barman *et al.*, 2018a] used this approach to show that an EF1+fPO allocation exists, and that an EF1+PO allocation can be computed in pseudo-polynomial time. Later, [Garg and Murhekar, 2021b] showed that an EF1+fPO allocation can be computed in pseudo-polynomial time in general, and in polynomial time for constantly many agents or when agents have k-ary valuations with constant k. For bivalued instances of goods, an EFX+fPO allocation was shown to be polynomial time computable [Garg and Murhekar, 2021a]. Designing a polynomial-time algorithm for computing an EF1+fPO allocation of goods remains a challenging open problem.

In contrast, the case of chores turns out to be much more difficult to work with, resulting in relatively slow progress despite significant efforts by many researchers. Neither of the above mentioned approaches seem to be directly applicable in the setting of chores. Indeed, given the absence of a global welfare function like Nash welfare, even the existence of EF1+PO allocations for chores is open. Using competitive equilibria for chores remains a promising approach, which also guarantees fPO. In a Fisher market for chores, agents have a monetary expectation, i.e., a salary, which they aim to obtain by performing chores which have associated *payments* instead of prices. The algorithms for computing an EF1+(f)PO allocation of goods [Barman *et al.*, 2018a; Garg and Murhekar, 2021b] use the CE framework and show termination via some potential function. The main difficulty in translating algorithms for goods to the chores setting seems to be that the price-rises and item transfers only increase the potential in the goods case. For chores, however, price (payment) changes and transfers do not push these potential functions in one direction like they do for goods, making it difficult to show termination.

Consequently, the existence of an EF1+(f)PO allocation for chores remains open except for the cases of two agents [Aziz *et al.*, 2019], bivalued instances [Ebadian *et al.*, 2022; Garg *et al.*, 2022a], and two types of chores [Aziz *et al.*, 2022b]. The problem becomes significantly difficult when there are $n > 2$ agents. Table 1 provides a summary of existing results that are relevant to our work. In this paper, we focus on the chores setting and make progress on the above mentioned problems. Our first set of results show that an allocation that is

- EF1+fPO exists when there are three agents. Our algorithm uses the competitive equilibrium framework (CE) to maintain an fPO allocation, while using the payments associated with chores to guide their transfer to reduce the envy between the agents. Our novel approach starts with one agent having the highest envy, and then makes careful chore transfers unilaterally away from this agent while maintaining that the other two agents have bounded envy.
- EF1+fPO exists when there are two types of agents, where agents of the same type have the same preferences. Note that this strictly generalizes the well-studied setting of identical agents [Barman *et al.*, 2018b; Plaut and Roughgarden, 2020] and subsumes the result of [Aziz *et al.*, 2019] computing EF1+fPO for two agents. We develop a novel approach combining the CE allocation of an appropriately constructed market with a round-robin procedure. Combining CE-based frameworks with envy-resolving algorithms may be an important tool in settling the problem in its full generality. Our approach also gives a similar result for the case of goods. Recently [Mahara, 2021] studied the same class for goods, and showed through an involved case analysis that an EFX (without fPO) allocation exists.
- EFX+fPO exists when there are three agents with bivalued preferences, where each disutility value is one of two values. This improves the result of [Zhou and Wu, 2022] which shows that EFX exists in this case. Similar to [Zhou and Wu, 2022], our algorithm is quite involved and is based on a case-by-case analysis. We first derive a simple algorithm for computing an EF1+fPO for bivalued preferences, which was recently shown to exist [Garg *et al.*, 2022a; Ebadian *et al.*, 2022]. An interesting aspect of the simple algorithm is that it outputs an allocation that gives each agent a *balanced* (i.e. almost equal) number of chores. We start from such a balanced EF1+fPO outcome and improve the guarantee from EF1 to EFX while maintaining the fPO property. Additionally, we show that EF1+PO exists for a class of 2-ary preferences, where each disutility value of an agent is one of two values, but these two values can be different for different agents. This class is not subsumed by bivalued instances.

All our existence results are accompanied by polynomial-time algorithms. Next, we investigate the non-existence of fair and efficient allocations and show that

- EFX+fPO need not exist when there are two agents with 2-ary disutility functions. Naturally, this also implies that an EFX+fPO allocation need not exist for two-type instances.

1.1 Further Related Work

The problems of computing an EF1/EFX + PO/fPO allocation have remained challenging open questions in their full generality. Most progress has been made for special cases, providing a better picture of (non-)existence and (in)tractable classes of discrete fair division.

EF1. Barman, Krishnamurthy, and Vaish showed that an EF1+PO allocation of goods is computable in pseudo-polynomial time [Barman *et al.*, 2018a] and in polynomial time for binary instances [Barman *et al.*, 2018b]. [Garg and Murhekar, 2021b] showed algorithms for computing an EF1+fPO allocation in (i) pseudo-polynomial time for general additive goods, (ii) poly-time for k-ary instances with constant k, (iii) poly-time for constantly many agents. For chores, it is not known if EF1+PO allocations exist. Recently, existence and poly-time computation of an EF1+(f)PO allocation was shown for two agents by [Aziz *et al.*, 2019], for bivalued chores by [Garg *et al.*, 2022a] and [Ebadian *et al.*, 2022], and for two types of chores by [Aziz *et al.*, 2022b].

EFX. [Plaut and Roughgarden, 2020] showed that an EFX+PO allocation exists for the case of identical goods using the leximin mechanism. [Garg and Murhekar, 2021a] showed an EFX+fPO allocation can be computed in poly-time for bivalued goods, and non-existence for 3-valued instances which was also shown in [Freeman *et al.*, 2019]. For both goods and chores, the existence of EFX allocations is a challenging open problem. Existence is known in the goods case for three agents [Chaudhury *et al.*, 2020; Akrami *et al.*, 2022] and two types of agents [Mahara, 2021], and for two types of goods [Gorantla *et al.*, 2022]. In the chores case, EFX allocations are known to exist for three bivalued agents [Zhou and Wu, 2022] and when there are two types of chores [Aziz *et al.*, 2022b].

We refer the reader to excellent recent surveys [Aziz, 2020; Walsh, 2020; Aziz *et al.*, 2022a] for a discussion of results in the extensive fair division literature, including other fairness notions like Prop/MMS.

Organization of the paper. Section 2 introduces relevant notation and definitions. Sections 3 and 4 discuss our algorithms computing an EF1+fPO allocation for three agents and two-type instances respectively. Section 5 states our result proving the efficient computation of an EFX+fPO allocation for three bivalued agents. We defer missing proofs and illustrative examples to the full version of our paper [Garg *et al.*, 2022b].

2 Notation and Preliminaries

In a *chore allocation* instance (N, M, D), we are given a set $N = [n]$ of n agents, a set $M = [m]$ of m indivisible chores, and a set $D = \{d_i\}_{i\in[n]}$, where $d_i : 2^M \to \mathbb{R}_{\geq 0}$ is agent i's *disutility* or *cost* function over the chores. We assume that $d_i(\varnothing) = 0$. We let $d_i(j)$ denote the disutility agent i incurs from chore j and assume disutility functions are additive, so that for every $i \in N$ and $S \subseteq M$, $d_i(S) = \sum_{j\in S} d_i(j)$. In this work, we focus on the following classes of the chore allocation problem. We call a chore allocation instance a:

- *Three agent instance* if there are $n = 3$ agents.
- *Two-type instance* if there exist disutility functions d_1 and d_2 so that for all $i \in N$, $d_i \in \{d_1, d_2\}$. That is, every agent has one of two unique disutility functions.
- *Bivalued instance* if there exist $a, b \in \mathbb{R}_+$ so that for all $i \in N$ and $j \in M$ we have $d_i(j) \in \{a, b\}$. Here, rather than two disutility functions, we have two chore costs.
- *2-ary instance* if for each $i \in N$ there exist $a_i, b_i \in \mathbb{R}_+$ so that for all $j \in M$ we have $d_i(j) \in \{a_i, b_i\}$. Clearly, 2-ary instances strictly generalize bivalued instances. However, neither is comparable with two-type instances.

An *allocation* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ is an n-partition of the chores where agent i receives bundle $\mathbf{x}_i \subseteq M$ and incurs disutility $d_i(\mathbf{x}_i)$. A *fractional allocation* $\mathbf{x} \in [0,1]^{n\times m}$, where $\mathbf{x}_{ij} \in [0, 1]$ denotes the fraction of chore j given to agent i, allows for the chores to be divided. Here $d_i(\mathbf{x}_i) = \sum_{j\in M} d_i(j)\cdot x_{ij}$. We will assume that allocations are integral unless explicitly stated otherwise.

We define our fairness notions. An allocation $\mathbf{x}$ is said to be:

- *Envy-free* if for all $i, h \in N$, $d_i(\mathbf{x}_i) \leq d_i(\mathbf{x}_h)$, i.e., every agent weakly prefers her own bundle to others' bundles.
- *Envy-free up to any chore* (EFX) if for all $i, h \in N$, $d_i(\mathbf{x}_i \setminus j) \leq d_i(\mathbf{x}_h)$ for any $j \in \mathbf{x}_i$, i.e., every agent weakly prefers her own bundle to any other agent's bundle after removing her own easiest (least disutility) chore.
- *Envy-free up to one chore* (EF1) if for all $i, h \in N$, $d_i(\mathbf{x}_i \setminus j) \leq d_i(\mathbf{x}_h)$ for some $j \in \mathbf{x}_i$, i.e., every agent weakly prefers her own bundle to any other agent's bundle after removing her own hardest (highest disutility) chore. We use $d_{i_{-1}}(S)$ to denote $\min_{j\in S} d_i(S \setminus j)$ for $S \subseteq M$. Thus $\mathbf{x}$ is EF1 if $\forall i, h \in N$, $d_{i_{-1}}(\mathbf{x}_i) \leq d_i(\mathbf{x}_h)$.

We next define the efficiency notions of Pareto optimality (PO) and fractional Pareto optimality (fPO). An allocation $\mathbf{y}$ dominates an allocation $\mathbf{x}$ if for all $i \in N$, $d_i(\mathbf{y}_i) \leq d_i(\mathbf{x}_i)$, and there exists $h \in N$ such that $d_h(\mathbf{y}_h) < d_h(\mathbf{x}_h)$. An allocation is then PO if it is not dominated by any other allocation. Similarly, an allocation is fPO if it is not dominated by any fractional allocation. Note that an fPO allocation is necessarily PO, but not vice-versa.

Two important concepts we use in our algorithms are *competitive equilibrium* and *Fisher markets*. In the Fisher market setting we attach payments $\mathbf{p} = (p_1, \dots, p_m)$ to the chores. Agents perform chores in exchange for payment, with each agent i aiming to earn her *minimum payment* $e_i \geq 0$. Given a (fractional) allocation $\mathbf{x}$ and a set of payments $\mathbf{p}$, the *earning* of an agent i under $(\mathbf{x}, \mathbf{p})$ is given by $\mathbf{p}(\mathbf{x}_i) = \sum_{j\in M} p_j \cdot x_{ij}$. For each agent i, we define the *pain-per-buck* ratio α_{ij} of chore j as $\alpha_{ij} = d_i(j)/p_j$ and the *minimum-pain-per-buck* (MPB) ratio $\alpha_i = \min_{j\in M} \alpha_{ij}$. Further, we let $\mathsf{MPB}_i = \{j \in M \mid d_i(j)/p_j = \alpha_i\}$ denote the set of chores which are MPB for agent i for payments $\mathbf{p}$.

We say that $(\mathbf{x}, \mathbf{p})$ is a *competitive equilibrium* if (i) for all $j \in M$, $\sum_{i\in N} x_{ij} = 1$, i.e., all chores are completely allocated, (ii) for all $i \in N$, $\mathbf{p}(\mathbf{x}_i) = e_i$, i.e., each agent receives her minimum payment, and (iii) for all $i \in N$, $\mathbf{x}_i \subseteq \mathsf{MPB}_i$, i.e., agents receive only chores which are MPB for them. Competitive equilibria are known to guarantee economic efficiency via the First Welfare Theorem [Mas-Colell *et al.*, 1995], i.e., for a competitive equilibrium $(\mathbf{x}, \mathbf{p})$, the allocation $\mathbf{x}$ is fPO.

Given a competitive equilibrium $(\mathbf{x}, \mathbf{p})$ with integral allocation $\mathbf{x}$, we let $\mathbf{p}_{-1}(\mathbf{x}_i)$ denote the payment agent i receives from $\mathbf{x}_i$ excluding her highest paying chore. That is, $\mathbf{p}_{-1}(\mathbf{x}_i) = \min_{j \in \mathbf{x}_i} \mathbf{p}(\mathbf{x}_i \setminus j)$. We say that $(\mathbf{x}, \mathbf{p})$ is *payment envy-free up to one chore* (pEF1) if for all $i, h \in N$ we have $\mathbf{p}_{-1}(\mathbf{x}_i) \leq \mathbf{p}(\mathbf{x}_h)$. We say that agent i *pEF1-envies* h if $\mathbf{p}_{-1}(\mathbf{x}_i) > \mathbf{p}(\mathbf{x}_h)$. The following lemma shows that the pEF1 condition is in fact a strengthening of EF1.

Lemma 1. *Let $(\mathbf{x}, \mathbf{p})$ be an integral competitive equilibrium. If $(\mathbf{x}, \mathbf{p})$ is pEF1, then $\mathbf{x}$ is EF1+fPO.*

Proof. As $(\mathbf{x}, \mathbf{p})$ is pEF1, for all agents $i, h \in N$ we have $\mathbf{p}_{-1}(\mathbf{x}_i) \leq \mathbf{p}(\mathbf{x}_h)$. Since an agent i has only MPB chores, i.e., $\mathbf{x}_i \subseteq \mathsf{MPB}_i$, $d_{i_{-1}}(\mathbf{x}_i) = \alpha_i \mathbf{p}_{-1}(\mathbf{x}_i)$ and $d_i(\mathbf{x}_h) \geq \alpha_i \mathbf{p}(\mathbf{x}_h)$, where α_i is the MPB ratio of agent i. This gives us $d_{i_{-1}}(\mathbf{x}_i) = \alpha_i \mathbf{p}_{-1}(\mathbf{x}_i) \leq \alpha_i \mathbf{p}(\mathbf{x}_h) \leq d_i(\mathbf{x}_h)$, showing that $\mathbf{x}$ is EF1. Additionally, the First Welfare Theorem [Mas-Colell *et al.*, 1995] implies that the allocation $\mathbf{x}$ is fPO for a competitive equilibrium $(\mathbf{x}, \mathbf{p})$. □

Lemma 1 suggests that in order to compute an EF1+fPO allocation, one can search instead for a pEF1+fPO allocation, if possible. To do this, a natural approach is to start with an integral competitive equilibrium $(\mathbf{x}, \mathbf{p})$, and perform changes to both the allocation $\mathbf{x}$ and associated payments $\mathbf{p}$ until the allocation is EF1 (or pEF1), while maintaining that $(\mathbf{x}, \mathbf{p})$ remains an integral competitive equilibrium. The latter ensures that $\mathbf{x}$ is always fPO. We therefore perform a transfer of a chore $j \in \mathbf{x}_h$ from an agent h to an agent i only if $j \in \mathsf{MPB}_i$. If we wish to take a chore away from an agent h to reduce her envy and yet for all agents i we have $\mathbf{x}_h \cap \mathsf{MPB}_i = \emptyset$, then no agent i can receive a chore of h. In such a case we must adjust the payments to ensure some chore of h enters the MPB set of another agent i. This can be done in one of two ways. The first way is to *reduce the payments* of chores owned by a subset of agents other than h, done in a manner that does not violate the MPB condition of any agent, until the agents undergoing *payment drops* find the chores of h attractive enough to enter their MPB set. Alternatively, one could *raise the payments* of chores of h, along with possibly some other chores to ensure that the MPB condition of no agent is violated, until an agent who did not undergo a *payment rise* finds the chores of h attractive enough to enter their MPB set. Once there is an agent i such that there is a chore $j \in \mathbf{x}_h \cap \mathsf{MPB}_i$, we can transfer j from h to i. Our goal is to perform chore transfers facilitated via payment drops or raises to modify an initial allocation to one that is EF1 (or pEF1).

Two agents will be of particular interest to execute the above ideas. An agent ℓ is called a *least earner* (LE) among agent set A if $\ell \in \mathsf{argmin}_{i \in A} \mathbf{p}(\mathbf{x}_i)$. An agent b is called a *big earner* (BE) among agent set A if $b \in \mathsf{argmax}_{i \in A} \mathbf{p}_{-1}(\mathbf{x}_i)$, i.e., among agents in A, b earns the highest payment from her bundle of chores after every agent removes their highest paying chore. We call an agent the *global* least/big earner when $A = N$. We will assume that a mentioned least earner or big earner is global unless explicitly noted otherwise. The next lemma shows the importance of the BE and LE agents.

Lemma 2. *An integral competitive equilibrium $(\mathbf{x}, \mathbf{p})$ is pEF1 if and only if a big earner b does not pEF1-envy a least earner ℓ.*

Proof. It is clear that if $(\mathbf{x}, \mathbf{p})$ is pEF1 then b does not pEF1-envy ℓ. We need only show that if b does not pEF1-envy ℓ, then $(\mathbf{x}, \mathbf{p})$ is pEF1. For all $i, h \in N$, we have that $\mathbf{p}_{-1}(\mathbf{x}_i) \leq \mathbf{p}_{-1}(\mathbf{x}_b)$ by the definition of BE, and that $\mathbf{p}(\mathbf{x}_\ell) \leq \mathbf{p}(\mathbf{x}_h)$ by the definition of LE. Putting these together with $\mathbf{p}_{-1}(\mathbf{x}_b) \leq \mathbf{p}(\mathbf{x}_\ell)$ since b does not pEF1-envy ℓ, we get that $\mathbf{p}_{-1}(\mathbf{x}_i) \leq \mathbf{p}(\mathbf{x}_h)$. Thus i does not pEF1-envy h and $(\mathbf{x}, \mathbf{p})$ is pEF1. □

3 EF1 + fPO for Three Agents

In this section we prove the first main result of our paper.

Theorem 1. *Given a chore allocation instance (N, M, D) with three agents, an EF1 + fPO allocation exists. Furthermore, it can be computed in strongly polynomial-time.*

We prove Theorem 1 by showing that Algorithm 1 computes an EF1 + fPO allocation in polynomial time. Algorithm 1 begins by allocating the entire set of chores M to an arbitrarily chosen agent $i \in N$, with payments set so that $p_j = d_i(j)$, giving $\alpha_i = 1$.[2] This gives us an initial competitive equilibrium $(\mathbf{x}, \mathbf{p})$ where i is the big earner (BE). We define agent ℓ to be the least earner (LE) and agent h to be the *middle earner* (ME), i.e., the agent who is neither the big earner nor the least earner. In the initial allocation, ℓ and h are chosen arbitrarily after the BE i is chosen.

At a high-level, Algorithm 1 maintains a competitive equilibrium while transferring chores away from the initial BE i, while ensuring that the ME h and LE ℓ remain EF1 w.r.t. each other. Eventually, i must cease to be the BE. We show that when this happens we arrive at an EF1 allocation almost immediately.

First, we check if a chore transfer is possible from b to ℓ directly (Line 8), and if so, we make the transfer (Lines 9-10). If not, we check if there is a chore $j \in \mathbf{x}_h \cap \mathsf{MPB}_\ell$ that can be potentially transferred from h to ℓ (Line 11). If h pEF1-envies ℓ w.r.t. a chore $j \in \mathbf{x}_h \cap \mathsf{MPB}_\ell$, then j is transferred from h to ℓ. If h does not pEF1-envy ℓ, and if a chore j' can be transferred from b to h, then we perform the transfer (Lines 15-17). If there is no such chore j', then the payments of chores owned by both ℓ and h are dropped until a chore of b joins one of their MPB sets (Lines 19-21). Finally, if no chore of either h or b can be transferred to ℓ, we lower payments of chores of ℓ until a chore of h or b joins the MPB set of ℓ (Lines 23-25).

Thus, Algorithm 1 makes progress towards obtaining an EF1 allocation by reallocating chores while maintaining the following key properties.

(i) Agent i neither gains chores nor experiences payment drops.

[2] We can assume w.l.o.g. that $d_i(j) > 0$ for all i, j. Otherwise if $d_i(j) = 0$ for some i, j, we can simply allocate j to i and remove j from further consideration. It is easy to check that doing so does not affect EF1 or fPO properties.

Algorithm 1 Computing an EF1+fPO allocation for 3 agents

Input: Fair division instance (N, M, D) with $|N| = 3$
Output: An integral allocation $\mathbf{x}$

$\mathbf{x} \leftarrow$ give M to some agent $i \in \{1, 2, 3\}$
for $j \in M$ **do**
  $p_j \leftarrow d_i(j)$
while $\mathbf{x}$ is not EF1 **do**
  $b \leftarrow \mathsf{argmax}_{k \in N} \mathbf{p}_{-1}(\mathbf{x}_k)$ ▷ Big earner
  $\ell \leftarrow \mathsf{argmin}_{k \in N} \mathbf{p}(\mathbf{x}_k)$ ▷ Least earner
  $h \leftarrow k \in N \setminus \{b, \ell\}$ ▷ Middle earner
  if $\exists j \in \mathbf{x}_b \cap \mathsf{MPB}_\ell$ **then** ▷ A chore can be potentially transferred from b to ℓ
    $\mathbf{x}_b \leftarrow \mathbf{x}_b \setminus j$
    $\mathbf{x}_\ell \leftarrow \mathbf{x}_\ell \cup j$
  else if $\exists j \in \mathbf{x}_h \cap \mathsf{MPB}_\ell$ **then** ▷ A chore can be potentially transferred from h to ℓ
    if $\mathbf{p}(\mathbf{x}_h \setminus j) > \mathbf{p}(\mathbf{x}_\ell)$ **then** ▷ h pEF1-envies ℓ w.r.t. any chore $j \in \mathbf{x}_h \cap \mathsf{MPB}_\ell$
      $\mathbf{x}_h \leftarrow \mathbf{x}_h \setminus j$
      $\mathbf{x}_\ell \leftarrow \mathbf{x}_\ell \cup j$
    else if $\exists j' \in \mathbf{x}_b \cap \mathsf{MPB}_h$ **then** ▷ A chore can be potentially transferred from b to h
      $\mathbf{x}_b \leftarrow \mathbf{x}_b \setminus j'$
      $\mathbf{x}_h \leftarrow \mathbf{x}_h \cup j'$
    else▷ No chore can be transferred from b to h or ℓ
      $\beta \leftarrow \max_{i \in \{\ell, h\}, j \in \mathbf{x}_b} \frac{\alpha_i}{d_i(j)/p_j}$
      for $j \in \mathbf{x}_\ell \cup \mathbf{x}_h$ **do**
        $p_j \leftarrow p_j \cdot \beta$ ▷ Lower payments until a chore of b is MPB for ℓ or h
  else ▷ No chore can be transferred from b or h to ℓ
    $\beta \leftarrow \max_{j \in \mathbf{x}_b \cup \mathbf{x}_h} \frac{\alpha_\ell}{d_\ell(j)/p_j}$
    for $j \in \mathbf{x}_\ell$ **do**
      $p_j \leftarrow p_j \cdot \beta$ ▷ Lower payments until a chore of b or h is MPB for ℓ
return $\mathbf{x}$

(ii) If agent i ceases to be the big earner, an EF1 allocation is found in at most one chore transfer.

(iii) The allocation is always fPO. This is done by performing payment drops as necessary to maintain a competitive equilibrium, and ensuring chore transfers do not violate the competitive equilibrium.

We use Property (i) to bound the number of steps for i to cease being the big earner, after which Property (ii) gives us an EF1 allocation. Property (iii) then guarantees that our EF1 allocation is additionally fPO. We start with a preliminary lemma.

Lemma 3. *Given an integral competitive equilibrium $(\mathbf{x}, \mathbf{p})$ with big earner b and least earner ℓ, if after a single chore transfer b becomes a least earner or ℓ becomes a big earner, then the resulting allocation must be pEF1.*

Proof. Let $(\mathbf{x}', \mathbf{p})$ be the CE after the transfer and let b' and ℓ' denote the big earner and least earner respectively in $(\mathbf{x}', \mathbf{p})$.

We first show that if $b = \ell'$ then $(\mathbf{x}', \mathbf{p})$ is pEF1. Since a chore transfer was performed, $(\mathbf{x}, \mathbf{p})$ was not pEF1 and thus $\mathbf{p}_{-1}(\mathbf{x}_b) > \mathbf{p}(\mathbf{x}_\ell)$. Since b becomes a least earner in $(\mathbf{x}', \mathbf{p})$, it must be that a chore was transferred to ℓ, as we would otherwise have $\mathbf{p}(\mathbf{x}'_b) \geq \mathbf{p}_{-1}(\mathbf{x}_b) > \mathbf{p}(\mathbf{x}_\ell) = \mathbf{p}(\mathbf{x}'_\ell)$ and b could not be a least earner in $(\mathbf{x}', \mathbf{p})$. The earning up to one chore of the big earner cannot increase in $(\mathbf{x}', \mathbf{p})$ as compared to $(\mathbf{x}, \mathbf{p})$, as ℓ is the only agent who gets an extra chore, and we have $\mathbf{p}_{-1}(\mathbf{x}'_\ell) \leq \mathbf{p}(\mathbf{x}_\ell) < \mathbf{p}_{-1}(\mathbf{x}_b)$. Thus, $\mathbf{p}_{-1}(\mathbf{x}'_{b'}) \leq \mathbf{p}_{-1}(\mathbf{x}_b)$. It follows then that $\mathbf{p}_{-1}(\mathbf{x}'_{b'}) \leq \mathbf{p}_{-1}(\mathbf{x}_b) \leq \mathbf{p}(\mathbf{x}'_b) = \mathbf{p}(\mathbf{x}'_{\ell'})$, since $b = \ell'$. Hence by Lemma 1 $(\mathbf{x}', \mathbf{p})$ is pEF1.

We now show that if $\ell = b'$ then $(\mathbf{x}', \mathbf{p})$ is pEF1. As before, since a chore transfer was performed, $(\mathbf{x}, \mathbf{p})$ was not pEF1 and so $\mathbf{p}_{-1}(\mathbf{x}_b) > \mathbf{p}(\mathbf{x}_\ell)$. It must then be that a chore was taken from b, as we would otherwise have $\mathbf{p}_{-1}(\mathbf{x}'_b) \geq \mathbf{p}_{-1}(\mathbf{x}_b) > \mathbf{p}(\mathbf{x}_\ell) \geq \mathbf{p}_{-1}(\mathbf{x}'_\ell)$ and ℓ could not be a big earner in $(\mathbf{x}'\mathbf{p})$. The earning of the least earner cannot decrease in $(\mathbf{x}', \mathbf{p})$ as compared to $(\mathbf{x}, \mathbf{p})$, as b is the only agent who loses a chore, and we have $\mathbf{p}(\mathbf{x}'_b) \geq \mathbf{p}_{-1}(\mathbf{x}_b) > \mathbf{p}(\mathbf{x}_\ell)$. Thus, $\mathbf{p}(\mathbf{x}'_{\ell'}) \geq \mathbf{p}(\mathbf{x}_\ell)$. It follows that $\mathbf{p}(\mathbf{x}'_{\ell'}) \geq \mathbf{p}(\mathbf{x}_\ell) = \mathbf{p}(\mathbf{x}_{b'}) \geq \mathbf{p}_{-1}(\mathbf{x}'_{b'})$ since $\ell = b'$. Hence by Lemma 1 $(\mathbf{x}', \mathbf{p})$ is pEF1. □

We now record an important property of the algorithm when the identity of the least earner changes.

Lemma 4. *During the run of Algorithm 1, if an agent ℓ stops being the least earner, then either the pEF1 condition is satisfied or ℓ becomes the middle earner and does not pEF1-envy the new least earner.*

Proof. Clearly it cannot happen that agent ℓ stops being the least earner due to a payment drop. Thus ℓ stops being a least earner due to a chore transfer. Let $\mathbf{x}$ be the allocation immediately before the transfer and let $\mathbf{x}'$ be the allocation immediately afterwards, and let $\mathbf{p}$ be the payment vector. Additionally let b be the big earner and h the middle earner in $(\mathbf{x}, \mathbf{p})$. By Lemma 3, if b becomes the least earner or ℓ becomes the big earner in $(\mathbf{x}', \mathbf{p})$, then $(\mathbf{x}', \mathbf{p})$ is pEF1. Thus, if $(\mathbf{x}', \mathbf{p})$ is not, it must be that in $(\mathbf{x}', \mathbf{p})$, agent ℓ is the new middle earner and h is the new least earner. We now show that the new middle earner ℓ does not pEF1-envy the new least earner h by considering the possible combinations of agents involved in the transfer:

- Suppose a chore was transferred from b to h. In this case, $\mathbf{p}(\mathbf{x}'_\ell) = \mathbf{p}(\mathbf{x}_\ell) < \mathbf{p}_{-1}(\mathbf{x}_b) \leq \mathbf{p}(\mathbf{x}'_b)$, so ℓ would remain the least earner in $(\mathbf{x}', \mathbf{p})$, leading to a contradiction.
- Suppose a chore was transferred from b to ℓ. In this case, $\mathbf{p}_{-1}(\mathbf{x}'_\ell) \leq \mathbf{p}(\mathbf{x}_\ell) \leq \mathbf{p}(\mathbf{x}_h) = \mathbf{p}(\mathbf{x}'_h)$, thus showing that ℓ does not pEF1-envy h.
- Suppose a chore was transferred from h to ℓ. In this case, $\mathbf{p}(\mathbf{x}_h \setminus j) > \mathbf{p}(\mathbf{x}_\ell)$ for $j \in \mathbf{x}_h$. Then, $\mathbf{p}_{-1}(\mathbf{x}'_\ell) \leq \mathbf{p}(\mathbf{x}_\ell) < \mathbf{p}(\mathbf{x}_h \setminus j) = \mathbf{p}(\mathbf{x}'_h)$. This again shows that the ℓ does not pEF1-envy h.

In conclusion, either the allocation after the transfer is pEF1, or the new middle earner does not pEF1-envy the new least earner. □

Initially agent i is allocated all the chores, and is thus the big earner. We show that if agent i stops being the big earner then an EF1+fPO allocation is found almost immediately.

Lemma 5. *If agent i ceases to be the big earner then Algorithm 1 finds an EF1 allocation in at most one chore transfer.*

We defer the proof of Lemma 5 to the full version [Garg *et al.*, 2022b]. We now complete the proof of Theorem 1 with the following lemma.

Lemma 6. *During the run of Algorithm 1, the initial BE i ceases to be the BE in* $\mathsf{poly}(m)$ *steps.*

Proof. Throughout the run of the algorithm, i undergoes no payment drops and may only lose chores. Then, she must cease to be the big earner before losing all of her chores, which are at most m in number.

While i is a big earner, note that there can be at most two payment drops before a chore transfer must occur: one drop which makes a chore in $\mathbf{x}_h$ join the MPB set of ℓ, and one drop which makes a chore in $\mathbf{x}_i$ join the MPB set for either h or ℓ. We now bound the number of chore transfers between middle earner h and least earner ℓ before a chore must be taken from i. Since transfers between h and ℓ are always from h to ℓ, it must be that h becomes the least earner by the time she transfers all of her at most m chores to ℓ. Suppose h becomes the new least earner after transferring chore j to ℓ. Let $\mathbf{x}$ be the allocation before transferring j and let $\mathbf{x}'$ be the allocation after transferring j. By Lemma 3, ℓ must be the middle earner in $(\mathbf{x}', \mathbf{p})$ or we already have the pEF1 condition. Since j was transferred from h to ℓ we have $\mathbf{p}(\mathbf{x}_h \setminus j) > \mathbf{p}(\mathbf{x}_\ell)$ and it follows that $\mathbf{p}_{-1}(\mathbf{x}'_\ell) \le \mathbf{p}(\mathbf{x}_\ell) < \mathbf{p}(\mathbf{x}_h \setminus j) = \mathbf{p}(\mathbf{x}'_h)$. Notably, it remains that j is MPB for the least earner h in $(\mathbf{x}', \mathbf{p})$ and the next chore transfer is guaranteed to not be between h and ℓ. Subsequently, payments of chores in $\mathbf{x}_\ell \cup \mathbf{x}_h$ will be decreased until some chore j' in $\mathbf{x}_i$ joins the MPB set for either ℓ or h, if one is not already. Then j' will be transferred from i. Thus, in at most $\mathsf{poly}(m)$ chore transfers and payment drops, a chore is taken from i.

Since i can lose only m chores, in $\mathsf{poly}(m)$ steps i ceases to be the big earner. □

Having shown that EF1+fPO allocation is efficiently computable for three agents, a natural follow-up question is to investigate the existence and computation of EFX+fPO allocations for the same class. The following result shows that EFX+fPO allocations need not exist even for two agents.

Theorem 2. *There exists a two-type, 2-ary chore allocation instance with two agents which does not admit an EFX+fPO allocation.*

Despite this, we show in Theorem 4 that for *bivalued* instances with three agents an EFX+fPO allocation exists and can be computed in strongly polynomial-time.

4 EF1 + fPO for Two Types of Agents

In this section, we present Algorithm 3 which computes an EF1+fPO allocation for two-type instances in strongly polynomial-time. Due to Lemma 1, we seek a pEF1 integral competitive equilibrium $(\mathbf{x}, \mathbf{p})$. Let N_1 (resp. N_2) be an ordered list of agents with disutility function d_1, called Type-1 agents (resp. d_2, called Type-2 agents). Algorithm 3 maintains a partition of the chores M into sets M_1 and M_2, where M_i is allocated to N_i, for $i \in \{1, 2\}$. Initially, $M_1 = M$ and

Algorithm 2 RoundRobin (RR) procedure

Input: Ordered list of identical v agents N', set of chores M'
Output: An allocation $\mathbf{x}$

$\mathbf{x}_i \leftarrow \emptyset$ for each $i \in N'$, $i \leftarrow 1$, $P \leftarrow M'$
while $P \neq \emptyset$ **do**
  $j \leftarrow \mathsf{argmin}_{j' \in P}\, d_i(j')$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i \cup \{j\}$, $P \leftarrow P \setminus \{j\}$, $i \leftarrow i \mod |N'| + 1$
return $\mathbf{x}$

$M_2 = \emptyset$, and $p_j = d_1(j)$ for each $j \in M$. The chores in M_1 are allocated to N_1 using the RoundRobin procedure (Algorithm 2). Given an ordered list of agents N' and chores M', RoundRobin allocates chores as follows. Agents take turns picking chores according to the order specified by N' and each agent picks the least cost chore among the pool of remaining chores during their turn. It is a well-known folklore result that RoundRobin returns an EF1 allocation.

Initially, agents in N_1 have chores while N_2 have none, causing agents in N_1 to potentially pEF1-envy agents in N_2. Thus we must transfer chores from M_1 to M_2 to reduce this pEF1-envy. When necessary, payments of chores in M_1 are raised appropriately before such a transfer to maintain that chores in M_i are MPB for agents in N_i, for each $i \in \{1, 2\}$[3]. After each transfer, the chores in the (updated) sets M_i are re-allocated to N_i using the RoundRobin procedure, always using the *same ordering* of agents. Since agents in N_i have the same disutility function and chores in M_i are MPB for N_i, we have the following feature.

Invariant 1. *Throughout the run of Algorithm 3, agents in N_i do not pEF1-envy each other, for each $i \in \{1, 2\}$.*

By Invariant 1, no agent can pEF1-envy another agent of the same type. Thus, if we do not have pEF1, it must be that the global BE pEF1-envies the global LE, with the BE and LE being in different groups. Initially, the BE is in N_1 and the LE is in N_2. Our goal is now to eliminate the pEF1-envy between the BE and LE, and to do this we transfer chores from M_1 to M_2 with necessary payment raises and RoundRobin re-allocations. We then reconsider the pEF1-envy between the new BE and the new LE. The algorithm terminates when the BE no longer pEF1-envies the LE. We argue that this must happen.

While the BE remains in N_1, chores are transferred from M_1 to M_2. Clearly there can be at most m such transfers, since always $|M_1| \le m$. If in some iteration both the BE and LE belong to the same group N_i, then we must be done due to Invariant 1. The only remaining case is if the BE becomes an agent in N_2 and the LE becomes an agent in N_1. We address this case in the following lemma.

Lemma 7. *If the BE is in N_2 and the LE is in N_1, the allocation must already be pEF1.*

Proof. We first note that payment-raises do not change the identity of the LE and BE. Therefore, suppose that there is a

[3]Note that in this algorithm we perform payment raises instead of payment drops as in Algorithm 1. This is purely for ease of presentation, as we could equivalently perform payment drops on chores in M_2 instead of performing payment raises on chores in M_1

Algorithm 3 EF1+fPO for two types of agents

Input: Two-type instance (N, M, D) with $d_i \in \{d_1, d_2\}$
Output: An allocation $\mathbf{x}$

```
1: N_1 ← {i ∈ N | d_i = d_1}, N_2 ← {i ∈ N | d_i = d_2}
2: For each j ∈ M, set p_j ← d_1(j)
3: M_1 ← M, M_2 ← ∅
4: x ← RR(N_1, M_1) ∪ RR(N_2, M_2)    ▷ RR = RoundRobin
5: while (x, p) is not pEF1 do
6:     MPB_2 ← {j ∈ M : j is MPB for agents in N_2}
7:     if ∃j ∈ M_1 ∩ MPB_2 then
8:         M_1 ← M_1 \ {j}, M_2 ← M_2 ∪ {j}
9:         x ← RR(N_1, M_1) ∪ RR(N_2, M_2)
10:    else
11:        Raise payments of M_1 until |M_1 ∩ MPB_2| > 0
12: return x
```

transfer prior to which the global BE is in N_1 and global LE in N_2, but after which the global BE is in N_2 and global LE in N_1. Let $\mathbf{x}$ (resp. $\mathbf{x}'$) denote the allocation immediately before (resp. after) the transfer. For $i \in \{1, 2\}$, let b_i and ℓ_i denote the BE and LE among agent set N_i before the transfer, and let b'_i and ℓ'_i denote the BE and LE among agent sets N_i after the transfer. Let $\mathbf{p}$ be the payments vector accompanying $\mathbf{x}$ and $\mathbf{x}'$. We use the following:

Observation 1. In a RoundRobin allocation of a set of chores M' to a list of agents $N' = \{1, \dots, n'\}$ with identical valuations, the big earner (assuming payment vector is proportional to disutility vector) is the agent i who picks the last chore while the least earner is the agent $h = (i \mod n') + 1$ who would pick immediately after i.

This is because agents pick chores according to increasing disutility since they all have the same cost function. Thus in $\mathbf{x}$, the BE b_i picked the last chore when M_i was RoundRobin -allocated to N_i, for $i \in \{1, 2\}$. We now examine how the identity and earning of the BE and LE of each agent set N_i change after a chore j is transferred from M_1 to M_2.

When $M_1 \setminus \{j\}$ is RoundRobin -allocated to N_1, the agent who picks immediately before b_1 now picks last, and is the new BE. Thus Obs. 1 implies that new LE in N_1 is in fact b_1, i.e., $\ell'_1 = b_1$. Additionally, b_1's new total earning is at least as much as her previous earning without her highest paying chore, since in each round up to the last round she must now pick a weakly higher disutility (and thus higher paying) chore than before, but she no longer picks a chore in the last round. Thus:

$$\mathbf{p}(\mathbf{x}'_{\ell'_1}) = \mathbf{p}(\mathbf{x}'_{b_1}) \geq \mathbf{p}_{-1}(\mathbf{x}_{b_1}). \quad (1)$$

Conversely, when $M_2 \cup \{j\}$ is RoundRobin -allocated to N_2, the previous LE ℓ_2 now picks the last chore. Thus by Obs.1, ℓ_2 is the new BE, i.e., $b'_2 = \ell_2$. In each round up to the last ℓ_2 now picks a weakly lower disutility (and thus lower paying) chore than before, but ℓ_2 now receives a new worst chore in the last round. Thus:

$$\mathbf{p}_{-1}(\mathbf{x}'_{b'_2}) = \mathbf{p}_{-1}(\mathbf{x}'_{\ell_2}) \leq \mathbf{p}(\mathbf{x}_{\ell_2}). \quad (2)$$

Let us now examine the pEF1-envy before and after the chore transfer. Prior to the transfer, i.e., in $(\mathbf{x}, \mathbf{p})$, the global BE and LE are b_1 and ℓ_2 respectively. Since $(\mathbf{x}, \mathbf{p})$ is not pEF1, $\mathbf{p}_{-1}(\mathbf{x}_{b_1}) > \mathbf{p}(\mathbf{x}_{\ell_2})$. Using (1) and (2), we get:

$$\mathbf{p}(\mathbf{x}'_{\ell'_1}) \geq \mathbf{p}_{-1}(\mathbf{x}_{b_1}) > \mathbf{p}(\mathbf{x}_{\ell_2}) \geq \mathbf{p}_{-1}(\mathbf{x}'_{b'_2}),$$

implying that after the transfer, i.e., in $(\mathbf{x}', \mathbf{p})$, the global BE b'_2 does not pEF1-envy the global LE ℓ'_1. Thus $(\mathbf{x}', \mathbf{p})$ must be pEF1 by Lemma 2. $\square$

In conclusion, chores are transferred unilaterally from M_1 to M_2 with necessary payment-raises and RoundRobin reallocations among agents of the same type until the allocation is pEF1+fPO. Clearly, Algorithm 3 runs in poly(n, m) time. We conclude:

Theorem 3. *Given a two-type chore allocation instance (N, M, D), an EF1 + fPO allocation exists and can be computed in strongly polynomial-time.*

In contrast, Theorem 2 shows that EFX+fPO allocations need not exist for two-type instances. We conclude this section by noting that the same techniques apply to goods.

Remark 1. *An EF1+fPO allocation is strongly polynomial-time computable for a two-type goods allocation instance.*

5 EFX for Three Bivalued Agents

Recall from Theorem 2 that an EFX+fPO allocation is not guaranteed to exist, even for 2-ary instances. We therefore study the computation of EFX+fPO allocations for bivalued instances. Our third result is:

Theorem 4. *Given a bivalued chore allocation instance (N, M, D) with three agents, an EFX+fPO allocation exists and can be computed in strongly polynomial-time.*

The proof of Theorem 4 is quite involved, and we defer the discussion of involved algorithms and their analyses to the full version of the paper due to space constraints [Garg *et al.*, 2022b]. Along the way, we also obtain the following result:

Lemma 8. *For 2-ary instances where $\forall i \in N, k_i \geq m$, an EF1+PO allocation can be found in polynomial time.*

6 Conclusion

In this work, we described new algorithms computing fair and efficient allocations of chores under the fairness notions of EF1/EFX, and the efficiency notion of fPO. Our algorithms for the three agents and two-types setting are among the few positive non-trivial results known for the EF1+fPO problem. Combining CE-based frameworks with envy-resolving algorithms like RoundRobin may be an important tool in settling the problem in its full generality. We also described an algorithm for computing an EFX+fPO allocation for three bivalued agents. Extending and generalizing our approach to higher numbers of agents is a natural next step.

References

[Akrami *et al.*, 2022] Hannaneh Akrami, Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, and Ruta Mehta. EFX allocations: Simplifications and improvements. *CoRR*, abs/2205.07638, 2022.

[Aziz *et al.*, 2019] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, page 53–59, 2019.

[Aziz *et al.*, 2022a] Haris Aziz, Bo Li, Herve Moulin, and Xiaowei Wu. Algorithmic fair allocation of indivisible items: A survey and new questions, 2022.

[Aziz *et al.*, 2022b] Haris Aziz, Jeremy Lindsay, Angus Ritossa, and Mashbat Suzuki. Fair allocation of two types of chores, 2022.

[Aziz, 2020] Haris Aziz. Developments in multi-agent fair allocation. *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 13563–13568, 2020.

[Barman and Krishnamurthy, 2019] Siddharth Barman and Sanath Krishnamurthy. On the proximity of markets with integral equilibria. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1748–1755, 2019.

[Barman *et al.*, 2018a] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 557–574, 2018.

[Barman *et al.*, 2018b] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy algorithms for maximizing Nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, page 7–13, 2018.

[Budish, 2011] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061 – 1103, 2011.

[Caragiannis *et al.*, 2019] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Trans. Econ. Comput.*, 7(3), sep 2019.

[Chaudhury *et al.*, 2020] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. EFX exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, page 1–19, 2020.

[Darmann and Schauer, 2014] Andreas Darmann and Joachim Schauer. Maximizing Nash product social welfare in allocating indivisible goods. *SSRN Electronic Journal*, 247, 01 2014.

[de Keijzer *et al.*, 2009] 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 Francesca Rossi and Alexis Tsoukias, editors, *Algorithmic Decision Theory*, pages 98–110, 2009.

[Ebadian *et al.*, 2022] Soroush Ebadian, Dominik Peters, and Nisarg Shah. How to fairly allocate easy and difficult chores. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2022.

[Foley, 1967] D.K. Foley. Resource allocation and the public sector. *Yale Economic Essays*, 7(1):45–98, 1967.

[Freeman *et al.*, 2019] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable allocations of indivisible goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 280–286, 2019.

[Garg and Murhekar, 2021a] Jugal Garg and Aniket Murhekar. Computing fair and efficient allocations with few utility values. *In Proc. 14th Symp. Algorithmic Game Theory (SAGT)*, 2021.

[Garg and Murhekar, 2021b] Jugal Garg and Aniket Murhekar. On fair and efficient allocations of indivisible goods. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[Garg *et al.*, 2017] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. Satiation in Fisher markets and approximation of Nash social welfare. *CoRR*, abs/1707.04428, 2017.

[Garg *et al.*, 2022a] Jugal Garg, Aniket Murhekar, and John Qin. Fair and efficient allocations of chores under bivalued preferences. *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5043–5050, 2022.

[Garg *et al.*, 2022b] Jugal Garg, Aniket Murhekar, and John Qin. Improving fairness and efficiency guarantees for allocating indivisible chores. *arXiv preprint arXiv:2212.02440*, 2022.

[Gorantla *et al.*, 2022] Pranay Gorantla, Kunal Marwaha, and Santhoshini Velusamy. Fair allocation of a multiset of indivisible items. *arXiv preprint arXiv:2202.05186*, 2022.

[Lee, 2015] Euiwoong Lee. APX-hardness of maximizing Nash social welfare with indivisible items. *Information Processing Letters*, 122, 07 2015.

[Lipton *et al.*, 2004] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

[Mahara, 2021] Ryoga Mahara. Extension of additive valuations to general valuations on the existence of EFX. In *29th Annual European Symposium on Algorithms (ESA)*, 2021.

[Mas-Colell *et al.*, 1995] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[Plaut and Roughgarden, 2020] Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2):1039–1068, 2020.

[Walsh, 2020] Toby Walsh. Fair division: The computer scientist's perspective. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4966–4972, 2020.

[Zhou and Wu, 2022] Shengwei Zhou and Xiaowei Wu. Approximately EFX allocations for indivisible chores. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, (IJCAI)*, pages 783–789, 2022.