Fair and Efficient Allocations of Chores under Bivalued Preferences

Jugal Garg, Aniket Murhekar, John Qin

University of Illinois, Urbana-Champaign
{jugal, aniket2, johnqin2}@illinois.edu

Abstract

We study the problem of fair and efficient allocation of a set of indivisible *chores* to agents with additive cost functions. We consider the popular fairness notion of envy-freeness up to one good (EF1) with the efficiency notion of Pareto-optimality (PO). While it is known that an EF1+PO allocation exists and can be computed in pseudo-polynomial time in the case of goods, the same problem is open for chores.

Our first result is a strongly polynomial-time algorithm for computing an EF1+PO allocation for *bivalued instances*, where agents have (at most) two disutility values for the chores. To the best of our knowledge, this is the first non-trivial class of indivisible chores to admit an EF1+PO allocation and an efficient algorithm for its computation.

We also study the problem of computing an envy-free (EF) and PO allocation for the case of divisible chores. While the existence of EF+PO allocation is known via competitive equilibrium with equal incomes, its efficient computation is open. Our second result shows that for bivalued instances, an EF+PO allocation can be computed in strongly polynomial-time.

1 Introduction

The problem of fair division is concerned with allocating items to agents in a *fair* and *efficient* manner. Formally introduced by (Steinhaus 1949), fair division is an active area of research studied across fields like computer science and economics. Most work has focused on the fair division of *goods*: items which provide non-negative *value* (or utility) to the agents to whom they are allocated. However, several practical scenarios involve *chores* (or bads). Chores are items which impose a *cost* (or disutility) to the agent to whom they are allocated. For instance, household chores such as cleaning and cooking often need to be fairly distributed among members of the household. Likewise, teachers have to divide teaching load, stakeholders have to divide liabilities upon dissolution of a firm, etc. These examples highlight the importance of allocating *chores* in a fair and efficient manner. Agencies responsible for designing such allocations must take into account the differences in preferences of agents in order for the allocation to be acceptable to all those involved.

Arguably, the most popular notion of fairness is envy-freeness (EF) (Foley 1967; Varian 1974), which requires that every agent weakly prefers the bundle of items allocated to her over the bundle of any other agent. When items are *divisible*, i.e., can be shared among agents, EF allocations are known to exist. However, in the case of *indivisible* items, EF allocations need not exist. For instance, while dividing one chore between two agents, the agent who is assigned the chore will envy the other. Since the fair division of indivisible items remains an important problem, several relaxations of envy-freeness have been defined, first in the context of goods, and later adapted to chores.

A widely studied relaxation of envy-freeness is *envy-freeness up to one item* (EF1), defined by (Budish 2011) in the context of goods. For chores, an allocation is said to be EF1 if for every agent, the envy disappears after removing one chore assigned to her. It is known that an EF1 allocation of chores exists and can be efficiently computed (Lipton et al. 2004; Bhaskar, Sricharan, and Vaish 2020). However, an EF1 allocation may be highly inefficient. Consider for example two agents A_1 and A_2 and 2 chores j_1 and j_2 where A_i has almost zero cost for j_i and high cost for the other chore. The allocation in which j_1 is assigned to A_2 and j_2 is assigned to A_1 is clearly EF1. However both agents incur high cost, which is highly inefficient. The allocation in which j_i is assigned to A_i is more desirable since it is both fair as well as efficient.

The standard notion of economic efficiency is Pareto optimality (PO). An allocation is said to be PO if no other allocation makes an agent better off without making someone else worse off. Fractional Pareto optimality (fPO) is a stronger notion requiring that no other fractional allocation makes an agent better off without making someone else worse off. Every fPO allocation is therefore PO, but not vice-versa.

An important question is whether the fairness and efficiency notions of EF1 and PO (or fPO) can be achieved in conjunction, and if so, can they be computed in polynomial-time. For the case of goods, (Barman, Krishnamurthy, and Vaish 2018a) showed that an EF1+PO allocation exists and can be computed in pseudopolynomial-time. Improving this result, (Murhekar and Garg 2021) showed that an EF1+fPO allocation can be computed in pseudopolynomial-time. For the case of chores, it is unclear whether EF1+PO allocations even exist, except for simple cases like identical valuations.

Settling the existence of EF1+PO allocations (and developing algorithms for computing them) has turned out to be a challenging open problem.

In this paper, we present the first non-trivial results on the EF1+PO problem for chores. We study the class of *bivalued* instances, where there are only two costs, i.e., for every agent a chore costs either a or b, for two positive numbers[1] a and b. Bivalued instances are a well-studied class in the fair division literature, we list several works in Section 1.1. In particular, (Amanatidis et al. 2020) showed that allocations that are envy-free up to *any* good (EFX), which is a strengthening of EF1, can be efficiently computed for bivalued goods. Recently, (Garg and Murhekar 2021) showed that EFX+fPO allocations can be computed in polynomial-time for bivalued goods. Showing positive results for bivalued chores, our first result is:

Result 1. *For bivalued instances with n agents and m indivisible chores, an EF1+fPO allocation exists and can be computed in* $\mathsf{poly}(n, m)$*-time.*

Next, we study the problem of computing an EF+PO allocation of divisible chores. For goods, it is known that an EF+PO allocation always exists (Varian 1974) and is in fact polynomial-time computable via the Eisenberg-Gale convex program (Nisan et al. 2007). This is done by computing the competitive equilibrium with equal incomes (CEEI). Here, the idea is to provide each agent with the same amount of fictitious money and then find prices and an allocation of items such that all items are completely bought and each agent buys her most preferred bundle subject to their budget constraint. This is an example of a *market* where demand (of agents) equals supply (of items), and is known as the Fisher market. For goods, there are polynomial-time algorithms for computing the competitive equilibrium (CE) (Devanur et al. 2008; Orlin 2010). For chores, the problem is harder: (Bogomolnaia et al. 2017) showed that the CE rule can be non-convex, multi-valued and disconnected. Algorithms with exponential run-times are known for computing CE for chores (Brânzei and Sandomirskiy 2019; Garg and McGlaughlin 2020; Chaudhury et al. 2021), but designing a polynomial-time algorithm is an open problem. Working towards this goal, our second result shows:

Result 2. *For bivalued instances with n agents and m divisible chores, an EF+PO allocation can be computed in* $\mathsf{poly}(n, m)$*-time.*

1.1 Further Related Work

(Barman, Krishnamurthy, and Vaish 2018a) showed that for n agents and m goods, an EF1+PO allocation can be computed in time $\mathsf{poly}(n, m, V)$, where V is the maximum utility value. Their algorithm first perturbs the values to a desirable form, and then computes an EF1+fPO allocation for the perturbed instance, which for a small-enough perturbation is EF1+PO for the original instance. Their approach is via *integral market equilibria*, which guarantees fPO at every step, and the concept of price-envy-freeness up to one good (pEF1) which is a strengthening of EF1. Using similar tools, (Murhekar and Garg 2021) showed that an EF1+fPO allocation can be computed in $\mathsf{poly}(n, m, V)$-time. They also showed that an EF1+fPO allocation can be computed in $\mathsf{poly}(n, m)$-time for k-ary instances (agents have at most k values for the goods) where k is a constant. It may seem a natural idea to try and use these approaches for chores, however they do not extend easily. While our algorithm also uses integral market equilibria to obtain the fPO property and pEF1 for chores to argue about EF1, our algorithm and its analysis are much more involved and significantly different from previous works.

Bivalued preferences are a well-studied class in literature. The following results are for the goods setting. (Aziz et al. 2019) showed PO is efficiently verifiable for bivalued instances and coNP-hard for 3-valued instances; (Aziz 2020), and (Vazirani and Yannakakis 2021) studied the Hylland-Zeckhauser scheme for probabilistic assignment of goods in bivalued instances; and (Bogomolnaia and Moulin 2004) studied matching problems with bivalued (dichotomous) preferences. More generally, instances with few values have also been studied: (Barman, Krishnamurthy, and Vaish 2018b) showed that EF1+PO allocations can be computed for binary valuations; (Babaioff, Ezra, and Feige 2021) studied truthful mechanisms for dichotomous valuations; (Golovin 2005) presented approximation algorithms and hardness results for computing max-min fair allocations in 3-valued instances; (Bliem, Bredereck, and Niedermeier 2016) studied fixed-parameter tractability for computing EF+PO allocations with parameter $n + z$, where z is the number of values; and (Garg et al. 2009) studied leximin assignments of papers ranked by reviewers on a small scale, in particular they present an efficient algorithm for 2 ranks, i.e., "high or low interest" and show NP-hardness for 3 ranks. Such instances have also been studied in resource allocation contexts, including makespan minimization with 2 or 3 job sizes (Woeginger 1997; Chakrabarty, Khanna, and Li 2015).

The fairness notion of equitability requires that each agent get the same amount of utility or disutility. Similar to EF1 and envy-freeness up to any item (EFX), equitability up to one (resp. any) item (EQ1 (resp. EQX)) are relaxations of equitability. Using approaches inspired by (Barman, Krishnamurthy, and Vaish 2018a), pseudopolynomial time algorithms for computing EQ1+PO allocations were developed for both goods (Freeman et al. 2019) and chores (Freeman et al. 2020). (Garg and Murhekar 2021) showed that for bivalued goods both an EQX+PO allocation and an EFX+PO allocation are polynomial time computable.

2 Preliminaries

Problem instance. A *fair division instance* (of chores) is a tuple (N, M, C), where $N = [n]$ is a set of $n \in \mathbb{N}$ agents, $M = [m]$ is a set of $m \in \mathbb{N}$ indivisible chores, and $C = \{c_1, \ldots, c_n\}$ is a set of cost or disutility functions, one for each agent $i \in N$. Each cost function $c_i : M \to \mathbb{R}_{\geq 0}$ is specified by m numbers $c_{ij} \in \mathbb{R}_{\geq 0}$, one for each chore $j \in M$, which denotes the cost agent i has for performing

[1] We can assume the two values are positive since one of them being zero implies the setting is *binary*, in which case computing an EF1+PO allocation is trivial by first assigning chores to agents which have 0 cost for them, and then allocating almost equal number of chores of non-zero cost to everyone.

(receiving) chore j. We assume that the cost functions are additive, that is, for every agent $i \in N$, and for $S \subseteq M$, $c_i(S) = \sum_{j\in S} c_{ij}$. For notational ease, we write $c(S \setminus j)$ instead of $c(S \setminus \{j\})$.

We call a fair division instance (N, M, C) a *bivalued instance* if there exist $a, b \in \mathbb{R}^+$, with $a \geq b$, such that for all $i \in N$ and $j \in M$, $c_{ij} \in \{a, b\}$. That is, the cost of any chore to any agent is one of at most two given positive numbers. We can assume w.l.o.g. for bivalued instances that all costs $c_{ij} \in \{1, k\}$, where $k = a/b \geq 1$, by re-scaling the costs. Given such an instance, we partition the set of chores into sets of low-cost chores M_{low} and high-cost chores M_{high}:

- $M_{low} = \{j \in M : \exists i \in N \text{ s.t. } c_{ij} = 1\}$, and
- $M_{high} = \{j \in M : \forall i \in N,\, c_{ij} = k\}$.

We can additionally assume for bivalued instances that for every agent i, there is at least one chore j s.t. $c_{ij} = 1$. This is w.l.o.g., since if $c_{ij} = k$ for all $j \in M$, then we can re-scale costs to set $c_{ij} = 1$ for all $j \in M$.

Allocation. An *allocation* $\mathbf{x}$ of chores to agents is a n-partition of the chores $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where agent i is allotted $\mathbf{x}_i \subseteq M$, and gets a total cost of $c_i(\mathbf{x}_i)$. A *fractional allocation* $\mathbf{x} \in [0, 1]^{n\times m}$ is a fractional assignment such that for each chore $j \in M$, $\sum_{i\in N} x_{ij} = 1$. Here, $x_{ij} \in [0, 1]$ denotes the fraction of chore j allotted to agent i.

Fairness notions. An allocation $\mathbf{x}$ is said to be *envy-free up to one chore* (EF1) if for all $i, h \in N$, where $\mathbf{x}_i \neq \varnothing$, there exists a chore $j \in \mathbf{x}_i$ such that $c_i(\mathbf{x}_i \setminus j) \leq c_i(\mathbf{x}_h)$. We say that an agent i *EF1-envies* an agent h if for all $j \in \mathbf{x}_i$, $c_i(\mathbf{x}_i \setminus j) > c_i(\mathbf{x}_h)$, i.e., the EF1 condition between i and h is violated.

A (fractional) allocation $\mathbf{x}$ is said to be *envy-free* if for all $i, h \in N$, $c_i(\mathbf{x}_i) \leq c_i(\mathbf{x}_h)$. We say that an agent i *envies* an agent h if $c_i(\mathbf{x}_i) > c_i(\mathbf{x}_h)$, i.e., the EF condition between i and h is violated.

Pareto-optimality. An allocation $\mathbf{y}$ dominates an allocation $\mathbf{x}$ if $c_i(\mathbf{y}_i) \leq c_i(\mathbf{x}_i), \forall i$ and there exists h s.t. $c_h(\mathbf{y}_h) < c_h(\mathbf{x}_h)$. An allocation is said to be *Pareto optimal* (PO) if no allocation dominates it. Further, an allocation is said to be fractionally PO (fPO) if no fractional allocation dominates it. Thus, a fPO allocation is PO, but not vice-versa.

Fisher markets. A Fisher market or a *market instance* is a tuple (N, M, C, e), where the first three terms are interpreted as before, and $e = \{e_1, \ldots, e_n\}$ is the set of agents' *mimimum payments*, where $e_i \geq 0$, for each $i \in N$. In this model, chores can be allocated fractionally. Given a payment vector, also called a *price*[2] vector, $\mathbf{p} = (p_1, \ldots, p_m)$, each chore j pays p_j per unit of chore. Agents perform chores in exchange for payment. Given chore payments, each agent i aims to obtain the set of chores that minimizes her total cost subject to her payment constraint, i.e., receiving a total payment of at least e_i.

A *market outcome* is a (fractional) allocation $\mathbf{x}$ of the chores to the agents and a set of prices $\mathbf{p}$ of the chores. The *spending* of an agent i under the market outcome $(\mathbf{x}, \mathbf{p})$ is given by $\mathbf{p}(\mathbf{x}_i) = \sum_{j\in M} p_j x_{ij}$. For an agent i, we define the *bang-per-buck* ratio α_{ij} of good j as c_{ij}/p_j, and the *minimum bang-per-buck* (mBB) ratio $\alpha_i = \min_j \alpha_{ij}$. We define $\text{mBB}_i = \{j \in M : c_{ij}/p_j = \alpha_i\}$, called the *mBB-set*, to be the set of chores that give mBB to agent i at prices $\mathbf{p}$. A market outcome $(\mathbf{x}, \mathbf{p})$ is said to be *'on mBB'* if for all agents i and chores j, $x_{ij} > 0 \Rightarrow j \in \text{mBB}_i$. For integral $\mathbf{x}$, this means that $\mathbf{x}_i \subseteq \text{mBB}_i$ for all $i \in N$.

A market outcome $(\mathbf{x}, \mathbf{p})$ is said to be a *market equilibrium* if (i) the market clears, i.e., all chores are fully allocated. Thus, for all j, $\sum_{i\in N} x_{ij} = 1$, (ii) each agent receives their minimum payment, for all $i \in N$, $\sum_{j\in M} x_{ij} p_j = e_i$, and, (iii) agents only receive chores that give them minimum bang-per-buck, i.e., $(\mathbf{x}, \mathbf{p})$ is on mBB.

Given a market outcome $(\mathbf{x}, \mathbf{p})$ with $\mathbf{x}$ integral, we say it is *price envy-free up to one chore* (pEF1) if for all $i, h \in N$ there is a chore $j \in \mathbf{x}_i$ such that $\mathbf{p}(\mathbf{x}_i \setminus j) \leq \mathbf{p}(\mathbf{x}_h)$. We say that an agent i *pEF1-envies* an agent h, if for all $j \in \mathbf{x}_i$, $\mathbf{p}(\mathbf{x}_i \setminus j) > \mathbf{p}(\mathbf{x}_h)$, i.e., the pEF1 condition between i and h is violated. For integral market outcomes on mBB, the pEF1 condition implies the EF1 condition.

Lemma 1. *Let* $(\mathbf{x}, \mathbf{p})$ *be an integral market outcome on mBB. If* $(\mathbf{x}, \mathbf{p})$ *is pEF1 then* $\mathbf{x}$ *is EF1 and fPO.*

Proof. We first show that $(\mathbf{x}, \mathbf{p})$ forms a market equilibrium for the Fisher market instance (N, M, C, e), where for every $i \in N$, $e_i = \mathbf{p}(\mathbf{x}_i)$. It is easy to see that the market clears and each agent receives their minimum payment. Further $\mathbf{x}$ is on mBB as per our assumption. Now the fact that $\mathbf{x}$ is fPO follows from the First Welfare Theorem (Mas-Colell, Whinston, and Green 1995), which shows that for any market equilibrium $(\mathbf{x}, \mathbf{p})$, the allocation $\mathbf{x}$ is fPO.

Since $(\mathbf{x}, \mathbf{p})$ is pEF1, for all pairs of agents $i, h \in N$, there is some chore $j \in \mathbf{x}_i$ s.t. $\mathbf{p}(\mathbf{x}_i \setminus j) \leq \mathbf{p}(\mathbf{x}_h)$. Since $(\mathbf{x}, \mathbf{p})$ is on mBB, $\mathbf{x}_i \subseteq \text{mBB}_i$. Let α_i be the mBB-ratio of i at the prices $\mathbf{p}$. By definition of mBB, $c_i(\mathbf{x}_i \setminus j) = \alpha_i \mathbf{p}(\mathbf{x}_i \setminus j)$, and $c_i(\mathbf{x}_h) \geq \alpha_i \mathbf{p}(\mathbf{x}_h)$. Combining these implies $\mathbf{x}$ is EF1. $\square$

We now define least spenders as agents with minimum spending, and big spenders as agents with maximum spending after the removal of their highest-priced chore.

Definition 1 (Least and big spenders)**.** An agent $\ell \in \mathsf{argmin}_{i\in N} \mathbf{p}(\mathbf{x}_i)$ is referred to as a *least spender* (LS). An agent $b \in \mathsf{argmax}_{i\in N} \min_{j\in \mathbf{x}_i} \mathbf{p}(\mathbf{x}_i \setminus j)$ is referred to as a *big spender* (BS).

We break ties arbitrarily to decide a unique LS and BS. Together with Lemma 1, the following lemma shows that in order to obtain an EF1 allocation, it is sufficient to focus on the pEF1-envy the big spender has towards the least spender.

Lemma 2. *Let* $(\mathbf{x}, \mathbf{p})$ *be an integral market outcome on mBB. If* $\mathbf{x}$ *is not EF1, then the big spender* b *pEF1-envies the least spender* ℓ*.*

Proof. If $\mathbf{x}$ is not EF1, then Lemma 1 implies that $\mathbf{x}$ is not pEF1. Hence there is a pair of agents i, h s.t. for every chore $j \in \mathbf{x}_i$, $\mathbf{p}(\mathbf{x}_i \setminus j) > \mathbf{p}(\mathbf{x}_h)$. By definition of big spender, we know $\mathbf{p}(\mathbf{x}_b \setminus j') \geq \mathbf{p}(\mathbf{x}_i \setminus j)$, for some $j' \in \mathbf{x}_b$. By definition

[2] We refer to payments as prices for sake of similarity with the Fisher market model in the goods case.

of least spender, $\mathbf{p}(\mathbf{x}_h) \geq \mathbf{p}(\mathbf{x}_\ell)$. Putting these together we get $\mathbf{p}(\mathbf{x}_b \setminus j') > \mathbf{p}(\mathbf{x}_\ell)$, implying that b pEF1-envies ℓ. $\square$

Given a market outcome $(\mathbf{x}, \mathbf{p})$, we define the mBB graph to be a bipartite graph $G = (N, M, E)$ where for an agent i and chore j, $(i, j) \in E$ iff $j \in \text{mBB}_i$. Further, an edge (i, j) is called an *allocation edge* if $j \in \mathbf{x}_i$, otherwise it is called an *mBB* edge.

For agents $i_0, \dots, i_\ell$ and chores $j_1, \dots, j_\ell$, a path $P = (i_0, j_1, i_1, j_2, \dots, j_\ell, i_\ell)$ in the mBB graph, where for all $1 \leq \ell' \leq \ell$, $j_{\ell'} \in \mathbf{x}_{i_{\ell'-1}} \cap \text{mBB}_{i_{\ell'}}$, is called a *special path*. We define the *level* $\lambda(h; i_0)$ of an agent h w.r.t. i_0 to be half the length of the shortest special path from i_0 to h, and to be n if no such path exists. A path $P = (i_0, j_1, i_1, j_2, \dots, j_\ell, i_\ell)$ is an *alternating path* if it is special, and if $\lambda(i_0; i_0) < \lambda(i_1; i_0) \cdots < \lambda(i_\ell; i_0)$, i.e., the path visits agents in increasing order of their level w.r.t. i_0. Further, the edges in an alternating path alternate between allocation edges and mBB edges. Typically, we consider alternating paths starting from a big spender agent.

Definition 2 (Component C_i of a big spender i)**.** For a big spender i, define C_i^ℓ to be the set of all chores and agents which lie on alternating paths of length ℓ. Call $C_i = \bigcup_\ell C_i^\ell$ the *component* of i, the set of all chores and agents reachable from the big spender i through alternating paths.

3 EF1+fPO Allocation of Indivisible Chores

In this section, we present our main result:

Theorem 1. *Given a bivalued fair division instance (N, M, C) of indivisible chores with all $c_{ij} \in \{a, b\}$ for some $a, b \in \mathbb{R}^+$, an EF1+fPO allocation can be computed in strongly polynomial-time.*

We prove Theorem 1 by showing that our Algorithm 2 computes an EF1+fPO allocation in polynomial-time. For ease of presentation and due to space constraints, proofs of this section appear in the full version of the paper (Garg, Murhekar, and Qin 2021).

3.1 Obtaining Initial Groups

Recall that we can scale the costs so that they are in $\{1, k\}$. The first step of Algorithm 2 is to obtain a partition of the set N of agents into *groups* $N_1, \dots, N_R$ with desirable properties. For this, we use Algorithm 1 (called MakeInitGroups).

Algorithm 1 starts with a cost-minimizing market outcome $(\mathbf{x}, \mathbf{p})$ where each chore j is assigned to an agent who has minimum cost for j. This ensures the allocation is fPO. The chore prices are set as follows. Each low-cost chore j is assigned to an agent i s.t. $c_{ij} = 1$. If an agent values all chores at k, then we can re-scale all values to 1. Each low-cost chore is priced at 1, and each high-cost chore is priced at k. This pricing ensures that the mBB ratio of every agent is 1. The algorithm then eliminates pEF1-envy from the component of the big spender b by identifying an agent i in C_b that is pEF1-envied by b, and transferring an additional chore j_ℓ to i from an agent $h_{\ell-1}$ who lies along a shortest alternating path from b to i (Lines 7 & 8). Note that the identity of the big spender may change after transferring j_ℓ if j_ℓ belonged to b, so we must check the identity of the big spender after each transfer (Line 10). Once the component of the current big spender b is pEF1, the same process is applied to the next big spender outside the previously made components. Repeated application of this process leaves us with disjoint partial components $H_1, \dots, H_R$ of agent sets $N_1, \dots, N_R$, where $R \leq n$, all of which are pEF1. We refer to $N_1, \dots, N_R$ as agent *groups*, and $H_1, \dots, H_R$ as initial (partial) components. Note also that the spending (up to the removal of the biggest chore) of the big spender h_r of H_r is weakly decreasing with r. We now record several properties of the output of Algorithm 1 .

Algorithm 1: MakeInitGroups

Input: Fair division instance (N, M, C) with $c_{ij} \in \{1, k\}$
Output: Integral alloc. $\mathbf{x}$, prices $\mathbf{p}$, agent groups $\{N_r\}_{r \in [R]}$

```
1:  (x, p) ← initial cost minimizing integral market allocation, where p_j = c_ij for j ∈ x_i.
2:  R ← 1, N' ← N
3:  while N' ≠ ∅ do
4:      b ← argmax_{i∈N'} min_{j∈x_i} p(x_i \ j)        ▷ Big Spender
5:      C_b ← Component of b                            ▷ See Definition 2
6:      while ∃ agent i ∈ C_b s.t. ∀j ∈ x_b, p(x_b \ j) > p(x_i)
7:          Let (b, j_1, h_1, j_2, ..., h_{ℓ−1}, j_ℓ, i) be the shortest alternating path from b to i
8:          x_{h_{ℓ−1}} ← x_{h_{ℓ−1}} \ {j_ℓ}              ▷ Chore transfer
9:          x_i ← x_i ∪ {j_ℓ}
10:         b ← argmax_{i∈N'} min_{j∈x_i} p(x_i \ j)
11:     H_R ← C_b ∩ (N' ∪ x_{N'})                       ▷ Partial component
12:     N_R ← H_R ∩ N                                   ▷ Agent group
13:     N' ← N' \ N_R, R ← R + 1
14: return (x, p, {N_r}_{r∈[R]})
```

Lemma 3. *Algorithm 1 returns in* poly(n, m)*-time a market outcome $(\mathbf{x}, \mathbf{p})$ with agents partitioned into groups $N_1, \dots, N_R$, with the following properties:*

(i) For all low-cost chores $j \in M$, $p_j = 1$, and for all high-cost chores $j \in M$, $p_j = k$.
(ii) The mBB ratio α_i of every agent i is 1.
(iii) Let H_r be the collection of agents N_r and chores allocated to them in $(\mathbf{x}, \mathbf{p})$. Then each H_r is a *partial component* of some agent. That is, for each $r \in [R]$, there is an agent $h_r \in H_r$ s.t. H_r comprises of all agents and chores not in $\bigcup_{r' < r} H_{r'}$ reachable through alternating paths from h_r. Further, h_r is the big spender among agents not in $\bigcup_{r' < r} H_{r'}$:
$$h_r \in \text{argmax}_{i \notin (\bigcup_{r' < r} H_{r'})} \min_{j \in \mathbf{x}_i} \mathbf{p}(\mathbf{x}_i \setminus j)$$
(iv) The spending (up to removal of the largest chore) $f(r)$ of the big spender in H_r weakly decreases with r. Here $f(r) = \max_{i \in H_r} \min_{j \in \mathbf{x}_i} \mathbf{p}(\mathbf{x}_i \setminus j)$.
(v) Each *group is pEF1*, i.e., an agent does not pEF1-envy other agents in the same group.
(vi) For every $i \in H_r$ and $j \in H_{r'}$ with $r' < r$, $c_{ij} = k$.
(vii) All high-cost chores belong to H_R.

We have that:

Algorithm 2: Computing an EF1+fPO allocation

Input: Fair division instance (N, M, C) with $c_{ij} \in \{1, k\}$
Output: An integral allocation $\mathbf{x}$

1: $(\mathbf{x}, \mathbf{p}, \{N_r\}_{r\in[R]}) \leftarrow \mathsf{MakeInitGroups(N, M, C)}$
2: $U \leftarrow [R]$ ▷ Unraised groups
3: $\mathbf{x}^0 \leftarrow \mathbf{x}$ ▷ Copy of initial allocation, used in Line 21
4: $b \leftarrow \mathsf{argmax}_{i\in N} \min_{j\in \mathbf{x}_i} \mathbf{p}(\mathbf{x}_i \setminus j)$ ▷ Big Spender
5: $\ell \leftarrow \mathsf{argmin}_{i\in N} \mathbf{p}(\mathbf{x}_i)$ ▷ Least Spender
6: Maintain (r, s) s.t. $b \in H_r$, $\ell \in H_s$
7: **while** $(\mathbf{x}, \mathbf{p})$ is not pEF1 **and** $s \in U$ **do**
8:     **if** $r \in U$ **then**
9:         Raise prices of chores in H_r by a factor of k
10:         $U \leftarrow U \setminus \{r\}$
11:     **else**
12:         Transfer a chore from b to ℓ along an mBB edge
13:     $b \leftarrow \mathsf{argmax}_{i\in N} \min_{j\in \mathbf{x}_i} \mathbf{p}(\mathbf{x}_i \setminus j)$
14:     $\ell \leftarrow \mathsf{argmin}_{i\in N} \mathbf{p}(\mathbf{x}_i)$
15: **while** $(\mathbf{x}, \mathbf{p})$ is not pEF1 **do**
16:     **if** $s > r$ **then**
17:         Transfer a chore from b to ℓ along an mBB edge
18:     **else if** $s < r$ **then**
19:         $\exists\, i \in N_{r'}$ s.t. $r' \in U$ and $\exists\, j \in \mathbf{x}_i$ s.t. $j \in \mathbf{x}^0_\ell$
20:         Transfer j from i to ℓ
21:         Transfer a chore from b to i
22:     $b \leftarrow \mathsf{argmax}_{i\in N} \min_{j\in \mathbf{x}_i} \mathbf{p}(\mathbf{x}_i \setminus j)$
23:     $\ell \leftarrow \mathsf{argmin}_{i\in N} \mathbf{p}(\mathbf{x}_i)$
24: **return** $(\mathbf{x}, \mathbf{p})$

Lemma 4. *Algorithm 1 terminates in time* $\mathsf{poly}(n, m)$.

3.2 Overview of Algorithm 2

Our main algorithm (Algorithm 2) begins by calling Algorithm 1, which returns a market outcome $(\mathbf{x}, \mathbf{p})$ and a set of agent groups $\{N_r\}_{r\in[R]}$ (with associated partial components $\{H_r\}_{r\in[R]}$) satisfying properties in Lemma 3. In the subsequent discussion, we refer to $(\mathbf{x}, \mathbf{p})$ as the *initial allocation*. Also in the subsequent discussion, all mentions of an agent receiving or losing chores are relative to this initial allocation.

The following is an important invariant of Algorithm 2.

Lemma 5. *The spending of the least spender does not decrease in the run of Algorithm 2.*

We say that a group N_r is above (resp. below) group N_s if $r < s$ (resp. $r > s$). Lemma 3 shows that each group N_r is initially pEF1. Hence if the initial allocation $(\mathbf{x}, \mathbf{p})$ is not pEF1, then the big spender b and the least spender ℓ must be in different components. Since $b \in H_1$, it must be the case that $\ell \in H_s$ for some $s > 1$. Since we want to obtain an fPO allocation, we can only transfer along mBB edges. Hence we raise the prices of all chores in H_1. We show that doing so creates an mBB edge from all agents $i \notin H_1$ to all $j \in H_1$ (Lemma 9 below). In particular, there is an mBB edge from ℓ to a chore assigned to b. Hence we transfer a chore *directly* from b to ℓ, thus reducing the pEF1-envy of b. This may change the identity of the big and least spenders. If the allocation is not yet pEF1, we must continue this process.

At an arbitrary step in the run of the algorithm, let b and ℓ be the big and least spenders. If the allocation is not pEF1, then b pEF1-envies ℓ (Lemma 2). We consider cases based on the relative positions of b and ℓ. First we argue that b and ℓ cannot lie in the same group, by showing that:

Lemma 6. *Throughout the run of Algorithm 2, each group N_r remains pEF1.*

Hence b and ℓ must lie in different groups. Once again, since we want to transfer chores away from b to reduce the pEF1-envy, and we want to obtain an fPO allocation, we only transfer chores along mBB edges. Doing so may require us to raise the prices of all chores belonging to certain agents in order to create new mBB edges to facilitate chore transfer. In our algorithm, *all* agents in a group undergo price-rise together. We call a group N_r a *raised group* if its agents have undergone price-rise, else it is called an *unraised group*. The set U (Line 2) records the set of unraised components.

We will use the terms *time-step* or *iteration* interchangeably to denote either a chore transfer or a price-rise step. We say 'at time-step t', to refer to the state of the algorithm *just before* the event at t happens. We denote by $(\mathbf{x}^t, \mathbf{p}^t)$ the allocation and price vector at time-step t.

Let T be the first time-step that the current LS enters a raised group. Note that such an event may or may not happen. Our algorithm performs instructions in Lines 7-14 before T, and Lines 15-23 after T, as we describe below.

3.3 Algorithm Prior to T (Lines 7-14)

We first record some properties of the algorithm prior to T. These observations directly follow from the algorithm.

Lemma 7. *Prior to T, the following hold:*

1. *Any transfer of chores only takes place directly from the big spender b to the least spender ℓ. Thus, an agent receives a chore only if she is a least spender, and an agent loses a chore only if she is a big spender.*
2. *An agent ceases to be a least spender only if she receives a chore. An agent ceases to be a big spender only if she loses a chore.*
3. *A group undergoes price-rise at t only if the group contains the big spender at t.*

We now have that:

Lemma 8. *If at any point in the run of Algorithm 2 prior to T, the big spender lies in a group which contains a former least spender, then the allocation is pEF1.*

This allows us to show:

Lemma 9. *Prior to T, the following hold:*

(i) Let r be the number of price-rise steps until time-step t, where $t < T$. Then the raised groups are exactly $N_1, \ldots, N_r$. Furthermore they underwent price-rise exactly once and in that order.

(ii) For any chore j allocated to an agent in a raised group N_r and any agent i in an unraised group $N_{r'}$, where $r' > r$, $j \in mBB_i$.

(iii) *For each $r' \in [r]$, at the time of price-rise of $N_{r'}$, no agent in $N_{r'}$ has either received or lost a chore since the initial allocation.*

Proof. We prove (i), (ii) and (iii) by induction. For $r = 0$, they are trivially true since there are no raised groups. Assume that at some time-step t, (i) groups $N_1, \ldots, N_r$ have undergone price-rise, once and in that order, for some $r \geq 1$, and (ii) and (iii) hold.

Note that our algorithm only raises the prices of chores owned by a group if the group contains the big spender at the time (Lemma 7). If the current BS is in a raised group, then the induction hypothesis ensures that there is an mBB edge from the LS (who is in an unraised group prior to T) to chores owned by the BS. The algorithm therefore performs a direct chore transfer and no price-rise is necessary.

If eventually the BS enters an unraised group, then a price-rise step is potentially necessary. Suppose b is an agent who has received a chore prior to t. If this happens then b must have been a former LS by Lemma 7. Then Lemma 8 shows that the allocation must already be pEF1.

Hence we assume b has not received a new chore. Furthermore since b pEF1-envies the LS ℓ, it must be the case that $\ell \in N_s$ where $s \geq r+2$. Lemma 3 shows that there is no mBB edge from ℓ to chores owned by b, hence a direct chore transfer is not possible and it is necessary for N_{r+1} to undergo price-rise. This shows (i).

Now if an agent $i \in N_{r+1}$ had previously received a chore, then i is a former LS at t. Lemma 8 shows that b is in a group containing a former LS, implying that the allocation must be pEF1. Similarly Lemma 7 shows that no agent in N_{r+1} can have lost a chore. This is because only BS agents lose chores. Prior to t, no agent of N_{r+1} can be BS. Hence no agent in N_{r+1} has received or lost a chore since the initial allocation at the time N_{r+1} undergoes price-rise, thus showing (iii).

The algorithm next raises the prices of all chores owned by N_{r+1} by a factor of k, and N_{r+1} becomes a raised group. Consider an agent $i \in N_{r'}$ for $r' \geq r+2$ and a chore allocated to an agent in N_{r+1}. Since the mBB ratio only changes upon a price-rise, the mBB ratio of i is 1 since $N_{r'}$ does not undergo a price-rise before N_{r+1}.

Observe that since $i \notin N_{r+1}$, there is no alternating path from agents in N_{r+1} to i. Hence $j \notin \text{mBB}_i$ before the price-rise. Thus $c_{ij}/p_j^t > 1$, showing $c_{ij} = k$ and $p_j^t = 1$. After the price-rise, we have that $p_j^{t+1} = k$, and $\alpha_i = c_{ij}/p_j^{t+1}$. Thus, $j \in \text{mBB}_i$ after the price-rise, which shows (ii). □

To summarize the behavior of the Algorithm prior to T, we have argued in the above proof that if the allocation is not pEF1, we can always either (i) transfer a chore directly from b to ℓ, or (ii) perform a price-rise on the group of b and then transfer a chore from b to ℓ. Further, we argue that the Algorithm makes progress towards getting a pEF1 allocation.

Lemma 10. *Algorithm 2 performs at most* $\mathsf{poly}(n, m)$ *steps prior to T.*

Proof. Prior to T, the LS always remains in an unraised group. Chores are transferred away from agents who become big spenders in raised groups. Once an agent undergoes price-rise, she cannot gain any additional chores, since doing so would mean she is the LS in a raised group, which cannot happen prior to T. When the BS is in an unraised group, the group undergoes a price-rise. Thus, effectively, either agents in raised components only lose chores, or the BS 'climbs-down' in the group list $N_1, \ldots, N_R$, while the LS remains below the BS. Since there are $R \leq n$ groups, and at most m chores allocated to raised groups, after $\mathsf{poly}(n, m)$ steps either of two events happen: (i) the LS and BS both belong to the same group, or (ii) the LS enters a raised group. In the former case, the allocation is pEF1 due to Lemma 6, and the algorithm terminates in $\mathsf{poly}(n, m)$ steps. We discuss the latter case in the next section. Thus, there are at most $\mathsf{poly}(n, m)$ steps prior to T. □

3.4 Algorithm After T (Lines 15-23)

We now describe the algorithm after T, i.e., once the LS enters a raised group (Lines 15-23). We show that subsequent to T, while the allocation is not pEF1, we can either (i) transfer a chore directly from b to ℓ, or (ii) transfer chores via an alternating path containing 3 agents. We do not perform any price-rises subsequent to T.

From Lemma 9, we know that at T, groups $N_1, \ldots, N_r$ have undergone price-rise, for some $r \in [R]$. Let $N_{<r} = \bigcup_{r'<r} N_{r'}$, and $N_{>r} = \bigcup_{r'>r} N_{r'}$. The allocation at T need not be pEF1, but we argue that it is already very close to being pEF1. Specifically, we show:

Lemma 11. *At T, agents in $N_{<r}$ are pEF1 towards others.*

Lemma 12. *At T, agents in $N_{>r}$ are pEF1 towards others.*

The above two lemmas imply that if the BS is not in N_r, then the allocation is pEF1. Let us assume that the allocation is not pEF1 at T. Let b, the BS at T be in N_r, and ℓ, the LS at T be in $N_{<r}$, since the LS is in a raised group at T.

Suppose ℓ has never lost a chore. Let $\ell \in N_{r'}$, where $r' < r$, and let t' be the time when $N_{r'}$ underwent price-rise. Let b' be the BS at t'. Since the spending of the BS (up to removal of one chore) just after price-rises does not increase, we have:

$$\mathbf{p}^T(\mathbf{x}_b^T \setminus j) \leq \mathbf{p}^{t'}(\mathbf{x}_{b'}^{t'} \setminus j') \leq \mathbf{p}^{t'}(\mathbf{x}_\ell^{t'}) = \mathbf{p}^T(\mathbf{x}_\ell^T),$$

for some chores $j \in \mathbf{x}_b^T, j' \in \mathbf{x}_{b'}^{t'}$. The intermediate transition follows from the property that $N_{r'}$ is pEF1. This shows that the allocation is pEF1.

On the other hand, suppose ℓ has lost at least one chore j prior to T. At T, j must be assigned to some unraised agent i (Lemma 9). Further, there is a chore $j' \in \mathbf{x}_b^T$ s.t. $j' \in \text{mBB}_i$. Thus, $b \xrightarrow{\mathbf{x}} j' \xrightarrow{\text{mBB}} i \xrightarrow{\mathbf{x}} j \xrightarrow{\text{mBB}} \ell$ is an alternating path and we now transfer chores along this path.

Note that as long as ℓ does not own a chore that she initially owned, such a path is available, and such a transfer is possible. If not, then it is as if ℓ has never lost a chore, and in that case the previous argument shows that the allocation must be pEF1.

If after the transfer(s) we do not have pEF1, we identify a new BS and LS and continue this process. We show that:

Lemma 13. *After T, the following are invariant:*

(i) Agents in $N_{<r}$ do not pEF1-envy any other agent.
(ii) Agents in $N_{>r}$ do not pEF1-envy any other agent.
(iii) Each group is pEF1.

Just after T, the BS is in N_r and the LS is in $N_{<r}$. After a chore transfer, the identity of the LS or BS can change. If the BS enters either $N_{<r}$ or $N_{>r}$, then using Lemma 13 the allocation would be pEF1. While the BS is in N_r: (i) if the LS is in N_r, the allocation would be pEF1, (ii) if the LS is in $N_{>r}$, then we can transfer from BS to LS directly along an mBB edge (which exists due to Lemma 9), (iii) if the LS is in $N_{<r}$, then we can transfer from the BS to LS via an alternating path with three agents as described above.

Finally we argue termination in polynomial-time:

Lemma 14. *Algorithm 2 performs at most $\mathsf{poly}(n, m)$ steps after T and terminates with a pEF1 allocation.*

Proof. Call the difference between the spending (up to the removal of the biggest chore) of the big spender and the total spending of the least spender the *spending gap*. If the allocation is not pEF1, the spending gap is positive. After T, there are no price-rises, hence the spending gap weakly decreases. If the allocation is not pEF1, the BS must be in N_r. Based on whether the LS is in $N_{<r}$ or $N_{>r}$, chore transfers which weakly decrease the spending gap are always possible. Further, each transfer reduces the number of chores owned by agents in N_r, and such agents do not receive any chores again. Hence there can only be $\mathsf{poly}(n, m)$ steps after T, eventually terminating in a pEF1 allocation. □

3.5 Summarizing Our EF1+fPO Algorithm

We summarize Algorithm 2.

1. Algorithm 2 first calls Algorithm 1 to partition agents into groups $N_1, \dots, N_R$ with properties as in Lemma 3 to obtain an initial allocation. Lemma 4 shows this takes $\mathsf{poly}(n, m)$ steps.
2. When the current allocation $(\mathbf{x}, \mathbf{p})$ is not pEF1, the BS b pEF1-envies the LS ℓ. While there is an mBB edge from ℓ to a chore owned by b, we transfer a chore directly from ℓ to b. If not, in order to transfer along mBB edges, we may have to raise the prices of chores belonging to the group of b, creating *raised* groups.
3. Let T be the first time-step when the LS enters a raised group. Prior to T, while the allocation is not pEF1, the algorithm either performs a direct chore transfer from the BS to the LS, or performs price-rise on the group of b. Lemma 9 shows that the groups are raised exactly once and in order of $N_1, \dots, N_R$. Lemma 10 shows the algorithm runs for $\mathsf{poly}(n, m)$ steps before T.
4. Once the LS enters a raised group, there are no more price-rise steps. The algorithm performs chore transfers from the BS to LS either directly or via an alternating path with at most 3 agents. Lemma 13 and Lemma 14 show that the algorithm performs at most $\mathsf{poly}(n, m)$ steps after T and terminates with a pEF1 allocation.
5. Finally, we note that the allocation is always fPO, since (i) Algorithm 1 returns a market outcome which is fPO, and (ii) any transfer of chores happens along mBB edges.

4 EF+PO Allocation of Divisible Chores

In this section, we state our second result:

Theorem 2. *Given a bivalued fair division instance (N, M, C) of divisible chores with all $c_{ij} \in \{a, b\}$ for some $a, b \in \mathbb{R}^+$, an EF+PO allocation can be computed in strongly polynomial-time.*

Due to space constraints, details and proofs of this section appear only in the full version of the paper (Garg, Murhekar, and Qin 2021). We first use a balanced flow network formulation (Devanur et al. 2008) to obtain a fractional allocation $\mathbf{x}$, prices $\mathbf{p}$, and agent groups $\{N_i\}_{i\in[R]}$ such that $(\mathbf{x}, \mathbf{p})$ is PO and each N_i is pEF.

We show that there exists some r^* such that raising the prices of the first r^* groups by a factor of $k = a/b$ allows us to obtain a pEF allocation by *draining* chores from the set of all big spenders B (here agents with maximum *total* spending, with no removal) to the set of all least spenders L, when the chores of agents in B are all on mBB for agents in L. That is, we take chores from all agents in B at a uniform rate ρ_B and distribute them among all agents in L at a uniform rate ρ_L. Note, since groups are pEF, if an agent $i \in N_r$ is a BS (LS), then every agent in N_r is also a BS (LS).

Suppose we raise the first r groups. If it is impossible for the algorithm to obtain a pEF allocation, one of two mutually exclusive cases must be true:

1. Raised groups have *too much* total spending. The spending of agents in L rises to spending level of N_r before the spending of agents in B falls to the level of N_r.
2. Raised groups have *too little* total spending. The spending of agents in B falls to the level of N_{r+1} before the spending of agents in L rises to the level of N_{r+1}.

We show that there exists r^* such that raising the first r^* groups results in neither too much nor too little spending. Informally, raising the first group cannot result in too much spending. Inductively, if raising r groups does not result in too much spending, neither does raising $r + 1$ groups. Yet, in raising $R - 1$ groups, we cannot have too little spending. Thus, there must be some $r^* \in [R - 1]$ such that raising the first r^* groups allows for chores to be drained in a manner that results in a pEF allocation. As before, transferring chores only along mBB edges ensures the allocation is PO.

5 Discussion

In this paper, we presented a strongly polynomial-time algorithm for computing an EF1+fPO allocation of chores to agents with bivalued preferences, constituting the first nontrivial result for the EF1+PO problem for chores. Our algorithm is novel and relies on several involved arguments. Given that the general case is a challenging open problem, we believe extending our algorithm and its analysis to the class of k-ary chores is an interesting and natural next step. Another interesting question is whether we can compute an EFX allocation in this setting. We also presented a strongly polynomial-time algorithm for computing an EF+PO allocation of divisible bivalued chores. Computing an EF+PO allocation of divisible chores in polynomial-time is also a compelling direction for future work.

Acknowledgements

Work on this paper is supported by NSF Grant CCF-1942321 (CAREER).

References

Amanatidis, G.; Birmpas, G.; Filos-Ratsikas, A.; Hollender, A.; and Voudouris, A. A. 2020. Maximum Nash Welfare and Other Stories About EFX. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI)*, 24–30.

Aziz, H. 2020. The Hylland-Zeckhauser Rule Under Bi-Valued Utilities. *CoRR*, abs/2006.15747.

Aziz, H.; Biró, P.; Lang, J.; Lesca, J.; and Monnot, J. 2019. Efficient reallocation under additive and responsive preferences. *Theoretical Computer Science*, 790: 1 – 15.

Babaioff, M.; Ezra, T.; and Feige, U. 2021. Fair and Truthful Mechanisms for Dichotomous Valuations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6): 5119–5126.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018a. Finding Fair and Efficient Allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, 557–574.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018b. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 7–13.

Bhaskar, U.; Sricharan, A. R.; and Vaish, R. 2020. On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources. *CoRR*, abs/2012.06788.

Bliem, B.; Bredereck, R.; and Niedermeier, R. 2016. Complexity of Efficient and Envy-Free Resource Allocation: Few Agents, Resources, or Utility Levels. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 102–108.

Bogomolnaia, A.; and Moulin, H. 2004. Random Matching Under Dichotomous Preferences. *Econometrica*, 72(1): 257–279.

Bogomolnaia, A.; Moulin, H.; Sandomirskiy, F.; and Yanovskaya, E. 2017. Competitive Division of a Mixed Manna. *Econometrica*, 85(6): 1847–1871.

Brânzei, S.; and Sandomirskiy, F. 2019. Algorithms for Competitive Division of Chores. *CoRR*, abs/1907.01766.

Budish, E. 2011. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, 119(6): 1061 – 1103.

Chakrabarty, D.; Khanna, S.; and Li, S. 2015. On (1, e)-Restricted Assignment Makespan Minimization. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1087–1101.

Chaudhury, B. R.; Garg, J.; McGlaughlin, P.; and Mehta, R. 2021. Competitive Allocation of a Mixed Manna. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1405–1424.

Devanur, N. R.; Papadimitriou, C. H.; Saberi, A.; and Vazirani, V. V. 2008. Market Equilibrium via a Primal–Dual Algorithm for a Convex Program. *J. ACM*, 55(5).

Foley, D. 1967. Resource Allocation and the Public Sector. *Yale Economic Essays*, 7(1): 45–98.

Freeman, R.; Sikdar, S.; Vaish, R.; and Xia, L. 2019. Equitable Allocations of Indivisible Goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 280–286.

Freeman, R.; Sikdar, S.; Vaish, R.; and Xia, L. 2020. Equitable Allocations of Indivisible Chores. AAMAS '20, 384–392.

Garg, J.; and McGlaughlin, P. 2020. Computing Competitive Equilibria with Mixed Manna. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 420–428.

Garg, J.; and Murhekar, A. 2021. Computing fair and efficient allocations with few utility values. *In Proc. 14th Symp. Algorithmic Game Theory (SAGT)*.

Garg, J.; Murhekar, A.; and Qin, J. 2021. Fair and Efficient Allocations of Chores under Bivalued Preferences. *CoRR*, abs/2110.09601.

Garg, N.; Kavitha, T.; Kumar, A.; Mehlhorn, K.; and Mestre, J. 2009. Assigning Papers to Referees. *Algorithmica, v.x, x-x (2009)*, 58.

Golovin, D. 2005. Max-min fair allocation of indivisible goods. *Technical Report, CMU-CS-05-144*.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On Approximately Fair Allocations of Indivisible Goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, 125–131.

Mas-Colell, A.; Whinston, M.; and Green, J. 1995. *Microeconomic Theory*. Oxford University Press.

Murhekar, A.; and Garg, J. 2021. On Fair and Efficient Allocations of Indivisible Goods. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 5595–5602.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. 2007. *Algorithmic Game Theory*. Cambridge University Press. ISBN 9781139466547.

Orlin, J. B. 2010. Improved Algorithms for Computing Fisher's Market Clearing Prices: Computing Fisher's Market Clearing Prices. STOC '10, 291–300.

Steinhaus, H. 1949. Sur La Division Pragmatique. *Econometrica*, 17(1): 315–319.

Varian, H. R. 1974. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1): 63 – 91.

Vazirani, V. V.; and Yannakakis, M. 2021. Computational Complexity of the Hylland-Zeckhauser Scheme for One-Sided Matching Markets. In *Proceedings of the 12th Innovations in Theoretical Computer Science Conference, (ITCS)*.

Woeginger, G. J. 1997. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4): 149 – 154.