Fair Allocation of Two Types of Chores

Haris Aziz
UNSW Sydney
Sydney, Australia
haris.aziz@unsw.edu.au

Jeremy Lindsay
UNSW Sydney
Sydney, Australia
j.lindsay@student.unsw.edu.au

Angus Ritossa
UNSW Sydney
Sydney, Australia
a.ritossa@student.unsw.edu.au

Mashbat Suzuki
UNSW Sydney
Sydney, Australia
mashbat.suzuki@unsw.edu.au

ABSTRACT

We consider the problem of fair allocation of indivisible chores under additive valuations. We assume that the chores are divided into two types and under this scenario, we present several results. Our first result is a new characterization of Pareto optimal allocations in our setting and a polynomial-time algorithm to compute an envy-free up to one item (EF1) and Pareto optimal allocation. We then turn to the question of whether we can achieve a stronger fairness concept called envy-free up any item (EFX). We present a polynomial-time algorithm that returns an EFX allocation. Finally, we show that for our setting, it can be checked in polynomial time whether an envy-free allocation exists or not.

KEYWORDS

Fair allocation; chores; envy-freeness; EFX; EF1; Pareto Optimality

ACM Reference Format:
Haris Aziz, Jeremy Lindsay, Angus Ritossa, and Mashbat Suzuki. 2023. Fair Allocation of Two Types of Chores. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

How to make allocation decisions fairly is a fundamental question that has been examined in many fields including computer science, economics, operations research and mathematics. We consider this question in the context of allocating indivisible chores among agents where each agent has additive valuations over the chores.

There are several formal criteria of fairness (see e.g., [9, 21]). Among the criteria, envy-freeness is referred to as the 'gold-standard' [11]. It requires that no agent prefers another agent's bundle to their own bundle. Although envy-freeness is a highly-desirable fairness concept, it poses several challenges. An envy-free allocation may not exist, and furthermore, it is NP-complete to check whether an envy-free allocation exists under additive valuations [3, 8]. For this reason, a major focus on fair allocation is to find relaxations of envy-freeness. A particularly attractive relaxation of envy-freeness is called *envy-freeness up to any item (EFX)* [2, 11]. However, the existence of EFX is a major open problem for goods and for chores. EFX requires that if an agent is envious of another agent, ignoring any item that lessens the envy results in the envy disappearing. A weaker concept is *envy-freeness up to one item (EF1)* that requires that if an agent is envious of another agent, then there exists some item such that ignoring the item results in the envy disappearing. It is open whether an EF1 and *Pareto optimal (PO)* allocation always exists for chores.

In view of the open problem concerning the existence of EFX as well as EF1+PO allocations and the absence of positive algorithmic results regarding envy-free allocations, we turn our attention to a natural scenario of chore allocation in which there are at most two types of chores. We assume that the items can be divided into two groups A and B. Chores within the same group are identical and hence a given agent has the same value for the identical items. A natural motivating example could be a group of 4 housemates allocating monthly household chores consisting of 18 room cleaning chores and 15 cooking chores.

There are several reasons for considering the case of two chore types. Firstly, it is natural to consider restrictions on the general chore allocation under which we can achieve positive algorithmic results. For example, there are many papers that assume that agents have binary valuations for items (see, e.g., [6, 7, 13]): 0 or 1 in the case of goods and 0 and -1 in the case of chores.[1] There are also some recent papers where agents have exactly two values in the valuation functions (bi-valued utilities) [14, 15] . In contrast, we allow the set of all agents to possibly have $2n$ different values for the set of items. Finally, two chore-types is a natural subclass of *personalized bi-valued instances* (see, e.g., [14]) in which each agent subjectively divides the items into two classes and has a corresponding value for items in each of the classes.

Contributions

We give a polynomial time algorithm for computing an EF1+fPO allocation for two chore type instances (Theorem 4.7) where fPO (fractional Pareto optimal) is a property stronger than Pareto optimality and requires Pareto optimality among all fractional outcomes. Since there are very few results known on the existence of EF1+PO allocation for chores - as the general additive valuation setting is a major open problem - we make concrete progress towards the problem by providing an affirmative answer in a restricted case. En route to our result, we also give a novel characterization of all fPO allocations in our setting.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

[1] Our assumption of two chore types does not assume that agents have zero as one of the two valuations. Zero valuations make many problems considerably easier.

We prove that for two chore type instances an EFX allocation exists and can be computed in polynomial time (Theorem 5.1). Our algorithm differs significantly from the natural adaptation of the goods algorithm of Gorantla et al. [16] and other existing approaches as they fail to produce an EFX allocation in our setting. Since the existence of EFX allocations for chores is not known even in the restricted setting of three agents with additive valuations, we remark that our work contributes towards the body of literature which explores this question in restricted settings.

In the full paper [4], we show that there exists a polynomial-time algorithm to check whether an envy-free allocation exists in the two chore types setting. Note that this problem is NP-hard for general additive instances of indivisible chores [8]. Table 1 summarizes existence and complexity results under additive valuations and Figure 1 summarizes the logical relations and compatibility of the key concepts that we consider.

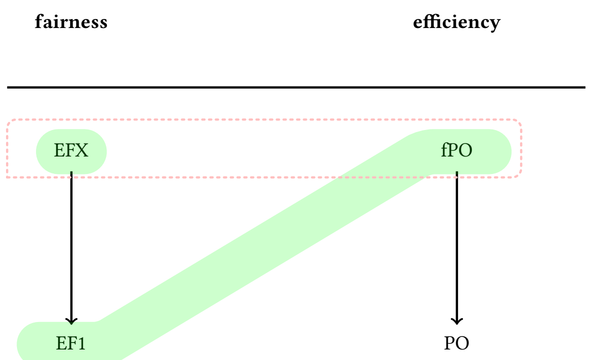

Figure 1: Logical relations between fairness and efficiency concepts. An arrow from (A) to (B) denotes that (A) implies (B). For our setting of 2 chore types, the properties in a connected solid green shape can be simultaneously satisfied, and the combined properties in connected dotted pink are impossible to simultaneously satisfy.

2 RELATED WORK

Given that an envy-free allocation may not exist, Budish [10] proposed a relaxation of envy-freeness called *envy-free up to one item (EF1)*. An allocation satisfies EF1 if it is envy-free or any agent's envy for another agent can be removed if some item is ignored. Under additive utilities, EF1 can be achieved by a simple algorithm called the round-robin sequential allocation algorithm. Agents take turn in a round-robin manner and pick their most preferred unallocated item. The interest in EF1 was especially piqued when Caragiannis et al. [11] proved that for positive additive utilities, a rule based on maximizing Nash social welfare finds an allocation that is both EF1 and Pareto optimal.

For negative additive valuations, the existence of an EF1 and PO allocation is a major open problem that Moulin [21] highlighted in his survey (page 436). Except for a limited number of cases such as binary utilities, bi-valued utilities ([14, 15]) and lexicographic valuations [17], the guaranteed existence of EF1 and PO allocations has not been established.

In their paper Caragiannis et al. [11] also presented the concept of EFX for goods which is strictly stronger than EF1. EFX requires that if an agent i is envious of another agent j, the envy can be removed by removing any item of j that is desirable to i. The concepts have been adapted for the case of chores or generalized to the case of mixed goods and chores (see e.g., [2, 5]). Procaccia [22] writes that the existence of EFX allocations is the biggest problem in fair division.

There are several papers that have explored the question concerning the existence of EFX allocations and have provided partial results. It is well-understood that EFX allocations exists for identical valuations. Chaudhury et al. [12] proved that an EFX allocation exists for the case 3 agents and goods. Mahara [19] showed that when items are goods and the agents have at most 2 types of valuation functions, then there exists an EFX allocation. Some of the results on sufficient conditions for the existence of EFX allocations have been extended to more general valuations [20]. On the other hand, Hosseini et al. [17] showed that when there are mixed goods and chores, then an EFX allocation may not exist. In this paper, we focus on EFX allocation of chores and identify conditions under which an EFX allocation exists. Zhou and Wu [23] presented algorithms that provide approximation of EFX for chores. Li et al. [18] considered PROPX which is a weaker property than EFX in the context of chores and they proposed algorithms for PROPX allocation of chores. One particular paper [16] focusses on positive valuations and among other results, presents an algorithm to compute an EFX allocation when there are at most two item types. The approach does not extend to the case of chores and our corresponding result requires a different approach and argument.

Garg et al. [15] and Ebadian et al. [14] examine problems in which agents have negative bi-valued valuations[2], and they both present a polynomial-time algorithm to compute an EF1 and Pareto optimal allocation. Ebadian et al. [14] also showed that for a subclass of personalised bi-valued allocations an MMS fair allocation can always be computed. Previously, Aziz et al. [1] characterized Pareto optimal allocations for positive bi-valued valuations.

3 PRELIMINARIES

Let M be a set of m indivisible chores, and N be a set of n agents. Each agent $i \in N$ has a valuation function $v_i : M \to \mathbb{R}_{\leq 0}$, where $v_i(r)$ indicates i's value for chore $r \in M$. Throughout the paper we assume that the valuation functions are additive, i.e., for each agent $i \in N$ and for each set of chores $S \subseteq M$, $v_i(S) = \sum_{r \in S} v_i(r)$. Our main focus is to study the following class of instances:

Definition 3.1. A fair division instance $I = (N, M, v)$ is *two chore types* if the item set can be partitioned into two sets A and B with $M = A \cup B$, such that for each $i \in N$ we have $v_i(r) = v_i(r')$ for all $r, r' \in A$, and $v_i(h) = v_i(h')$ for all $h, h' \in B$.

In plain English, an instance is two chore types if there are at most two item types such that each agent is indifferent among items of the same type. Denote v_i^A as agent i's value for an item of type A, and v_i^B as value for an item of type B. For notational convenience, we order the agents so that $\frac{v_i^A}{v_i^B} \leq \frac{v_{i+1}^A}{v_{i+1}^B}$ for all $1 \leq i < n$, where we

[2] Each agent i and item o, the valuation is either some value a or b.

	EF1 & PO	EFX
Chores: general	existence open	existence open
Chores: personalised bi-valued	existence open	existence open
Chores: bi-valued	in P, exists [14, 15]	existence open
Chores: binary	in P, exists	in P, exists
Chores: 2 item types	in P, exists (Theorem 4.7)	in P, exists (Theorem 5.1)

Table 1: Existence and complexity results under additive valuations

consider $\frac{v_i^A}{0}$ to be ∞.[3] More formally, this condition can be restated as $v_i^A v_{i+1}^B \leq v_{i+1}^A v_i^B$. Informally, this means that agents who prefer type A items have smaller indices, and agents who prefer type B items have larger indices. We divide the agents into two sets N_A and N_B, where agents in N_A prefer type A items and agents in N_B prefer type B items. In particular, if $v_i^A \geq v_i^B$ then $i \in N_A$, and otherwise $i \in N_B$. We say that an agent $i \in N_A$ *strongly prefers A* if $2v_i^A \geq v_i^B$, and define it similarly for agents in N_B.

A valuation function is called *bi-valued* if there exist $a, b \in \mathbb{R}$ such that $v_i(h) \in \{a, b\}$ for all $i \in N$ and $h \in M$. There have been several works which focus on bi-valued valuations [14, 15]. We remark that bi-valued valuations are incomparable to two chore types valuations. Two chore type instances allow the set of agents to have $2n$ different values across agents and items whereas bi-valued instances allow for exactly two. A generalization of both bi-valued and two chore type instances is called *personalized bi-valued*, where for each agent $i \in N$ there exist $a_i, b_i \in \mathbb{R}$ such that $v_i(h) \in \{a_i, b_i\}$ for all $h \in M$. For personalized bi-valued instances, the existence of EF1+PO or EFX allocations are not known.

Allocation: An *allocation* is a partition $X = (X_1, ..., X_n)$ of the item set M, where $X_i \subseteq M$ is the bundle allocated to agent $i \in N$. An allocation is called *partial* if $\bigcup_{i \in N} X_i \neq M$. We say that the allocation is *fractional* if items are allocated (possibly) fractionally such that no more than one unit of each chore is allocated. In a fractional allocation, the valuation that an agent derives from an item is directly proportional to the fraction of that item that they are allocated. Observe that for two chore type instances any bundle can be succinctly represented by the number of items of each type in the bundle. Thus we denote $X_i = (\alpha_i, \beta_i)$ where α_i is the number of type A items and β_i is the number of type B items in agent i's bundle. We write $(\alpha, \beta) \uplus (\alpha', \beta')$ to denote the set $(\alpha + \alpha', \beta + \beta')$ for convenience.

Fairness Notions: An allocation $X = (X_1, ..., X_n)$ is *envy-free (EF)* if for any agents $i, j \in N$, we have $v_i(X_i) \geq v_i(X_j)$. It is easy to see that EF allocations may not exist in general[4]. As a result weaker fairness notions EF1 and EFX have been introduced. An allocation X is *envy-free up to one chore* (EF1) if for any agents $i, j \in N$, where $X_i \neq \emptyset$, there exists a chore $h \in X_i$ such that $v_i(X_i \setminus h) \geq v_i(X_j)$. An allocation X is *envy-free up to any chore* (EFX) if for any agents $i, j \in N$, and for any chore $h \in X_i$ with $v_i(h) < 0$, we have $v_i(X_i \setminus h) \geq v_i(X_j)$.

Observe that EFX implies EF1, but not vice versa. We say that an agent i *EF1-envies* (respectively *EFX-envies*) another agent j if i envies j and this envy is not EF1 (respectively EFX).

Efficiency Notions: An allocation Y *Pareto dominates* another allocation X if $v_i(Y_i) \geq v_i(X_i)$ for all agents i and there exists an agent j such that $v_j(Y_j) > v_j(X_j)$. An allocation is *Pareto optimal* (PO) if it is not Pareto dominated by any allocation. An allocation is *fractionally Pareto optimal* (fPO) if it is not Pareto dominated by any fractional allocation. Note that an fPO allocation is also PO, but a PO allocation is not necessarily fPO.

For the remainder of the paper, we assume that all agents have strictly negative valuations for both item types. We make this assumption since if there is at least one agent who values a chore at zero then both EF1+fPO and EFX allocations can be found in a straightforward way. To see this, observe that if there is an agent i with $v_i^A = 0$ and an agent j with $v_j^B = 0$, then we can give all type A items to agent i and all type B items to agent j. In this case, every agent values their bundle at 0 and so this is trivially EF1+fPO and also EFX. On the other hand, without loss of generality, if there exists an agent i with $v_i^A = 0$, but $v_j^B < 0$ for all agents j then we assign all type A items to agent i and we assign the type B items in a round-robin way to all the agents. This gives an EFX allocation because each agent has at most one more type B item than any other agent. Additionally, this allocation is fPO since all type A items were allocated to an agent who values them at zero, and so redistributing these items cannot lead to a Pareto improvement. Furthermore, if any agent were to receive fewer type B items (possibly fractionally), a different agent must receive more type B items, and hence no Pareto improvements are possible.

4 EF1+ fPO

In this section, we present a polynomial-time algorithm that computes an EF1 and fPO allocation for the fair division problem with two chore type instances. En route, we give a novel characterization of fPO allocations in our setting.

Characterization of fPO Allocations

We begin by providing a new characterization of structure of fPO allocations by showing Lemma 4.1. Due to space constraints we defer the proof of Lemma 4.1 to the full paper [4].

[3]We assume that no agent values both item types at 0, as otherwise we can simply allocate all the chores to that agent.

[4]Consider an instance where there is one chore and two agents who have negative values for the chore

Lemma 4.1. *Given a two chore types instance where all agents have strictly negative valuations, an allocation $X = (X_1, ..., X_n)$ is fPO if and only if there exists an agent i such that:*

- *For all agents j where $\frac{v_j^A}{v_j^B} < \frac{v_i^A}{v_i^B}$, the bundle X_j only contains type A items.*
- *For all agents j where $\frac{v_j^A}{v_j^B} > \frac{v_i^A}{v_i^B}$, the bundle X_j only contains type B items.*

We remark that Lemma 4.1 allows us to restrict our attention to allocations that obey the structure outlined in the lemma. In Figure 2, we give a visualisation of this structure.

$$\underbrace{\frac{v_1^A}{v_1^B} \leq \dots \leq \frac{v_{i-1}^A}{v_{i-1}^B}}_{\text{Only type } A} < \underbrace{\frac{v_i^A}{v_i^B} = \dots = \frac{v_j^A}{v_j^B}}_{\text{No restrictions}} < \underbrace{\frac{v_{j+1}^A}{v_{j+1}^B} \leq \dots \leq \frac{v_n^A}{v_n^B}}_{\text{Only type } B}$$

Figure 2: The general form of allocations that satisfy Lemma 4.1.

Algorithm for EF1+fPO

To find an EF1 and fPO allocation, it is sufficient to consider only a subset of the allocations that satisfy Lemma 4.1. In particular, we consider a set of allocations with the following structure.

Definition 4.2. An allocation $X = (X_1, ..., X_n)$ is *ordered with respect to agent i* (or *ordered* for short) if there exists some agent i where:

- For all agents j where $j < i$, the bundle X_j only contains type A items.
- For all agents j where $j > i$, the bundle X_j only contains type B items.

We remark that all ordered allocations satisfy Lemma 4.1, but the converse does not necessarily hold (in particular, it does not always hold when there are multiple agents with identical preferences).

First, we consider an even more restricted class of allocations, namely *split-round-robin*.

Definition 4.3. Let i be an agent such that $1 \leq i < n$. The allocation *split-round-robin(i)* is the allocation formed by distributing the type A items to agents 1 through i in a round-robin way, and distributing the type B items to agents $i+1$ through n in a round-robin way. In both cases, we allocate to agents with smaller indices first.

By Lemma 4.1, the allocation *split-round-robin(i)* is fPO for all i. We introduce terminology to describe whether a *split-round-robin* allocation is EF1. Let i be an agent such that $1 \leq i < n$. We say that the allocation *split-round-robin(i)* has *A-envy* if there is an agent $j \leq i$ who has EF1-envy towards another agent $k > i$. Similarly, we say that the allocation *split-round-robin(i)* has *B-envy* if there is an agent $j > i$ who has EF1-envy towards another agent $k \leq i$.

Observe that *split-round-robin(i)* is EF1 if and only if it does not have A-envy nor B-envy. We can now begin describing our algorithm for finding an EF1 and fPO allocation. Algorithm 1 begins by checking whether *split-round-robin(i)* is EF1 for any $1 \leq i < n$. If so, then the algorithm has found an EF1 and fPO allocation. Otherwise, we create an allocation which is ordered with respect to a carefully chosen agent, who we call a *split-agent.*

Definition 4.4. An agent i is a *split-agent* if both of the following conditions hold:

- Either $i = 1$ or *split-round-robin(i-1)* has A-envy, and
- Either $i = n$ or *split-round-robin(i)* has B-envy.

Lemma 4.5. *If split-round-robin(i) is not EF1 for all $1 \leq i < n$, then there exists a split-agent.*

Proof. Observe that if *split-round-robin(i)* is not EF1 (for any $1 \leq i < n$), it must have A-envy or B-envy. If neither 1 nor n are split-agents, then *split-round-robin(1)* has A-envy and *split-round-robin(n-1)* has B-envy. Hence, there must exist some $1 < i < n$ such that *split-round-robin(i-1)* has A-envy and *split-round-robin(i)* has B-envy. □

We select a split-agent i^*, and will create an instance that is ordered with respect to i^*. We now explore a useful property of ordered allocations.

Lemma 4.6. *Let $I = (N, M, v)$ be a two chore types instance and X be an allocation that is ordered with respect to agent i^*. Consider a modified valuation profile $\tilde{v}$, where $\tilde{v}_j = v_{i^*}$ for all $j \in N$. If X is EF1 with respect to the modified valuation profile $\tilde{v}$ then it is EF1 in the original valuation profile v.*

Proof. As X is ordered with respect to agent i^*, any agent $j < i^*$ has only type A items i.e., $X_j = (\alpha_j, 0)$. Consider now some other agent $k \in N$. We show that if agent j does not EF1-envy k under a modified valuation $\tilde{v}_j = v_{i^*}$, then j does not EF1-envy k in the original instance.

Observe that if $\alpha_j = 0$, then agent j is not allocated any chores, and thus she does not have envy towards any other agent. Hence we assume that $\alpha_j > 0$. Since X is EF1 under the modified valuation profile, agent j does not EF1-envy k when $\tilde{v}_j = v_{i^*}$. It follows that,

$$\begin{aligned} \tilde{v}_j(\alpha_j - 1, 0) &= (\alpha_j - 1)v_{i^*}^A \\ &\geq \alpha_k v_{i^*}^A + \beta_k v_{i^*}^B \end{aligned} \tag{1}$$

Recalling $j < i^*$, we have $\frac{v_j^A}{v_j^B} \leq \frac{v_{i^*}^A}{v_{i^*}^B}$. Rearranging we have that $\frac{v_j^A}{v_{i^*}^A} \leq \frac{v_j^B}{v_{i^*}^B}$. As both sides of Equation (1) are non-positive, it follows that $(\alpha_j - 1)v_j^A \geq \alpha_k v_j^A + \beta_k v_j^B$, and hence j does not EF1-envy k under the original valuation function.

We can apply a similar argument for agents $j > i^*$. □

Theorem 4.7. *Given a two chore types instance, Algorithm 1 finds an allocation that is EF1 and fPO in polynomial-time.*

Proof. First observe that the algorithm only outputs an ordered allocation and thus fPO by Lemma 4.1. Furthermore if *split-round-robin(i)* is EF1 for some i then the algorithm returns an allocation that is both EF1 and fPO immediately. Thus the main challenge is to analyse the algorithm on instances where *split-round-robin(i)* is not EF1 for any $1 \leq i < n$. In the remainder of the proof we restrict our attention to these instances.

Algorithm 1: Computing an EF1 and fPO allocation

Input : A fair allocation instance with two chore types, where all agents have strictly negative valuations
Output : An allocation which is EF1 and fPO

1 **for** $i \leftarrow 1$ **to** $n-1$ **do**
2   **if** *split-round-robin(i) is EF1* **then**
3   **return** *split-round-robin(i)*
4 $i^* \leftarrow$ a *split agent* ; ▷ Note that such an agent is guaranteed to exist by Lemma 4.5.
5 $X = (X_1, ..., X_n) \leftarrow$ an allocation where $X_{i^*} = M$ and $X_j = \emptyset$ for all $j \neq i^*$.
6 **while** *X is not EF1* **do**
7     $j \leftarrow$ an agent in $\arg\max_{j \in N \setminus i^*} v_{i^*}(X_j)$
8   **if** $j < i^*$ **then**
9   Transfer a type A item from X_{i^*} to X_j
10   **else**
11   Transfer a type B item from X_{i^*} to X_j
12 **return** X

Recall that by Lemma 4.5 there exists a split agent i^*. At a high level the algorithm transfers items from the split agent to other agents until the allocation becomes EF1 whilst maintaining that the allocation is ordered with respect to i^*.

Consider now a modified valuation $\tilde{v}_j = v_{i^*}$ for all $j \in N$. We show that the algorithm outputs an EF1 allocation with respect to the modified instance. By Lemma 4.6, the same allocation is also EF1 with respect to the original instance. In the modified instance, there is no EF1-envy among all agents other than i^* since their bundles are formed by repeatedly transferring an item to the agent with the highest valuation. In particular, if X is not EF1, this must be due to EF1-envy that agent i^* has for another agent, or EF1-envy that another agent has towards agent i^*.

Let X^L be the earliest allocation X encountered in Algorithm 1 where $v_{i^*}(X_{i^*}) \neq \min_{j \in N} v_{i^*}(X_j)$ holds (assuming that Algorithm 1 does not terminate prior to this). If Algorithm 1 terminates prior to X^L, for simplicity we say that every allocation in the algorithm is prior to X^L. We show that for all allocations X prior to (and including) X^L, no agent has EF1-envy towards i^*. The statement holds for all allocations prior to X^L from the definition of X^L. We now prove that in the allocation X^L, no agent has EF1-envy towards i^*. Let X' be the allocation immediately prior to X^L. Note that X' is not EF1, or otherwise Algorithm 1 would have terminated. By definition of X', we have $v_{i^*}(X'_{i^*}) = \min_{j \in N} v_{i^*}(X'_j)$. Since X' is not EF1 it follows that agent i^* must EF1-envy agent j, where $j \in \arg\max_{j \in N \setminus \{i^*\}} v_{i^*}(X'_j)$. Note that j is the agent who was transferred an item in Algorithm 1 when the allocation X^L was created. Since the bundle $X^L_{i^*}$ has one less item than X'_{i^*} and agent i^* had EF1-envy towards j when the allocation was X', it follows that $v_{i^*}(X^L_{i^*}) < v_{i^*}(X'_j)$. For all agents $k \neq j, i^*$, their bundle is unchanged between X'_k and X^L_k. Because agent k does not EF1-envy the bundle X'_j they do not EF1-envy the even worse bundle $X^L_{i^*}$. Therefore in the allocation X_L, no agent has EF1-envy towards i^*.

Claim 1: For all allocations X prior to and including X^L, we have that X_{i^*} contains at least one item of each type.

Proof of Claim 1. Recall that no agent has EF1 envy towards i^* and thus every agent other than i^* has no EF1-envy towards any agent.

We first prove that X_{i^*} has at least one type A item. If $i^* = 1$, then this is immediately true. Otherwise, assume $i^* > 1$. We proceed by contradiction. Assume that X_{i^*} has no type A items. Then, agents 1 through $i^* - 1$ have all the type A items, and agents i through n have all the type B items, just as in the allocation *split-round-robin*$(i^* - 1)$. However, because i^* is a split-agent, we know that *split-round-robin*$(i^* - 1)$ has A-envy. Thus there must exist some agent $j < i^*$ who has EF1-envy towards another agent $k \geq i^*$ in X which is a contradiction.

We now prove that there is at least one type B item. If $i^* = n$, it follows immediately. Otherwise, if $i^* < n$, we can use a symmetrical argument to the type A item case. □

In the next paragraph, we will show that the algorithm terminates (i.e. returns an EF1 allocation) prior to or at allocation X^L. Therefore, by Claim 1, whenever Line 9 is reached, X_{i^*} has at least one type A item, and whenever Line 11 is reached, X_{i^*} has at least one type B item.

If the algorithm terminates prior to X^L then we are done. Otherwise, if every allocation prior to X^L is not EF1 then we show that X^L must be EF1. By the definition of X^L, there exists some agent k such that $v_{i^*}(X^L_k) < v_{i^*}(X^L_{i^*})$. Since no agent in $N \setminus i^*$ has any EF1-envy towards any other agent in N, it follows that k does not EF1-envy any agent i.e., there exists some chore $r \in X^L_k$ such that $v_{i^*}(X^L_k \setminus r) \geq v_{i^*}(X^L_l)$ for all agents l. By Claim 1, $X^L_{i^*}$ contains at least one item of each type, and therefore contains an item r' of the same type as r. Therefore $v_{i^*}(X^L_{i^*} \setminus r') > v_{i^*}(X^L_k \setminus r) \geq v_{i^*}(X^L_l)$ for all $l \in N$. Hence X^L is EF1 with respect to the modified instance.

As for time complexity, the algorithm runs in polynomial-time since the while loop on Line 6 can only run at most m times. □

EFX and fPO are not always compatible

A natural extension of Theorem 4.7 is to ask whether an allocation always exists that is EFX and fPO. Here, we disprove this by providing an instance with no allocation that is both EFX and fPO.

Consider an instance with 3 agents, where $v_1^A = -10$, $v_2^A = -11$, $v_3^A = -12$, and $v_1^B = v_2^B = v_3^B = -1$. There are 3 type A items and 2 type B items. For the allocation to be EFX, each agent must receive one type A item. Otherwise, one agent would receive at least 2 type A items and another agent would receive no type A items, which cannot be EFX. However, if the allocation is fPO it must satisfy Lemma 4.1 and so agent 3 must receive both type B items. However, this is not EFX. Hence, in this instance, there does not exist any allocation that is both EFX and fPO.

Due to this nonexistence result, we instead consider the question of whether an EFX allocation always exists.

5 EFX

In this section, we give an algorithm to compute an EFX allocation of chores when there are two item types. Our first observation is that important algorithms for chore allocation as well natural adaptations for fair allocation of goods to the case of chores do not give EFX guarantees even for two item types. These include two algorithms ("The Top-trading Envy Cycle Elimination Algorithm" and "The Bid-and-Take Algorithm") for PROPX allocations by Li et al. [18] as well as an adaptation the algorithm of Gorantla et al. [16] to the case of chores. This is detailed in the full paper [4].

The main result of this section is Theorem 5.1, which we use the remainder of this section to prove.

Theorem 5.1. *For two chore type instances, an EFX allocation always exists and can be found in polynomial-time.*

5.1 Allocation algorithm when $|A| \leq |N_A|$ or $|B| \leq |N_B|$

The main algorithm in Section 5.2 requires $|A| > |N_A|$ and $|B| > |N_B|$, and so we begin with an algorithm for when this does not hold. This case is detailed in the full paper [4].

5.2 Allocation algorithm when $|A| > |N_A|$ and $|B| > |N_B|$

In this section, we prove that Algorithm 2 always finds an EFX allocation in polynomial time. We assume without loss of generality that $|N_A| \geq |N_B|$.

We begin with an overview of Algorithm 2. Algorithm 2 starts by computing an EFX partial allocation X^* on Line 3. In this initial allocation, all type B (and potentially some type A) items are allocated. Algorithm 2 then applies one of following two update rules until all type A items are allocated:

- Rule 1 (Line 8). Let a be the number of unallocated type A items and let $X' = (X'_1, ..., X'_n)$ be an allocation where $X'_i = X_i$ for all $i \in N_A$ and $X'_j = X_j \uplus (1, 0)$ for all $j \in N_B$. If $a \geq |N_B|$ and X' is EFX, then set X to be X'. We refer to the condition "X' is EFX" as the "EFX condition of Rule 1".
- Rule 2 (Line 11). If Rule 1 does not apply, then let $i \in N_A$ be an agent who is envy-free (we will prove that such an agent always exists under our choice of X^*). We allocate a type A item to i.

Note that both rules preserve EFX. In particular, Rule 1 preserves EFX by definition, and Rule 2 preserves EFX because any envy that agent i has will disappear if a single type A item is removed from their bundle. Hence, if Algorithm 2 returns, then the returned allocation will be EFX. Additionally, Algorithm 2 runs in polynomial time because the update rules will be applied at most m times.

However, it is not guaranteed that the updates rules can always be applied for every choice of X^*: Example 5.2 demonstrates a case where neither rule can be applied. Therefore, the initial allocation X^* must be chosen carefully so that a situation similar to Example 5.2 never occurs. In particular, for the chosen initial allocation X^* we must show that whenever Line 10 is reached, there always exists an agent $i \in N_A$ where $v_i(X_i) \geq v_i(X_j)$ for all $j \in N$. We introduce some terminology to reason about this: if there exists such an agent i, we say that "Rule 2 can be applied". If it is possible to apply Rule 2 k times consecutively, then we say that "Rule 2 can be applied k times". Note that we use these terms regardless of whether Rule 1 can be applied.

Agents	v_i^A	v_i^B	X_i
1	-1	-5	$(1, 1)$
2	-5	-1	$(0, 1)$
3	-5	-1	$(0, 2)$

Example 5.2. An instance with an EFX allocation X. If there is $a = 1$ unallocated type A item, then neither update rule can be applied. In particular, Rule 1 cannot be applied because there are insufficient unallocated items. Rule 2 cannot be applied because agent 1 would EFX-envy agent 2 if the rule were to be applied.

Algorithm 2: Computing an EFX allocation

Input : A fair allocation instance with two chore types, where all agents have strictly negative valuations and $|N_A| \geq |N_B|$

Output : An EFX allocation

1 **if** $|A| \leq |N_A|$ *or* $|B| \leq |N_B|$ **then**
2   **return** the allocation described in Section 5.1
3 $X = (X_1, ..., X_n) \leftarrow X^*$, an initial partial EFX allocation, described in Section 5.3
4 **while** X *is a partial allocation* **do**
5     $a \leftarrow$ the number of unallocated type A items
6     $X' = (X'_1, ..., X'_n) \leftarrow$ an allocation where $X'_i = X_i$ for all $i \in N_A$ and $X'_j = X_j \uplus (1, 0)$ for all $j \in N_B$
7   **if** $a \geq |N_B|$ *and* X' *is EFX* **then**
8         $X \leftarrow X'$ ▷ Rule 1
9   **else**
10         $i \leftarrow$ an agent in N_A where $v_i(X_i) \geq v_i(X_j)$ for all $j \in N$
11         $X_i \leftarrow X_i \uplus (1, 0)$ ▷ Rule 2
12 **return** X

The remainder of this section is structured as follows: We begin by introducing some results in Lemma 5.3-5.6 that are helpful later in the section. We then provide conditions for X^* under which Algorithm 2 always finishes and returns an allocation. In particular, both Lemma 5.7 and Lemma 5.9 give sufficient conditions for X^*. Finally, in Section 5.3, we show how to compute the initial allocation X^*. To do this, we must consider several cases that together cover every possible input instance for Algorithm 2. In every case, we show that we can find an initial allocation X^* that satisfies the criteria of Lemma 5.7 or Lemma 5.9.

Due to space constraints, the proofs of Lemma 5.3-5.9 are deferred to the full paper [4].

Lemma 5.3. *Let i and j be two agents, and let X be an allocation. If $i > j$ and X_i has at least as many type B items as X_j, then i and j cannot both envy each other. That is, if i envies j, then j does not envy i.*

Lemma 5.4. *Let $i \in N_A$ and $j \in N_B$ be two agents, and let X_i and X_j be their bundles. If X_j has strictly more type B items than X_i and j EFX-envies i, then $|X_i| < |X_j| - 1$.*

Lemma 5.5. *Let X be an EFX allocation where all agents $j \in N_B$ have strictly more type B items than all agents $i \in N_A$. If there exists an agent $i \in N_A$ such that $|X_i| < |X_j|$ for all $j \in N_B$, then for all $i' \in N_A$ and $j' \in N_B$ it holds that agent i' does not envy agent j'.*

Lemma 5.6. *Let X be an EFX partial allocation where all type B items are allocated. Assume that Algorithm 2 applies Rule 2 to X to create a new allocation X', and then applies Rule 1 to X' to create X''. Then, the EFX condition of Rule 1 does not hold for X''.*

We are now ready to state our first set of sufficient conditions for the initial allocation X^*.

Lemma 5.7. *Let X^* be an EFX partial allocation where all type B items are allocated. If X^* satisfies the following conditions, then the update rules can be applied until all items are allocated:*

(1) The EFX condition of Rule 1 does not hold for X^*, and
(2) Consider a partial allocation Y formed by applying the update rules 0 or more times to X^*. Whenever the EFX condition of Rule 1 does not hold for Y, Rule 2 can be applied $|N_B|$ times to Y.

The proof of Lemma 5.7 uses Lemma 5.6 and is in the full paper [4].

We now present Lemma 5.8, that gives a set of conditions under which the second condition of Lemma 5.7 is satisfied.

Lemma 5.8. *Let X be an EFX partial allocation. If X satisfies the following conditions, then Rule 2 can be applied $|N_B|$ times:*

(1) For all agents $i \in N_A$ and $j \in N_B$, X_j has strictly more type B items than X_i,
(2) For all agents $i \in N_A$ and $j \in N_B$, i does not envy j, and
(3) Consider a partial allocation Y formed by applying the update rules 0 or more times to X. For any such allocation Y and any nonempty subset $S \subseteq N_A$, there exists some agent $i \in S$ who does not envy any other agent in S.

Finally, we provide a result which gives an alternate set of conditions for the initial allocation X^*.

Lemma 5.9. *Let X^* be an EFX partial allocation where all type B items are allocated. If X^* satisfies the following conditions, then the update rules can be applied until all items are allocated:*

(1) The EFX condition of Rule 1 does not hold for X^*,
(2) $|X_i^*| = |X_{i'}^*|$ for all $i, i' \in N_B$,
(3) For all agents $i \in N_A$ and $j \in N_B$, X_j^* has strictly more type B items than X_i^*, and
(4) Consider a partial allocation Y formed by applying the update rules 0 or more times to X^*. For any such allocation Y and any nonempty subset $S \subseteq N_A$, there exists some agent $i \in S$ who does not envy any other agent in S.

The proof of Lemma 5.9 uses Lemma 5.5, Lemma 5.7 and Lemma 5.8 and is in the full paper [4].

5.3 Computing X^*

In this section, we describe how to compute X^* and justify how this initial allocation is sufficient for Algorithm 2 to output an EFX allocation. We consider several cases, depending on the input instance.

Let a and b be the number of unallocated type A and B items respectively. Initially, $a = |A|$ and $b = |B|$.

Let $k = \left\lfloor \frac{b-|N_B|}{n} \right\rfloor$. We begin by assigning k type B items to all agents in N_A and $k+1$ to all agents in N_B. In particular,

$$X_i^* = \begin{cases} (0,k) & \text{for } i \in N_A, \\ (0,k+1) & \text{for } i \in N_B. \end{cases}$$

Now, $0 \le b < n$. We consider two cases, depending on b.

5.3.1 *Case 1: $b \ge |N_B|$.*

Let $N'_A = \{ i \in N_A : i > n-b \}$. Note that $|N'_A| = b - |N_B|$. We allocate one more type B item to all agents in $N_B \cup N'_A$, so that $b = 0$. We also allocate one type A item to all agents in $N_A \backslash N'_A$. In particular, the partial allocation is:

$$X_i^* = \begin{cases} (1,k) & \text{for } i \in N_A \backslash N'_A, \\ (0,k+1) & \text{for } i \in N'_A, \\ (0,k+2) & \text{for } i \in N_B. \end{cases}$$

We use Lemma 5.9 to show that the update rules can be applied until all items are allocated. First, note that the partial allocation is EFX and the first three conditions of Lemma 5.9 clearly hold. For the fourth condition, consider a partial allocation Y as described in Lemma 5.9, and some nonempty subset $S \subseteq N_A$. If $S \subseteq N'_A$ or $S \subseteq N_A \setminus N'_A$, then the fourth condition holds as any agent $i \in \arg\min_{j \in S} |Y_j|$ does not envy any other agents in S. Otherwise, let i be an agent in $\arg\min_{j \in S \cap N'_A} |Y_j|$ and i' be an agent in $\arg\min_{j \in S \cap (N_A \backslash N'_A)} |Y_j|$. By Lemma 5.3 these agents cannot both envy each other, and so assume without loss of generality that i does not envy i'. Then, i does not envy any agents in S. Hence this allocation satisfies all the conditions of Lemma 5.9.

5.3.2 *Case 2: $b < |N_B|$.*

Let $N'_B = \{ i \in N_B : i > n-b \}$. Note that $|N'_B| = b$. We assign one more type B item to all agents in N'_B, so that $b = 0$. This gives us the following partial allocation that is not EFX:

$$X_i^* = \begin{cases} (0,k) & \text{for } i \in N_A, \\ (0,k+1) & \text{for } i \in N_B \backslash N'_B, \\ (0,k+2) & \text{for } i \in N'_B. \end{cases}$$

If $|A| \le 2|N_A|$, then we allocate the type A items to agents in N_A in a round-robin way. Note that each agent in N_A will receive 1 or 2 type A items (since $|N_A| < |A| \le 2|N_A|$). In particular, let N'_A be the agents who receive 1 type A item. Then we will have the following EFX allocation:

$$X_i^* = \begin{cases} (1,k) & \text{for } i \in N'_A, \\ (2,k) & \text{for } i \in N_A \backslash N'_A, \\ (0,k+1) & \text{for } i \in N_B \backslash N'_B, \\ (0,k+2) & \text{for } i \in N'_B. \end{cases}$$

Since there are no unallocated items, this case is complete. Otherwise, we know that $|A| > 2|N_A|$. We consider three final subcases.

Case 2.1: For all $j \in N_B \backslash N'_B$, agent j does not strongly prefer B (recall that j strongly prefers B if $2v_j^B \ge v_j^A$). In this case, we allocate one type A item to all agents in $N_A \cup N_B \backslash N'_B$, resulting in the following EFX partial allocation:

$$X_i^* = \begin{cases} (1,k) & \text{for } i \in N_A, \\ (1,k+1) & \text{for } i \in N_B \backslash N_B', \\ (0,k+2) & \text{for } i \in N_B'. \end{cases}$$

This partial allocation is EFX because agents in $N_B \backslash N_B'$ prefer 1 type A item over 2 type B items. We use Lemma 5.9 to show that the update rules can be applied until all items are allocated. The first three conditions of Lemma 5.9 clearly hold. For the fourth condition, consider a partial allocation Y as described in Lemma 5.9 and a nonempty subset $S \subseteq N_A$. Then, any agent $i \in \arg\min_{j \in S} |Y_j|$ does not envy any other agents in S. Hence this allocation satisfies all the conditions of Lemma 5.9.

Case 2.2: There are at least $|N_B|$ agents $i \in N_A$ who strongly prefer A. In this case, we give one type A item to all agents in N_A, resulting in the following EFX partial allocation:

$$X_i^* = \begin{cases} (1,k) & \text{for } i \in N_A, \\ (0,k+1) & \text{for } i \in N_B \backslash N_B', \\ (0,k+2) & \text{for } i \in N_B'. \end{cases}$$

We use Lemma 5.7 to show that the update rules can always be applied. The first condition clearly holds. For the second condition, consider a partial allocation Y as described in Lemma 5.7, and assume that the EFX condition of Rule 1 does not hold for Y. Then, there must exist some agent $j \in N_B$ who would EFX-envy some agent $i \in N_A$ if Rule 1 were to be applied. Then, by Lemma 5.4, we know that $|Y_i| < |Y_j|$. However, observe that $|Y_j| \leq |Y_{j'}| + 1$ for all $j' \in N_B$ and so $|Y_i| < |Y_j| \leq |Y_{j'}|+1$ implying that $|Y_i| \leq |Y_{j'}|$ for all $j' \in N_B$. Additionally, since all agents in N_A have the same number of type B items and Y is EFX, it follows that $|Y_{i'}| \leq |Y_i|+1 \leq |Y_{j'}|+1$ for all $i' \in N_A$ and $j' \in N_B$.

We can therefore apply Rule 2 at least $|N_B|$ times to Y as follows:

- While there exists an agent $i \in N_A$ where $|Y_i| \leq |Y_j|$ for all $j \in N_B$, apply Rule 2 to such an agent with the smallest $|Y_i|$. This maintains EFX as i did not envy any agent prior to the rule being applied.
- After doing the above step one or more times, all agents $i \in N_A$ have identical bundles (with $|Y_i| \leq |Y_j| + 1$ for all $j \in N_B$). We can apply Rule 2 once to all agents who strongly prefer A. This maintains EFX as these agents will not EFX-envy any $j \in N_B$ because they prefer two type A items over a type B item.

Case 2.3: Cases 2.1 and 2.2 do not hold. Since Case 2.2 does not hold, there are less than $|N_B|$ agents $i \in N_A$ who strongly prefer A. Since Case 2.1 does not hold, there exists some agent $j \in N_B \backslash N_B'$ who strongly prefers B and so all agents $j' \in N_B'$ must strongly prefer B.

Let $N_A' = \{ i \in N_A : i \leq |N_B| \}$. Note that $|N_A'| = |N_B|$. We transfer one type B item from each agent in N_A' to the agents in N_B, allocate 2 type A items to all agents in N_A' and allocate 1 type A item to all agents in $N_A \backslash N_A'$. In particular,

$$X_i^* = \begin{cases} (2,k-1) & \text{for } i \in N_A', \\ (1,k) & \text{for } i \in N_A \backslash N_A', \\ (0,k+2) & \text{for } i \in N_B \backslash N_B', \\ (0,k+3) & \text{for } i \in N_B'. \end{cases}$$

Since there are less than $|N_B|$ agents $i \in N_A$ who strongly prefer A, all these agents must be in N_A' and so no agent in $N_A \backslash N_A'$ strongly prefers A. Thus, there is no EFX-envy from any agent $i \in N_A$ towards any other agent in N. Additionally, since all agents in N_B' strongly prefer B there is no EFX-envy from any agent $j \in N_B$ towards any other agent in N. Therefore, X^* is EFX.

We use Lemma 5.7 to show that this initial allocation is sufficient for Algorithm 2. The first condition of Lemma 5.7 clearly holds. For the second condition, consider a partial allocation Y as described in Lemma 5.7, and assume that the EFX condition of Rule 1 does not hold for Y. We use Lemma 5.8 to show that the second condition of Lemma 5.7 holds.

(1) The first condition of Lemma 5.8 holds for Y because it holds for X^*.
(2) For the second condition of Lemma 5.8, note that the EFX condition of Y does not hold by the definition of Y. Hence, there exists some $i \in N_A$ and $j \in N_B$ such that j would EFX-envy i if Rule 1 was applied. By Lemma 5.4, $|Y_i| < |Y_j|$. If $j \in N_B \backslash N_B'$, then $|Y_i| < |Y_{j'}|$ for all $j' \in N_B$. Therefore, by Lemma 5.5 we know that for all agents $i' \in N_A$ and $j' \in N_B$, agent i' does not envy agent j'.
If $j \in N_B'$, then we show that $|Y_i| < |Y_j| - 1$, by proving that $|Y_i| \neq |Y_j| - 1$. Let $Y_j = (\alpha_j, k+3)$ and assume $|Y_i| = |Y_j| - 1$. Then,
$$Y_i = \begin{cases} (\alpha_j+3,k-1) & \text{if } i \in N_A', \\ (\alpha_j+2,k) & \text{if } i \in N_A \backslash N_A'. \end{cases}$$
If Rule 1 was applied, Y_j would be $(\alpha_j+1, k+3)$. However, if this occurred, j would not EFX-envy i in either case (since j strongly prefers B) and so $|Y_i| < |Y_j| - 1$. This implies that $|Y_i| < |Y_{j'}|$ for all $j' \in N_B$ and so we can apply Lemma 5.5.
(3) For the third condition of Lemma 5.8, we can use the same argument that is used in Section 5.3.1.

This completes our proof of Theorem 5.1.

6 DISCUSSION

The existence of EF1 and PO allocations or EFX allocations for the case of chores are major open problems in fair division. In this paper, we identified a natural setting or valuation restriction under which not only can we guarantee the existence of allocations that satisfy EF1 and PO, and EFX respectively, but such allocations can be computed in polynomial time. A related question is the complexity of checking whether there exists an envy-free allocation. Whereas this problem is NP-complete for chores in general, there exists a dynamic program for two chore types instances that can solve the problem in polynomial time (details are in the full paper [4]). There are several relevant problems that remain open. The existence and complexity of EF1 and PO allocations or EFX allocations is open for personalized bi-valued utilities. It is also open whether there always exists a PO and EFX allocation for our setting.

ACKNOWLEDGMENTS

Aziz is supported by the Defence Science and Technology Group through the Centre for Advanced Defence Research in Robotics and Autonomous Systems under the project "Task Allocation for Multi-Vehicle Coordination" (UA227119). Mashbat Suzuki is supported by the ARC Laureate Project FL200100204 on "Trustworthy AI".

REFERENCES

[1] H. Aziz, P. Biro, J. Lang, J. Lesca, and J. Monnot. 2019. Efficient Reallocation under Additive and Ordinal Preferences. *Theoretical Computer Science* (2019).

[2] H. Aziz, I. Caragiannis, A. Igarashi, and T. Walsh. 2022. Fair allocation of indivisible goods and chores. *Journal of Autonomous Agents and Multi-Agent Systems* (2022).

[3] H. Aziz, S. Gaspers, S. Mackenzie, and T. Walsh. 2014. Fair Assignment Of Indivisible Objects Under Ordinal Preferences. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1305–1312.

[4] H. Aziz, J. Lindsay, A. Ritossa, and M. Suzuki. 2022. Fair Allocation of Two Types of Chores. https://arxiv.org/abs/2211.00879

[5] H. Aziz, H. Moulin, and F. Sandomirskiy. 2020. A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation. *Operations Research Letters* 48, 5 (2020), 573–578.

[6] H. Aziz and S. Rey. 2020. Almost Group Envy-free Allocation of Indivisible Goods and Chores. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*. 39–45.

[7] S. Barman, S. K. Krishnamurthy, and R. Vaish. 2018. Greedy algorithms for maximizing Nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[8] Umang Bhaskar, AR Sricharan, and Rohit Vaish. 2021. On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[9] S. Bouveret, Y. Chevaleyre, and J. Lang. 2016. Fair Allocation of Indivisible Goods. In *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia (Eds.). Cambridge University Press, Chapter 12, 284–311.

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

[11] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation (TEAC)* 7, 3 (2019).

[12] B. R. Chaudhury, J. Garg, and K. Mehlhorn. 2020. EFX Exists for Three Agents. (2020). Working paper.

[13] A. Darmann and J. Schauer. 2015. Maximizing Nash product social welfare in allocating indivisible goods. *European Journal of Operational Research* 247, 2 (2015), 548–559.

[14] S. Ebadian, D. Peters, and N. Shah. 2022. How to Fairly Allocate Easy and Difficult Chores. In *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*. 372–380.

[15] J. Garg, A. Murhekar, and J. Qin. 2022. Fair and Efficient Allocations of Chores under Bivalued Preferences. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 5043–5050.

[16] Pranay Gorantla, Kunal Marwaha, and Santhoshini Velusamy. 2022. Fair allocation of a multiset of indivisible items. *CoRR* abs/2202.05186 (2022).

[17] H. Hosseini, S. Sikdar, R. Vaish, and L. Xia. 2022. Fairly Dividing Mixtures of Goods and Chores under Lexicographic Preferences. *CoRR* abs/2203.07279 (2022). arXiv:2203.07279

[18] B. Li, Y. Li, and X. Wu. 2021. *Almost Proportional Allocations for Indivisible Chores*. Technical Report arXiv:2103.11849. arXiv.org.

[19] R. Mahara. 2020. Existence of EFX for Two Additive Valuations. *CoRR* abs/2008.08798 (2020). arXiv:2008.08798 https://arxiv.org/abs/2008.08798

[20] R. Mahara. 2021. Extension of Additive Valuations to General Valuations on the Existence of EFX. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference) (LIPIcs, Vol. 204)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 66:1–66:15.

[21] H. Moulin. 2019. Fair Division in the Internet Age. *Annual Review of Economics* 11 (2019), 1–37.

[22] A. D. Procaccia. 2020. An answer to fair division's most enigmatic question: technical perspective. *Commun. ACM* 63, 4 (2020), 118.

[23] S. Zhou and X. Wu. 2021. Approximately EFX Allocations for Indivisible Chores. *CoRR* abs/2109.07313 (2021).