Towards Envy-Freeness Relaxations for General Nonmonotone Valuations

AAAI Track

Umang Bhaskar
Tata Institute of Fundamental Research
Mumbai, India
umang@tifr.res.in

Gunjan Kumar
Indian Institute of Technology Kanpur
Kanpur, India
gunjan@iitk.ac.in

Yeshwant Pandit
Tata Institute of Fundamental Research
Mumbai, India
yeshwant.pandit@tifr.res.in

Rakshitha
Indian Institute of Technology Delhi
Delhi, India
rakshitha.mt121@maths.iitd.ac.in

ABSTRACT

In the fair division of items among interested agents, envy-freeness is possibly the most favoured and widely studied formalization of fairness. For indivisible items, envy-free allocations may not exist in trivial cases, and hence research and practice focus on relaxations, particularly envy-freeness up to one item (EF1) and up to any item (EFX). Though EFX is a tighter relaxation, a significant reason for the popularity of EF1 is the simple fact of its existence.

This raises the question if in fact EF1 allocations exist for *all* valuations. Towards this objective, we present three results. We show that for *all valuations*, there exists an EFX allocation with charity, when some non-envied subset of items can remain unallocated. Secondly, we consider two new but natural classes of valuations: (i) Trilean valuations — an extension of Boolean valuations — when the value of any subset is 0, a, or b for any integers a and b, and (ii) Separable single-peaked valuations, when the set of items is partitioned into types. For each type, an agent's value is a single-peaked function of the number of items of the type. The value for a set of items is the sum of values for the different types. We prove the existence of complete EF1 allocations for identical trilean valuations for any number of agents and for separable single-peaked valuations for three agents. For both classes of valuations, we also show that complete EFX allocations do not exist.

KEYWORDS

Fair Division; EFX; Charity; EF1; Chores; Approximate Envy-Freeness; Trilean; Separable Single-Peaked

ACM Reference Format:
Umang Bhaskar, Gunjan Kumar, Yeshwant Pandit, and Rakshitha. 2025. Towards Envy-Freeness Relaxations for General Nonmonotone Valuations: AAAI Track. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Fair division refers to the fundamental problem of allocating a set of items — called manna — *fairly* among a set of agents. The items to be allocated could be as diverse as seats in courses in a university, items in a contested inheritance or divorce proceedings, carbon credits among nations, chores among members of a household, cab fare for a shared ride, household rent among roommates, etc. These problems are frequent and universal and have naturally attracted a lot of attention from researchers in various fields.

Given the widespread applications, a number of different formalizations of what it means to be fair have naturally been proposed. A popular and possibly predominant formalization is *envy-freeness*, which informally requires that each agent prefers her own allocated manna over the allocation to any other agent. When items are indivisible and have to be wholly allocated to a single agent, envy-free allocations may not exist, and hence relaxations are studied. The most prevalent relaxation is *envy-freeness up to one item (EF1)*, which allows envy among agents as long as this envy is eliminated by removing a single item [15, 30]. Envy-free allocations are a focus of theoretical research and also implemented in practical tools, e.g., spliddit.org [25, 37], and fairoutcomes.com [38].

In fair division, given a set M of items to be allocated among n agents, we assume that each agent i has a valuation function $v_i : 2^M \to \mathbb{Z}$ that specifies a value for each subset of items. An allocation $A = (A_1, \ldots, A_n)$ is a partition of M where agent i gets the set A_i. An allocation is EF1 if, whenever $v_i(A_i) < v_i(A_j)$, there is an item $x \in A_i \cup A_j$ so that $v_i(A_i \setminus \{x\}) \geq v_i(A_j \setminus \{x\})$, i.e., on removing item x, agent i weakly prefers her own allocation to j's allocation. A stricter notion than EF1 is EFX, where an allocation A is EFX if, whenever i envies j, this envy is resolved by removing *any* item with a positive marginal value (a good) from A_j, as well as *any* item with a negative marginal value (a chore) from A_i.[1]

For a formalization such as EFX or EF1 to be practically relevant, an important criterion is existence. If a notion of fairness is not easily satisfied or does not exist in an instance, its practical use is limited. Unfortunately, EFX allocations are not known to exist beyond three agents, even for nondecreasing additive valuations.

[1] An even stricter notion requires the envy to be resolved by removing any item with nonnegative marginal value from A_j, and any item with a nonpositive marginal value from A_i. This is made precise later.

With nonmonotone additive valuations, or non-additive valuations, EFX allocations are known to not exist [16, 21, 28]. However, if we allow some items to remain unallocated (or be donated to charity), then EFX allocations are known to exist for general monotone valuations [12, 20]. As a relaxation of EFX, EF1 allocations exist in very broad classes of valuations. The robustness of EF1 existence, as well as algorithms for obtaining such allocations, have encouraged research into broader objectives. For example, researchers have studied allocations that are EF1 satisfying other properties such as Pareto efficiency [8, 10, 24], strategy-proofness [2, 3]), and ex-ante envy-freeness [6, 22]).

Initial results established the existence of EF1 allocations for monotone valuations of the agents [30]. A different algorithm is known for nonmonotone but additive valuations when each agent has a positive or negative value for each item and values a subset at the sum of the individual item values [7, 8]. These were extended to the large class of doubly monotone valuations, where each agent partitions the items into goods and chores; the goods always have nonnegative marginal value for the agent, while chores always have nonpositive marginal values [14]. While this is a broad class, one particular property not captured by doubly monotone valuations is when "too much of a good thing is bad." Doubly monotone valuations require an agent's good to always be a good, no matter what other items the agent is allocated. But this is often not the case. For example:

- Our daily diet requires specific amounts of fats, sugars, proteins, etc. For each nutrient — fats, sugars, proteins, etc. — there is a recommended threshold. Consuming excess of a nutrient, or too little, can both be harmful. Thus an item which initially has positive dietary value, may have negative value if too many similar items have already been consumed.
- Another example is the composition of a research lab, consisting of postdocs, PhDs, undergraduates, etc. Typically a PI looks for an ideal number of each type of researcher. Both fewer and more researchers of a particular type may hinder the research agenda.

Moulin also provides an example of such valuations [32].

Towards establishign EF1 existence for arbitrary valuations, a possible approach to this goal is to consider discretized valuations, i.e., to restrict the range of possible values for any bundle to a smaller set, and obtain results on the existence of EF1. In this direction, prior work shows that EF1 allocations exist if (a) all agents have values that are in $\{0, 1\}$ for all sets, or (b) agents are identical, and have values in $\{0, -1\}$ for all sets [16].[2]

In this paper, we present results for both EFX and EF1 allocations. Firstly, we show that if we allow a subset of items to be donated to charity, then there always exists an EFX allocation for *arbitrary* valuations. The subset of items donated has the property that no agent envies them. Secondly, for complete EF1 allocations, we extend previous results on the existence in two directions.

(1) We show that EF1 always exists for identical *trilean* valuations — when agents are identical, and the value for each set of items is 0, a, or b for any integers a, b. Our work builds on prior work on Boolean $\{0, 1\}$ and $\{0, -1\}$ valuations.

Trilean valuations are naturally motivated from two different perspectives. Practically, trilean valuations allow agents to express dissatisfaction (with value = -1), neutrality or irrelevance (value = 0), or satisfaction (value = $+1$) with their allocation, which allows finer-grained user inputs. Theoretically, this is a step towards EF1 for arbitrary (though identical) valuations by considering discretized valuations, i.e., when all values lie in a finite set.

Along the way, we extend prior work to show that EF1 allocations also exist for non-identical $\{0, -1\}$-valuations.

(2) We introduce a new class of valuations that we call *separable single-peaked* (SSP) valuations. For SSP valuations, the set of items is partitioned into t types. For each type of item j, each agent has a threshold θ_{ij}. Agent i's valuation for type j is single-peaked with peak θ_{ij}: it monotonically increases with the number of items up to θ_{ij}, and monotonically decreases after that. The valuation is additive across items of different types. Thus, these are a relaxation of separable piece-wise linear concave valuations, widely studied in fair division and market equilibria (e.g., [19, 23]).

We note that both the examples considered previously — of getting balanced nutrients in a diet, and of research group composition — are naturally captured by separable single-peaked valuations. The types correspond to nutrients in the diet example, and type of researcher in the research group example.

For SSP valuations, we show two results: we show that EF1 allocations exist either when agents have the same threshold for a type, and when agents have different thresholds, but there are three agents. Finally, we give a tight example to show that complete EFX allocations do not exist for the two valuation classes studied, even for two identical agents and three items.

Because of space limitations, the missing algorithms and proofs are provided in the full version of the paper [13].

2 RELATED WORK

Fair division has traditionally focused on allocating divisible resources, also known as cake-cutting. A survey on computational results for cake-cutting is presented by Procaccia [35]. For non-monotone valuations (sometimes called "burnt cake"), results are known only when the number of agents is either 4 or a prime number [31, 36].

For indivisible manna, EFX allocations were studied by Plaut and Roughgarden [34], and shown to exist for general monotone valuations if there are two agents, or if all agents were identical. This was extended to three agents [20] and then to more agents of three types [29]. With nonmonotone additive valuations, or non-additive valuations, EFX allocations are known to not exist [16, 21, 28].

Various relaxations of EFX are also studied [4, 5, 34]. A particular relaxation of interest to us is EFX with charity, when certain items may remain unallocated, subject to constraints. The notion of allocations with charity was introduced by Caragiannis, Gravin and Huang [17]. They showed that with nondecreasing additive valuations, there exists a partial allocation that is EFX, and for which the Nash welfare (or the geometric mean of the agents' utilities) is at

[2] More specifically, existence is shown for the more restricted class of EFX allocations.

least half the maximum possible. The remaining items are said to be donated to charity. For nondecreasing monotone valuations, there exists a partial allocation that is EFX and where (i) no agent envies the set of unallocated items, and (ii) the number of unallocated items is at most $n-2$ [12, 20]. This is a remarkably strong result, since it holds for general monotone valuations.

The existence of EF1 allocations for monotone nondecreasing valuations — when all items are goods — was given by Lipton et al. [30]. For additive nonmonotone utilities a double round-robin algorithm for EF1 allocations was given by Aziz et al. [7, 8]. These results were extended to doubly monotone valuations [14] by suitably modifying the envy-cycle elimination algorithm of Lipton et al.

EF1 allocations always exist for two agents, with general valuations [16]. Further, EFX allocations exist (a) for Boolean valuations, i.e., $v_i(S) \in \{0,1\}$ for any agent i and set S of items, and (b) for identical and negative Boolean valuations, i.e., $v_i(S) = v(S) \in \{0,-1\}$ for all i, S. An EFX allocation is also an EF1 allocation. We note that these existence results are through a variety of different techniques — envy-cycle elimination, round-robin, sequential allocation of minimal subsets, local search, etc. It remains open if an EF1 allocation exists for arbitrary valuations, even, e.g., for the case of 3 identical agents.

EF1 is also considered alongside Pareto-optimality (PO), where an allocation is PO if no allocation gives at least as much utility to every agent and strictly higher utility to at least one agent. It is known that an EF1 and PO allocation always exists for additive goods and can be computed in pseudopolynomial time [10, 18, 33]. For additive chores and mixed items, partial results are known [8, 24, 27].

Besides envy-freeness, numerous other fairness notions are also studied in the literature, including proportionality, equitability, and maximin share. Amanatidis et al. [1] present a survey on recent developments on these.

3 BASIC NOTATION

A fair division instance with indivisible manna is specified by a set M of m items, a set N of n agents, and for each agent $i \in N$, a valuation function $v_i : 2^M \to \mathbb{Z}$ that specifies a value for each subset of items with $v_i(\emptyset) = 0$. We use $\mathcal{V} = (v_1, \ldots, v_n)$ to denote a valuation profile. Agents are identical if $v_i(S) = v(S)$ for all agents $i \in N$ and all subsets $S \subseteq M$. An allocation $A = (A_1, \ldots, A_n)$ is a partition of items where agent i gets the set A_i. The set of items allocated to an agent is also sometimes called a *bundle*. An allocation is *envy-free* if for all agents i, j, $v_i(A_i) \geq v_i(A_j)$, i.e., each agent prefers their own bundle to that of any other agent's. An allocation is *envy-free upto one item* (EF1) if, whenever $v_i(A_i) < v_i(A_j)$, there is some item $x \in A_i \cup A_j$ so that $v_i(A_i \setminus \{x\}) \geq v_i(A_j \setminus \{x\})$. Given an allocation A, we say a subset N' of agents is *mutually EF1* if the EF1 condition holds for every pair of agents in N'.

To formally define EFX allocations for nonmonotone valuations, for agent i and subset $S \subseteq M$, let $M_i^+(S) = \{x \in S \mid v_i(S) > v_i(S \setminus \{x\})\}$ be the set of items with positive marginal value (goods), $M_i^-(S) = \{x \in S \mid v_i(S) < v_i(S \setminus \{x\})\}$ be the set of items with negative marginal value (chores), and $M_i^0(S) = \{x \in S \mid v_i(S) = v_i(S \setminus \{x\})\}$ be the items with zero marginal value. Then allocation A is EFX_0^0 if, whenever agent i envies agent j, the set $M_i^+(A_j) \cup M_i^0(A_j) \cup M_i^0(A_i) \cup M_i^-(A_i)$ is nonempty, and for every $x \in M_i^+(A_j) \cup M_i^0(A_j) \cup M_i^0(A_i) \cup M_i^-(A_i)$, $v_i(A_i \setminus \{x\}) \geq v_i(A_j \setminus \{x\})$. A more relaxed version of EFX is EFX_-^+, which ignores items with zero marginal value: Allocation A is EFX_-^+ if, whenever agent i envies agent j, the set $M_i^+(A_j) \cup M_i^-(A_i)$ is nonempty, and for every $x \in M_i^+(A_j) \cup M_i^-(A_i)$, $v_i(A_i \setminus \{x\}) \geq v_i(A_j \setminus \{x\})$.

We introduce additional notation specific to the valuations we study in the relevant sections.

4 EFX WITH CHARITY

It is known that for nondecreasing monotone valuations, there exists a partial allocation that is EFX where (i) no agent envies the set of unallocated items, and (ii) the number of unallocated items is at most $n-2$ [12, 20]. The unallocated goods are said to be allocated to charity. This is a remarkably strong result since it holds for general monotone valuations.

We first show that these results can be nearly replicated for *arbitrary nonmonotone valuations*. That is, for n agents with any valuations, there exists a partial allocation $A = (A_1, \ldots, A_n)$ that is EFX, and for which no agent envies any subset of the unallocated items. Thus if $P = M \setminus \bigcup_{i \in N} A_i$ is the set of unallocated items, then for each agent i and any set $S \subseteq P$, $v_i(A_i) \geq v_i(P)$. We thus nearly obtain the same results as in prior work for a much broader class of valuations, but lose the bound on cardinality of the set of unallocated items. Algorithm 1 obtains the required partial allocation. We note that the algorithm is conceptually similar to previous algorithms which also look for minimally envied subsets (e.g., [16, 20]).

Algorithm 1 EFX-with-charity

Input: Fair division instance $(N, M, \mathcal{V})$.
Output: A partial EFX allocation A.
1: Initialize $A = (\emptyset, \ldots, \emptyset)$, $P = M$ {P is the set of unallocated items.}
2: **while** ($\exists i$ and items $S \subseteq P$ such that $v_i(A_i) < v_i(S)$) **do**
3: Let $S' \subseteq P$ be a minimum cardinality set so that for some agent i', $v_{i'}(A_{i'}) < v_{i'}(S')$
4: $P = P \cup A_{i'} \setminus S'$, $A_{i'} = S'$ {No agent envies $S' \setminus \{x\}$ for all $x \in S'$.}
5: **Return** $A = (A_1, \ldots, A_n)$

Theorem 1. *For agents with arbitrary valuations, Algorithm 1 returns a partial allocation A that is EFX_0^0, and so that no agent envies any subset of unallocated items.*

Proof. Firstly, we note that every time the while loop executes, the value for agent i' that gets set S' strictly increases, while the allocation to the other agents is unchanged. Hence there is a Pareto improvement with each iteration, and the algorithm must terminate in finite time. Secondly, clearly when the algorithm terminates, no agent envies any subset of P, the set of unallocated items (or P itself).

We thus need to show that the partial allocation A is EFX_0^0. Suppose i envies j in A. Since $v_i(A_i) \geq 0$, $A_j \neq \emptyset$. Then for all $x \in A_j$, $v_i(A_i) \geq v_i(A_j \setminus \{x\})$. Hence also $M_i^+(A_j) \neq \emptyset$. Further, if

$A_i \neq \emptyset$, then for all $x \in A_i$, $v_i(A_i) > v_i(A_i \setminus \{x\})$ (else A_i would not have been a minimum cardinality envied set when i received A_i). Thus $M_i^-(A_i) \cup M_i^0(A_i) = \emptyset$, and hence for all items $x \in M_i^+(A_j) \cup M_i^0(A_j) \cup M_i^0(A_i) \cup M_i^-(A_i)$, $v_i(A_i \setminus \{x\}) \geq v_i(A_j \setminus \{x\})$. □

5 TRILEAN VALUATIONS

For a fair division instance, valuations are trilean if for some $a, b \in \mathbb{Z}$, for every agent $i \in N$ and $S \subseteq M$, $v_i(S) \in \{0, a, b\}$. Valuations are Boolean $\{0, 1\}$-valued if for every agent i and subset S of items, $v_i(S) \in \{0, 1\}$. Similarly, valuations are Boolean $\{0, -1\}$-valued if for every agent i and subset S of items, $v_i(S) \in \{0, -1\}$.

To show EF1 exists for identical trilean valuations, we claim that it is sufficient to prove existence for two cases: $a = 1$, $b = 2$, and $a = -1$, $b = 1$. This is immediate if either a or b is nonnegative. It also holds if a and b are both negative, with the proof provided in the full version of the paper.

Proposition 2. *Suppose an EF1 allocation exists in all instances $(N, M, \mathcal{V})$ with identical agents, where either $v(S) \in \{0, 1, -1\}$ for all $S \subseteq M$, or $v(S) \in \{0, 1, 2\}$. Then an EF1 allocation exists in all instances with identical agents where $v(S) \in \{0, a, b\}$ for any integers a, b.*

Our proof for trilean valuations will thus focus on these two cases, called negative trilean if values are in $\{0, -1, 1\}$, and positive trilean if values are in $\{0, 1, 2\}$.

Since EF1 studies values of sets upon removal of items, we introduce some notation for this. For a set of items S, any immediate subset ($S' \subset S$ s.t. $|S'| = |S| - 1$) is called a *child* of S.

Definition 1. *For an agent i and a subset S of items, and $a, b \in \mathbb{Z}$,*

- *We use $v_i(S) = a \longrightarrow b$ to denote that $v_i(S) = a$, and for* some *item $x \in S$, $v_i(S \setminus \{x\}) = b$.*
- *We use $v_i(S) = a \rightrightarrows b$ to denote that $v_i(S) = a$, and for* every *$x \in S$, $v_i(S \setminus \{x\}) = b$.*
- *For $B \subset \mathbb{Z}$, we use $v_i(S) = a \rightrightarrows B$ to denote that $v_i(S) = a$, and for every $x \in S$, $v_i(S \setminus \{x\}) \in B$.*

For example, $v_i(S) = 0 \longrightarrow 1$ denotes that $v_i(S) = 0$, and $\exists x \in S$ for which $v_i(S \setminus \{x\}) = 1$. The notation $v_i(S) = -1 \rightrightarrows \{-1, 0\}$ denotes that $v_i(S) = -1$, and on removal of any $x \in S$, $v_i(S \setminus \{x\})$ is either -1 or 0.

A note on Boolean and trilean valuations. In our work, we use trilean for intances where any set of items has one of three values. We note that the term dichotomous valuations has been used earlier for the case where any item has 0 or 1 *marginal* value [9] (and thus a set of items can take any value between 0 and m). However various other terms have also been used for this case, including binary [11, 26] and bivalued instances [24]. Bérczi et al. [16] use Boolean and negative Boolean for instances where any set of items has one of two values. Given that there is a lack of consistent notation, and that we are the first to study this class of valuations, we believe the use of trilean to denote the valuations we study is a reasonable choice.

5.1 Boolean Valuations

As noted, Bérczi et al. show that for Boolean $\{0, 1\}$ valuations, there exists an EFX allocation, and give an algorithm for this [16]. They also give an algorithm for obtaining an EFX allocation for *identical* Boolean $\{0, -1\}$ valuations. Since EFX is a stronger requirement than EF1, these algorithms give EF1 allocations for the respective cases. This however leaves open the existence of EF1 allocations for nonidentical Boolean $\{0, -1\}$ valuations.[3] In the full version of the paper, we give such an algorithm, called NegBooleanEF1. Our algorithm is similar to the algorithm for Boolean $\{0, 1\}$ valuations, with suitable modifications for negative valuations. We use certain properties of the allocation produced by this algorithm later, in our result for trilean valuations (Proposition 6).

Theorem 3. *Given a fair division instance with negative Boolean valuations, Algorithm NegBooleanEF1 returns an EF1 allocation in polynomial time.*

5.2 Negative Trilean Valuations

In this section, we establish the existence of EF1 allocations when agents are identical and their valuations are negative trilean. We prove the following theorem in the remainder of this section.

Theorem 4. *Every instance with identical negative trilean valuations has an EF1 allocation.*

Before we give a brief description of how we achieve this, let us define a few terms.

While executing our algorithm, each agent will belong to one or more sets, depending on the bundle allocated to the agent. This classification forms the basis of our algorithm and the analysis, as it clarifies when EF1 violations occur. The conditions on EF1 violations are shown in Lemma 5.

(1) Unallocated: $\mathsf{U} = \{i : A_i = \emptyset\}$.
(2) Zero: $\mathsf{Zero} = \{i : v(A_i) = 0\}$.
(3) Favourable: $\mathsf{Fav} = \{i : v(A_i) = 1 \longrightarrow -1$ or $v(A_i) = -1 \longrightarrow 1\}$.
(4) Flexible: $\mathsf{Flex}^+ = \{i : v(A_i) = 0 \longrightarrow 1\}$, and $\mathsf{Flex}^- = \{i : v(A_i) = 0 \longrightarrow -1\}$.
(5) Resolved: $\mathsf{Res}^+ = \{i : v(A_i) = 1 \longrightarrow 0\}$, and $\mathsf{Res}^- = \{i : v(A_i) = -1 \longrightarrow 0\}$.
(6) Bad: $\mathsf{Bad}^+ = \{i : v(A_i) = 1 \rightrightarrows 1\}$, and $\mathsf{Bad}^- = \{i : v(A_i) = -1 \rightrightarrows -1\}$

Note that the above sets are not mutually exclusive (e.g., an agent could be in both Flex^- and Flex^+, and an agent in U is also in Zero), but are exhaustive, i.e., for any allocation A and identical trilean valuations for the agents, each agent i falls in one or more of the above sets. We will not explicitly move agents in and out of these sets. Rather, agents will acquire or lose membership depending on the bundle allocated to them.

We use these terms to describe the respective sets as well. Thus a set of items S is:

(1) Zero-valued if $v(S) = 0$.
(2) Favourable if $v(S) = 1 \longrightarrow -1$ or $v(S) = -1 \longrightarrow 1$.
(3) Flexible if $v(S) = 0 \longrightarrow 1$ or $v(S) = 0 \longrightarrow -1$.
(4) Resolved if $v(S) = 1 \longrightarrow 0$ or $v(S) = -1 \longrightarrow 0$.
(5) Bad if $v(S) = 1 \rightrightarrows 1$ or $v(S) = -1 \rightrightarrows -1$.

[3] For identical valuations, there is a reduction from finding an EF1 allocation in Boolean $\{0, -1\}$ valuations to finding one in Boolean $\{0, 1\}$ valuations — replace each -1 value with $+1$, and find an EF1 allocation A in the resulting $\{0, 1\}$-valued instance. Then A is an EF1 allocation in the original $\{0, -1\}$-valued instance as well. This reduction, however, does not work for *nonidentical* valuations.

Apart from the set Fav, agents in the same set have the same value for their bundles, and hence do not envy each other. Given an allocation A and a pair i, j of agents, we now use these sets to give necessary conditions for the violation of EF1.

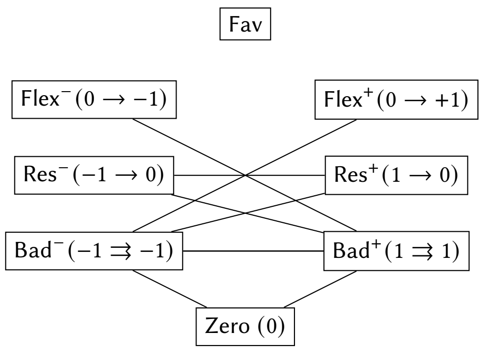

Figure 1: The edges depict possible EF1 violations between different sets of agents.

Lemma 5. *Given agents with identical negative trilean valuations and an allocation A, agents i, j are NOT mutually EF1 only if:*[4]

(1) $i \in Bad^+$, *and* $j \in Zero \cup Flex^- \cup Res^- \cup Bad^-$, *or*
(2) $i \in Bad^-$, *and* $j \in Zero \cup Flex^+ \cup Res^+ \cup Bad^+$, *or*
(3) $i \in Res^+$ *and* $j \in Res^-$.

In particular, note that agents in Fav are mutually EF1 with every other agent, regardless of the other agent's bundle. The lemma is depicted in Figure 1, with edges between various sets showing possible EF1 violations.

The name of the sets (favourable, flexible, resolved, and bad) are an attempt to describe how the sets are used in our algorithm as well. *Favourable* sets are good for us since they are mutually EF1 with all other sets. *Flexible* sets are the next best, since they have the least number of EF1 violations among the remaining sets. *Resolved* sets stay fixed once assigned: Once an agent is assigned a resolved set, her bundle does not change (an agent assigned a flexible set may however have her allocation modified later). Finally, a *bad* set is so named because in our algorithm, any EF1 violation must involve a bad set.

At a very high level, our algorithm proceeds by first allocating *favourable* sets until there are no more, then by allocating *flexible* sets until the remaining set of items is either zero-valued or Boolean-valued ($\{0, 1\}$ or $\{0, -1\}$). If the remaining set is zero-valued, we assign it and return the resulting allocation. By Lemma 5 this is EF1. If the remaining set is Boolean-valued, then we use the previous algorithms for Boolean values to allocate *resolved* and *zero* sets. Eventually, we will be left with possibly a single *bad* set. The resolved sets and the bad set assigned will be of the same sign (either Res^+ and Bad^+, or Res^- and Bad^-). It follows from Lemma 5 that the only possible EF1 violations will be between the flexible sets and the single bad set, which we will then fix in Algorithm FixEF1ViolationsNeg.

[4]Note that $\mathrm{U} \subseteq \mathrm{Zero}$.

We now describe our algorithm in more detail. We first allocate any favourable sets ($v(S) = 1 \to -1$ or $-1 \to 1$) to the first $n-1$ agents, as long as there are any favourable sets. If there are items remaining but no more favourable sets, and not all agents in $[n-1]$ are allocated, let M' be the set of items remaining. If $v(M') = 0$ (or $M' = \emptyset$), we assign M' to the next agent and return the resulting allocation. Else, we next try and find *maximal* flexible sets. That is, if $v(M') = 1$, we look for a maximal set S such that $v(S) = 0 \to -1$, and if $v(M') = -1$, we look for a maximal set S such that $v(S) = 0 \to 1$. If M' is trilean, such a set S must exist: since there are no favourable sets, if M' and $S \subset M'$ have opposite signs, there must exist a set T so that $S \subset T \subset M'$ and $v(T) = 0$.

If after allocating any flexible sets to agents in $[n-1]$, at least two agents remain and $v(M') = 0$, assign M' to the next agent and return the resulting allocation, which is EF1. If $v(M') \neq 0$ then M' is either Boolean $\{0, 1\}$-valued or Boolean $\{0, -1\}$-valued. We then call the respective algorithms for obtaining EF1 allocations for these respective cases. This completes the allocation, though there may be EF1 violations. We then call Algorithm FixEF1ViolationsNeg to fix any violations.

The remaining case is that all agents in $[n-1]$ have been allocated either favourable or flexible sets. In this case, we allocate the remaining items to agent n, and call Algorithm FixEF1ViolationsNeg to fix any EF1 violations.

We will use the algorithms for obtaining EF1 allocations for Boolean $\{0, 1\}$ and $\{0, -1\}$ valuations as subroutines. The next two propositions state some properties of the allocations returned by these algorithms. For Boolean $\{0, 1\}$-valuations, we use Algorithm 2 from Bèrczi et al. 2024, which we will call Algorithm BooleanEF1. For Boolean $\{0, -1\}$-valuations, we use Algorithm NegBooleanEF1.

Proposition 6. *For identical* $\{0, -1\}$*-valuations, the allocation returned by the Algorithm NegBooleanEF1 satisfies one or both of the following conditions.*

(1) *Each agent is in* Res^- *or Zero.*
(2) *The first* $n-1$ *agents are in* Res^-.

Proposition 7. *For identical* $\{0, 1\}$*-valuations, the allocation returned by Algorithm BooleanEF1 satisfies one or both of the following conditions:*

(1) *Each agent is in* Res^+ *or Zero.*
(2) *The first* $n-1$ *agents are in* Res^+.

Algorithm TernaryNegEF1 terminates in one of five places — Lines 13, 15, 17, 20, and 23. The next claim shows that if the algorithm terminates in Line 13 or Line 15, the allocation returned is EF1.

Claim 8. *If Algorithm TernaryNegEF1 terminates in Line 13 or Line 15, the allocation returned is EF1.*

Proof. Note that prior to Lines 13 and 15, any agents with non-empty bundles were assigned either favourable sets or flexible sets in the preceding while loops. In the execution of Lines 13 or 15, every remaining agent is either unassigned or assigned a zero-valued bundle. From Lemma 5, there is no EF1 violation between favourable, flexible, and zero-valued agents. Hence the resulting allocation is EF1. ◻

Algorithm 2 TernaryNegEF1

Input: Fair division instance $(N, M, \mathcal{V})$ with identical negative trilean valuations.
Output: An EF1 allocation A.

1: Initialize $A = (\emptyset, \dots, \emptyset)$, $M' = M$, and $i = 1$.
2: **while** ($\exists S \subseteq M'$ such that S is favourable) AND ($i < n$) **do**
3: $\quad A_i = S$, $M' = M' \setminus S$, $i = i + 1$ {Assign favourable sets.}
4: **while** ($M' \neq \emptyset$) AND ($i < n$) AND (M' is trilean) AND ($v(M') \neq 0$) **do** {Assign flexible sets.}
5: $\quad$ **if** $v(M') = 1$ **then**
6: $\quad\quad$ Let S be an inclusion-wise maximal subset such that $v(S) = -1$.
7: $\quad\quad$ Pick any $x \notin S$. $A_i = S \cup \{x\}$, $M' = M' \setminus A_i$. $\{v(A_i) = 0 \to -1.\}$
8: $\quad$ **else**
9: $\quad\quad$ Let S be a inclusion-wise maximal subset such that $v(S) = 1$.
10: $\quad\quad$ Pick any $x \notin S$. $A_i = S \cup \{x\}$, $M' = M' \setminus A_i$. $\{v(A_i) = 0 \to 1.\}$
11: $\quad$ $i = i + 1$
12: **if** ($M' = \emptyset$) **then**
13: $\quad$ **return** allocation A.
14: **if** ($v(M') = 0$) **then**
15: $\quad A_i = M'$, **return** allocation A.
16: **if** ($i = n$) **then**
17: $\quad A_i = M'$, $A =$ FixEF1ViolationsNeg(A), **return** allocation A.
18: **if** (M' is Boolean $\{0, 1\}$-valued) **then** {Assign resolved sets.}
19: $\quad A =$ Algorithm BooleanEF1($M', N \setminus [i - 1], \mathcal{V}$).
20: $\quad A =$ FixEF1ViolationsNeg(A), **return** allocation A.
21: **else**
22: $\quad A =$ Algorithm NegBooleanEF1($M', N \setminus [i - 1], \mathcal{V}$).
23: $\quad A =$ FixEF1ViolationsNeg(A), **return** allocation A.

If Line 17 executes, then before FixEF1ViolationsNeg(A) is called, agents $1, \dots, n - 1$ are assigned either favourable or flexible sets, and hence by Lemma 5, any EF1 violation must involve agent n. Claim 10 then details the possible EF1 violations before Algorithm FixEF1ViolationsNeg is called.

For the remaining Lines 20 and 23, the following proposition states what the allocation looks like before FixEF1ViolationsNeg(A) is called.

Claim 9. *Let k be the last agent to be allocated in the while loop in Algorithm TernaryNegEF1. Then clearly agents $1, \dots, k$ are either favourable or flexible. Further,*

(1) If FixEF1ViolationsNeg(A) is called in Line 20, then either:
  (a) agents $k + 1, \dots, n$ are in $\text{Res}^+ \cup$ Zero, or
  (b) agents $k + 1, \dots, n - 1$ are in Res^+, and A_n is Boolean $\{0, 1\}$-valued.

(2) If FixEF1ViolationsNeg(A) is called in Line 23, then either:
  (a) agents $k + 1, \dots, n$ are in $\text{Res}^- \cup$ Zero, or
  (b) agents $k+1, \dots, n-1$ are in Res^-, and A_n is Boolean $\{0, -1\}$-valued.

Proof. Prior to calling FixEF1ViolationsNeg(A) in Line 20, the algorithm calls Algorithm BooleanEF1 with agents $N \setminus [k]$ and items M' that are Boolean $\{0, 1\}$-valued. From Proposition 7, either each agent $i > k$ is in $\text{Res}^+ \cup$ Zero, or agents $k + 1, \dots, n - 1$ are in Res^+, and A_n is Boolean $\{0, 1\}$-valued. This proves the claim for Line 20. A similar proof (using Proposition 6) shows the claim for Line 23. □

We then have the following claim, regarding possible EF1 violations in the allocation passed to Algorithm FixEF1ViolationsNeg.

Claim 10. *Let A be the allocation given as input to Algorithm FixEF1ViolationsNeg. Then any EF1 violation must be of one of the following types:*

- *Type 1: Agent $i \in [n - 1]$ is in Flex^- and agent n is in Bad^+. Other agents are favourable, flexible, or in Res^+.*
- *Type 2: Agent $i \in [n - 1]$ is in Flex^+ and agent n is in Bad^-. Other agents are favourable, flexible, or in Res^-.*

Finally, in Algorithm FixEF1ViolationsNeg, we resolve any violations with agent n. Claim 10 tells us that any EF1 violations must be between agent n and an agent i assigned a flexible set. Further, in this case, n must be a bad agent ($v(A_n) = 1 \rightrightarrows 1$ or $v(A_n) = -1 \rightrightarrows -1$), and i must be a flexible agent of the opposite sign ($v(A_i) = 0 \to -1$ or $v(A_i) = 0 \to 1$, respectively). Assume $n \in \text{Bad}^+$. Then Algorithm FixEF1ViolationsNeg proceeds by picking an agent i in Flex^- and transferring items (arbitrarily picked) from A_n to A_i, until at least one of them is in Res^+ (has value $1 \to 0$). If $n \in \text{Res}^+$, we have reached an EF1 allocation. Else, the set of agents in Flex^- is reduced. We then pick the next agent i from Flex^- and continue transferring items from A_n to A_i.

We call the `repeat...until` loops in the algorithm the *inner* loops, and the while loops the *outer* loops. Note that if the initial allocation is not EF1, then from Claim 10 either $n \in \text{Bad}^+$ or $n \in \text{Bad}^-$, and hence exactly one of the two `if` conditions holds true.

We claim that when an inner loop terminates, either agent n is resolved and the algorithm terminates with an EF1 allocation, or the chosen agent i is in resolved and n is in Bad. We first show that each inner `repeat...until` loop runs for at most $|A_n| - 1$ iterations over all iterations of the outer while loop.

Claim 11. *Let $t = |A_n|$ be the initial size of A_n. Each inner loop terminates in at most $t - 1$ iterations over all invocations.*

Claim 12. *After every iteration of the first while loop (Line 8), either agent n moves from Bad^+ to Res^+ and the algorithm returns an EF1 allocation, or agent i moves from Flex^- to Res^+ and agent n remains in Bad^+.*

Similarly, after every iteration of the second while loop (Line 15), either agent n moves from Bad^- to Res^- and the algorithm returns an EF1 allocation, or agent i moves from Flex^+ to Res^- and agent n remains in Bad^-.

We now complete the proof of our main theorem, showing existence of EF1.

Proof of Theorem 4. We show that Algorithm TernaryNegEF1 returns an EF1 allocation. By Claim 8, if Algorithm TernaryNegEF1 terminates in Line 13 or Line 15, the allocation returned is EF1. Otherwise, the algorithm calls Algorithm FixEF1ViolationsNeg to fix the allocation. Claim 10 then shows that for the allocation passed

Algorithm 3 FixEF1ViolationsNeg

Input: An allocation A with possible EF1 violations.
Output: An EF1 allocation A.

if (Allocation A is EF1) **then**
  return allocation A.
if ($n \in \mathsf{Bad}^+$) AND ($\mathsf{Flex}^- \neq \emptyset$) **then** $\{v(A_n) = 1 \rightrightarrows 1$ and $\exists i : v(A_i) = 0 \to -1\}$
  while ($n \in \mathsf{Bad}^+$) AND ($\mathsf{Flex}^- \neq \emptyset$) **do**
    Let $i \in \mathsf{Flex}^-$.
    repeat
      Choose an item $x \in A_n$, $A_n = A_n \setminus \{x\}$, $A_i = A_i \cup \{x\}$.
    until ($i \in \mathsf{Res}^+$) OR ($n \in \mathsf{Res}^+$) {Either ($v(A_i) = 1 \to 0$) or ($v(A_n) = 1 \to 0$).}
  return allocation A.
if ($n \in \mathsf{Bad}^-$) AND ($\mathsf{Flex}^+ \neq \emptyset$) **then** $\{v(A_n) = -1 \rightrightarrows -1$ and $\exists i : v(A_i) = 0 \to 1\}$
  while ($n \in \mathsf{Bad}^-$) AND ($\mathsf{Flex}^+ \neq \emptyset$) **do**
    Let $i \in \mathsf{Flex}^+$.
    repeat
      Choose an item $x \in A_n$, $A_n = A_n \setminus \{x\}$, $A_i = A_i \cup \{x\}$.
    until ($i \in \mathsf{Res}^-$) OR ($n \in \mathsf{Res}^-$) {Either ($v(A_i) = -1 \to 0$) or ($v(A_n) = -1 \to 0$).}
  return allocation A.

to Algorithm FixEF1ViolationsNeg, any EF1 violation must be of either Type 1 or Type 2.

Suppose the EF1 violation is of Type 1. The case when the violation is of Type 2 is handled similarly. For a Type 1 violation, there must be agents in Flex^-, agent n is in Bad^+, and the other agents are in Fav, Res^+, or Flex^+. Since there is an EF1 violation, and agent n is in Bad^+, the first while loop in Algorithm FixEF1ViolationsNeg will execute, selecting an agent i from Flex^-. From Claim 12, each time the inner loop terminates, either agent n moves from Bad^+ to Res^+ and the algorithm returns an EF1 allocation, or agent i moves from Flex^- to Res^+ and agent n remains in Bad^+ (and note that by Claim 11, the inner loop terminates in at most m iterations). Thus, eventually, the algorithm either moves agent n from Bad^+ to Res^+ and returns an EF1 allocation, or moves all the agents in Flex^- to Res^+. At this point, all agents are Fav, Flex^+, Res^+, or Bad^+. By Lemma 5, this is an EF1 allocation. □

We note that our algorithm may need to make an exponential number of queries, e.g., in the very first step to find favourable sets. Given the lack of structure in the problem, we do not know if there is a way of avoiding this.

5.3 Positive Trilean Valuations

We now turn our attention to the case where agents are identical and their valuations are positive trilean, i.e., $v_i(S) \in \{0, 1, 2\}$ for all agents $i \in N$ and subsets $S \subseteq M$. Similar to before, we show the existence of EF1 valuations. The algorithm is similar to the case of negative ternary valuations, with some simplifications and modifications due to the positive valuations.

THEOREM 13. *Given an instance with identical positive trilean valuations, Algorithm TernaryPosEF1 returns an EF1 allocation.*

6 SEPARABLE SINGLE-PEAKED VALUATIONS

We now turn to EF1 allocations for separable, single-peaked (SSP) valuations. As before, N and M denote the set of agents and items respectively. The set M is partitioned into t types $(M_1, \dots, M_t)$, with $m_j = |M_j|$ for $j \in [t]$. For an allocation A, we denote agent i's bundle A_i as a t-tuple $(a_{i1}, a_{i2}, ..., a_{it})$, where a_{ij} denotes the number of items of type j allocated to agent i.

To define the valuation functions for the agents, we first define θ_{ij} for $i \in N$ and $j \in [t]$ as the threshold of agent i for items of type j. Then agent i's valuation $v_i(A_i) = \sum_{j=1}^{t} v_{ij}(a_{ij})$, where the valuations v_{ij} are single-peaked: for all $x \le y \le \theta_{ij}$, $v_{ij}(x) \le v_{ij}(y)$, while for $\theta_{ij} \le x \le y$, $v_{ij}(x) \ge v_{ij}(y)$.

We point out one basic problem that occurs with SSP valuations, that must be overcome by an algorithm returning EF1 allocations. Consider a simple instance with just two agents and a partial EF1 allocation A, where agent 1 envies agent 2. Suppose there is an item x that remains to be assigned. Item x is a chore for agent 1 given A_1 *and is a good for agent 1 given* A_2. That is, $v_1(A_1 \cup \{x\}) < v_1(A_1)$, and $v_1(A_2 \cup \{x\}) > v_2(A_2)$. There is no obvious way to assign item x while maintaining the EF1 property. This situation does not arise with doubly-monotone valuations.

For SSP valuations, we show two results. Firstly, we show existence of EF1 allocations when for each type j, all agents have the same threshold θ_j. In this case, our proof shows that the two-phase algorithm for doubly monotone valuations [14] works in this case as well. We reproduce the algorithm and provide the complete proof in the full version of the paper.

THEOREM 14. *For SSP valuations where for each type, there is a common threshold for all agents, there always exists an EF1 allocation.*

Secondly, for the general case where agents may have different thresholds for a type, we show the existence of EF1 allocations for three agents. Here, the algorithm considerably differs from the algorithm for doubly-monotone valuations. While it is still a two-phase algorithm, the two phases are very carefully crafted for the special case of three agents. We present the algorithm here, and the proof is provided in the full version of the paper.

THEOREM 15. *For SSP valuations with 3 agents, there always exists an EF1 allocation.*

For the algorithm, given a partial allocation A, as in prior work, we let $G_A = (V, E)$ denote the envy graph, where $V = N$ and $(i, k) \in E$ if agent i envies agent k. T_A is the top-trading envy graph, a subgraph of G_A with a directed edge (i, k) if $v_i(A_k) = \max_{i' \in N} v_i(A_{i'}) > v_i(A_i)$. Given a directed cycle C in G_A or T_A, A_C is the allocation obtained by giving each agent in C the bundle of the agent they envy in C.

7 NON-EXISTENCE OF EFX ALLOCATIONS

Given the existence of EF1 allocations, a natural question is whether EFX allocations exist for the valuations studied. Bérczi et al. refine the definition of EFX for nonmonotone valuations [16]. They define an EFX^+_- allocation A as one where if agent i envies agent j, then this should be resolved by removing any item with a strictly positive marginal value from A_j, and any item with a strictly negative marginal value from A_i. Further at least one such item must exist.

Algorithm 4 An EF1 algorithm for separable single-peaked valuations with 3 agents

Initialise A_i to $(0, 0, ..., 0)$ $\forall$ $i = 1, 2, 3$
Initialise $\bar{m}_j \leftarrow \lfloor m_j/3 \rfloor$, $\hat{m}_j \leftarrow m_j \bmod 3$, $N_j \leftarrow \{i : \theta_{ij} > \bar{m}_j\}$ $\forall$ $j \in [t]$
for all $j \in [t]$ s.t. $|N_j| \geq \hat{m}_j$ **do** {Phase 1}
  while G_A has a cycle C **do**
    $A \leftarrow A_C$ {Swap bundles along C}
  Let 1, 2, 3 be the topological order of the agents in G_A
  $a_{ij} \leftarrow \bar{m}_j$ for each agent i
  for $i = 1$ to 3 **do**
    if $i \in N_j$ and $\hat{m}_j > 0$ **then**
      $a_{ij} \leftarrow a_{ij} + 1$
      $\hat{m}_j \leftarrow \hat{m}_j - 1$
for all $j \in [t]$ s.t. $\hat{m}_j > |N_j|$ **do** {Phase 2}
  $C \leftarrow$ any cycle in T_A, $A \leftarrow A_C$ {Swap bundles along C}
  $a_{ij} \rightarrow \bar{m}_j$ for each agent i
  if $|N_j| = 0$ **then**
    while $\hat{m}_j > 0$ **do**
      Choose a sink k in the graph G_A
      $a_{kj} \leftarrow a_{kj} + 1$, $\hat{m}_j \leftarrow \hat{m}_j - 1$
      $C \leftarrow$ any cycle in T_A, $A \leftarrow A_C$ {Swap bundles along C}
  else if $|N_j| = 1$ **then**
    Let $N_j = \{k\}$, and let ℓ be a sink in G_A.
    $a_{kj} \leftarrow a_{kj} + 1$
    $a_{\ell j} \leftarrow a_{\ell j} + 1$
Return A

They show that for identical negative Boolean valuations, as well as for positive Boolean valuations, an EFX^+_- allocation always exists.

To complete the picture, we now show that EFX^+_- allocations do not exist for identical negative trilean valuations, or for separable single-peaked valutions, even with two identical agents and three items of a single type. In fact the same example shows nonexistence for both valuation classes.

Theorem 16. *EFX^+_- allocations may not exist even for two agents with identical negative trilean valuations, or two agents with identical SSP valuations.*

Proof. Consider an instance with three items and two identical agents. For $S \subseteq M$, $v(S) = 0$ if $S = \emptyset$, 1 if $|S| = 1$, and -1 if $|S| \geq 2$. This valuation is clearly both negative trilean and separable single-peaked with a single type.

To show non-existence of EFX^+_- there are two cases to consider: (i) agent 1 gets nothing, and (ii) agent 1 gets one item. In case (i), since $v(A_1) = 0$ and $v(A_2) = -1$, agent 2 envies agent 1. However $M^+_2(A_1) \cup M^-_2(A_2) = \emptyset$, and hence this allocation is not EFX^+_-. In case (ii), $v(A_1) = 1$ and $v(A_2) = -1$ and again agent 2 envies agent 1. However removing the single item from A_1 does not remove the envy, and hence this allocation is also not EFX^+_-. □

8 CONCLUSION

Our paper extends work on the existence of relaxations of envy-free allocations in multiple directions. Firstly, for EFX allocations, we show that for arbitrary valuations, a partial EFX^0_0 allocation where no agent envies any subset of unallocated items exists. We also show that complete EFX^+_- do not exist, even for two identical agents and three items.

We then define two classes of nonmonotone valuations — trilean valuations, and separable single-peaked valuations, and study complete EF1 allocations in these classes. We view separable single-peaked valuations as a natural class of valuations to study the existence of EF1 allocations. As mentioned, these generalize the well-known class of SPLC valuations.

Of the two, it appears likely that our algorithm and the structures introduced for trilean valuations may be useful in further extending results on the existence of EF1 to general identical valuations. For example, if max and min are the maximum and minimum possible values for any set, then a set S with value $v(S) = \max \rightarrow \min$ or $v(S) = \min \rightarrow \max$ should be assigned immediately, similar to how we treat favourable sets. More directly, just as our algorithm uses algorithms for Boolean $\{0, 1\}$ and $\{0, -1\}$ valuations as subroutines, it is possible that algorithms for k-ary valuations use algorithms for $(k-1)$-ary valuations as subroutines.

The big open question that remains open is the existence of EF1 allocations, even for three agents with identical valuations. More immediately, EF1 existence is left open for agents with nonidentical trilean valuations, and for more than three agents with separable single-peaked valuations. It is also an open question if one can obtain a partial EFX allocation where at most $(n-2)$ items are donated to charity, and no agent envies the items donated, replicating previous bounds for monotone valuations.

ACKNOWLEDGMENTS

The first and third author acknowledge support from the Department of Atomic Energy, Government of India, under project no. RTI4001. Part of the work was done when the second author was at NUS, and fourth author was visiting TIFR.

REFERENCES

[1] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A Voudouris, and Xiaowei Wu. 2023. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence* (2023), 103965.
[2] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. 2017. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 545–562.
[3] Georgios Amanatidis, Georgios Birmpas, Federico Fusco, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. 2023. Allocating indivisible goods to strategic agents: Pure nash equilibria and fairness. *Mathematics of Operations Research* (2023).
[4] Georgios Amanatidis, Aris Filos-Ratsikas, and Alkmini Sgouritsa. 2024. Pushing the Frontier on Approximate EFX Allocations. In *Proceedings of the 25th ACM Conference on Economics and Computation, EC 2024, New Haven, CT, USA, July 8-11, 2024*, Dirk Bergemann, Robert Kleinberg, and Daniela Sabán (Eds.). ACM, 1268–1286.
[5] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. 2020. Multiple Birds with One Stone: Beating 1/2 for EFX and GMMS via Envy Cycle Elimination. *Theoretical Computer Science* 841 (2020), 94–109.
[6] Haris Aziz. 2020. Simultaneously achieving ex-ante and ex-post fairness. In *Web and Internet Economics: 16th International Conference, WINE 2020, Beijing, China, December 7–11, 2020, Proceedings 16*. Springer, 341–355.
[7] Haris Aziz, Ioannis Caragiannis, and Ayumi Igarashi. 2018. Fair allocation of combinations of indivisible goods and chores. *CoRR* abs/1807.10684 (2018).
[8] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. 2019. Fair Allocation of Indivisible Goods and Chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 53–59.
[9] Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2021. Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5119–5126.

[10] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Finding Fair and Efficient Allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation*. 557–574.

[11] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*. 7–13.

[12] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. 2022. Almost Full EFX Exists for Four Agents. In *AAAI*. AAAI Press, 4826–4833.

[13] Umang Bhaskar, Gunjan Kumar, Yeshwant Pandit, and Rakshitha. 2024. EF1 Allocations for Identical Trilean and Separable Single-Peaked Valuations. *arXiv preprint arXiv:2411.19881* (2024).

[14] Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. 2021. On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources. In *APPROX)*, Vol. 207. 1:1–1:23.

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

[16] Kristóf Bérczi, Erika R. Bérczi-Kovács, Endre Boros, Fekadu Tolessa Gedefa, Naoyuki Kamiyama, Telikepalli Kavitha, Yusuke Kobayashi, and Kazuhisa Makino. 2024. Envy-free relaxations for goods, chores, and mixed items. *Theoretical Computer Science* 1002 (2024), 114596.

[17] Ioannis Caragiannis, Nick Gravin, and Xin Huang. 2019. Envy-Freeness Up to Any Item with High Nash Welfare: The Virtue of Donating Items. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, Anna R. Karlin, Nicole Immorlica, and Ramesh Johari (Eds.). ACM, 527–545.

[18] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation* 7, 3 (2019), 12.

[19] Bhaskar Ray Chaudhury, Yun Kuen Cheung, Jugal Garg, Naveen Garg, Martin Hoefer, and Kurt Mehlhorn. 2022. Fair Division of Indivisible Goods for a Class of Concave Valuations. *J. Artif. Intell. Res.* 74 (2022), 111–142.

[20] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. 2021. A Little Charity Guarantees Almost Envy-Freeness. *SIAM J. Comput.* 50, 4 (2021), 1336–1358.

[21] Vasilis Christoforidis and Christodoulos Santorinaios. 2024. On the Pursuit of EFX for Chores: Non-existence and Approximations. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 2713–2721.

[22] Rupert Freeman, Nisarg Shah, and Rohit Vaish. 2020. Best of Both Worlds: Ex-Ante and Ex-Post Fairness in Resource Allocation. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 21–22.

[23] Jugal Garg, Ruta Mehta, Milind A. Sohoni, and Vijay V. Vazirani. 2015. A Complementary Pivot Algorithm for Market Equilibrium under Separable, Piecewise-Linear Concave Utilities. *SIAM J. Comput.* 44, 6 (2015), 1820–1847.

[24] Jugal Garg, Aniket Murhekar, and John Qin. 2022. Fair and Efficient Allocations of Chores under Bivalued Preferences. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*. 5043–5050.

[25] Jonathan Goldman and Ariel D Procaccia. 2015. Spliddit: Unleashing Fair Division Algorithms. *ACM SIGecom Exchanges* 13, 2 (2015), 41–46.

[26] Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. 2020. Fair Division with Binary Valuations: One Rule to Rule Them All. In *Web and Internet Economics (WINE) (Lecture Notes in Computer Science, Vol. 12495)*. Springer, 370–383.

[27] Hadi Hosseini, Aghaheybat Mammadov, and Tomasz Wąs. 2024. Almost Envy-Freeness under Weakly Lexicographic Preferences. *arXiv preprint arXiv:2404.19740* (2024).

[28] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2023. Fairly dividing mixtures of goods and chores under lexicographic preferences. *In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (2023).

[29] Vishwa Prakash HV, Pratik Ghosal, Prajakta Nimbhorkar, and Nithin Varma. 2024. EFX Exists for Three Types of Agents. *CoRR* abs/2410.13580 (2024).

[30] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On Approximately Fair Allocations of Indivisible Goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*. 125–131.

[31] Frédéric Meunier and Shira Zerbib. 2019. Envy-Free Cake Division Without Assuming the Players Prefer Nonempty Pieces. *Israel Journal of Mathematics* 234, 2 (2019), 907–925.

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

[33] Aniket Murhekar and Jugal Garg. 2021. On Fair and Efficient Allocations of Indivisible Goods. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 5595–5602.

[34] Benjamin Plaut and Tim Roughgarden. 2020. Almost Envy-Freeness with General Valuations. *SIAM Journal on Discrete Mathematics* 34, 2 (2020), 1039–1068.

[35] Ariel D Procaccia. 2015. Cake Cutting Algorithms. In *Handbook of Computational Social Choice, Chapter 13*. Citeseer.

[36] Erel Segal-Halevi. 2018. Fairly Dividing a Cake after Some Parts were Burnt in the Oven. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 1276–1284.

[37] Nisarg Shah. 2017. Spliddit: two years of making the world fairer. *XRDS: Crossroads, The ACM Magazine for Students* 24, 1 (2017), 24–28.

[38] Christopher Taylor, James Ring, John Chu, Steven Brams, and William Hazel. 2006. *Fair Outcomes, Inc.* https://www.crunchbase.com/organization/fair-outcomes