Theoretical Computer Science 1002 (2024) 114596

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Envy-free relaxations for goods, chores, and mixed items ☆

Kristóf Bérczi [a,*], Erika R. Bérczi-Kovács [a], Endre Boros [b], Fekadu Tolessa Gedefa [c], Naoyuki Kamiyama [d], Telikepalli Kavitha [e], Yusuke Kobayashi [f], Kazuhisa Makino [f]

[a] MTA-ELTE Matroid Optimization Research Group and HUN-REN-ELTE Egerváry Research Group, Department of Operations Research, ELTE Eötvös Loránd University, Budapest, Hungary
[b] MSIS Department and RUTCOR, Rutgers University, New Jersey, USA
[c] Department of Operations Research, Eötvös Loránd University, Budapest, Hungary
[d] Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan
[e] School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India
[f] Research Institute for Mathematical Sciences (RIMS), Kyoto University, Kyoto, Japan

ARTICLE INFO

Communicated by X. Deng

Keywords:
Fair division
Indivisible items
Envy-freeness
Non-monotone utility function
Non-additive utility function

ABSTRACT

In fair division problems, we are given a set S of m items and a set N of n agents with individual preferences, and the goal is to find an allocation of items among agents so that each agent finds the allocation fair. There are several established fairness concepts and envy-freeness is one of the most extensively studied ones. However envy-free allocations do not always exist when items are indivisible and this has motivated relaxations of envy-freeness: envy-freeness up to one item (EF1) and envy-freeness up to any item (EFX) are two well-studied relaxations. We consider the problem of finding EF1 and EFX allocations for utility functions that are not necessarily monotone, and propose four possible extensions of different strength to this setting.

In particular, we present a polynomial time algorithm for finding an EF1 allocation for two agents with arbitrary utility functions. An example is given showing that EFX allocations need not exist for two agents with non-monotone, non-additive, identical utility functions. However, when all agents have monotone (not necessarily additive) identical utility functions, we give a pseudo-polynomial time algorithm that always finds an EFX allocation of chores. As a step toward understanding the general case, we discuss two subclasses of utility functions: Boolean utilities that are $\{0,+1\}$-valued functions, and negative Boolean utilities that are $\{0,-1\}$-valued functions. For the latter, we give a polynomial time algorithm that finds an EFX allocation when the utility functions are identical.

1. Introduction

Fair division of items among competing agents is an important and well-studied problem in Economics and Computer Science. There is a set S of m indivisible items and a set N of n agents with individual preferences, and the goal is to find an allocation

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

* Corresponding author.
E-mail addresses: kristof.berczi@ttk.elte.hu (K. Bérczi), erika.berczi-kovacs@ttk.elte.hu (E.R. Bérczi-Kovács), endre.boros@rutgers.edu (E. Boros), fekadut@student.elte.hu (F.T. Gedefa), kamiyama@imi.kyushu-u.ac.jp (N. Kamiyama), kavitha@tifr.res.in (T. Kavitha), yusuke@kurims.kyoto-u.ac.jp, makino@kurims.kyoto-u.ac.jp (Y. Kobayashi).

https://doi.org/10.1016/j.tcs.2024.114596
Received 25 May 2022; Received in revised form 15 January 2024; Accepted 22 April 2024

Available online 26 April 2024

of items among agents so that each agent finds the allocation fair. There are several fairness concepts in the literature that use different metrics to measure the degree of equity or fairness. Among them, the possibly most compelling one is envy-freeness (EF), introduced by Foley [28] and Varian [42]. Every agent has a utility value for each subset of items: the i-th agent has a utility function $u_i : 2^S \to \mathbb{R}$. An allocation is envy-free if each agent finds the utility of her bundle at least as much as that of any other agent. Although envy-freeness provides a very natural criterion for the fairness of an allocation, such a solution does not always exist as shown by the following simple example: just consider an instance with two agents and a single item having positive utility for both of them. In order to circumvent the possible non-existence of envy-free allocations, two natural relaxations of envy-freeness have been studied.

In the setting where all utilities are non-negative (items are called *goods* in such a case) Lipton et al. [37] and Budish [19] introduced the notion of envy-freeness up to one good (EF1) which allows the presence of envies between agents – however any envy must be removable by deleting some good from the envied agent's bundle. Note that no good is really removed from the envied agent's bundle, this is just a thought experiment to measure the amount of envy the envious agent has towards the envied agent.

EF1 may seem too weak as a relaxation of envy-freeness. Consider the simple example given below where there are three goods a, b, c and two agents with *additive*[1] utilities. Both agents value c three times as much as a or b. The allocation where agent 1 gets $\{a\}$ and agent 2 gets $\{b, c\}$ is EF1 – however this allocation does not seem fair towards agent 1 since, e.g., item b could be also given to agent 1.

	a	b	c
Agent 1	1	1	3
Agent 2	1	1	3

As a less permissive relaxation, Caragiannis et al. [22] proposed envy-freeness up to any good (EFX), which requires that an agent's envy towards another bundle can be eliminated by removing *any* good from the envied bundle. Observe the subtle difference between the above definitions: when utility functions are additive, EF1 requires that for any pair of agents i, j, agent i's envy towards agent j can be eliminated by removing agent i's most valued good from agent j's bundle, while EFX requires that the same should hold even after removing agent i's least valued good from agent j's bundle. In the example given above, the allocation where one agent gets $\{a, b\}$ and the other agent gets $\{c\}$ is the only EFX allocation. Thus EFX, though strictly weaker than EF, is strictly stronger than EF1. As remarked in [20]: "*Arguably, EFX is the best fairness analog of envy-freeness of indivisible items*".

Most works on fair division focused on allocation of 'goods', i.e., utilities are always non-negative and monotone: the utility of any set is at least as much as each of its subsets. However, in practice, it might easily happen that an item is a *chore*, i.e., this is a task or responsibility that all agents find cumbersome, so the utility of a chore is zero or negative – hence utilities are no longer monotone. We can also have *mixed* items: so an item has positive utility for some agent and negative for another agent. Furthermore, the utility functions are not necessarily additive, hence the utility of a subset of items might be completely independent of the utilities of the individual items within.

Our work continues the study of fair allocations in the setting of having both indivisible goods and chores, and aims at discovering this direction by showing the existence or non-existence of fair solutions in various settings. We also focus on the setting where all items are chores: so all utilities are non-positive here. As in the case of goods, envy-free allocations of chores need not exist. Consider the simple example with two agents and a single chore that neither agent wants to do: one of the agents has to be assigned this chore and she envies the other agent who is assigned no chore.

We formally define EF1 and EFX allocations in the generalized setting of goods and chores in Section 2. Roughly speaking, an allocation $\pi = \langle \pi(1), \dots, \pi(n) \rangle$ is EF1 if for any pair of agents i, j: either (1) i does not envy j's bundle, or (2) there is some item $s \in \pi(i) \cup \pi(j)$ such that i likes $\pi(i) - s$ at least as much as $\pi(j) - s$. Similarly, π is EFX if for any pair of agents i, j: (1) i does not envy j's bundle, or (2) i likes $\pi(i)$ at least as much as $\pi(j) - s$ for any 'good' $s \in \pi(j)$ and i likes $\pi(i) - s$ at least as much as $\pi(j)$ for any 'chore' $s \in \pi(i)$. Note that in the general case, whether an item s is a good or chore in $X \subseteq S$ is determined by the marginal utility that s brings to the set $X - s$. We refer to Section 2 for details.

1.1. Previous work

We only discuss works that are closely related to our results, for a detailed survey of the fair division literature we refer the reader to [3,4,9,38,43]. Further related work appears in Section 1.4.

EF1 allocations Envy-freeness and its relaxations were mainly considered for monotone additive utility functions. The idea of envy-freeness up to one good (EF1) implicitly appeared in the paper of Lipton et al. [37], and then was explicitly introduced and analyzed by Budish [19].

When allocating divisible goods, the maximum Nash welfare solution selects an allocation that maximizes the product of utilities and it is known to have strong fairness guarantees; moreover it also satisfies Pareto optimality.[2] Caragiannis et al. [22] showed that the maximum Nash welfare solution is fair in the indivisible setting as well – in the sense that it forms an EF1 allocation. In [12],

[1] For *additive* utilities, the utility of a subset X of items is the sum of utilities of items in X.

[2] An allocation is *Pareto optimal* if there is no other allocation which makes some agent strictly better off while making no agent worse off.

Table 1
Landscape of results on EF1 allocations, where bold letters represent the results obtained in the present paper. A utility function u is called *Boolean* if $u(X) \in \{0,1\}$ for every $X \subseteq S$, and *negative Boolean* if $-u$ is Boolean.

Utilities	Additive		Non-additive	
	Monotone	Non-monotone	Monotone	Non-monotone
Identical	Exists [7]	Exists [7]	Exists [37] (goods)	Open Double monotonic [16] **$n = 2$ (Theorem 1)** **Boolean (Theorem 7)** **Neg. Boolean (Theorem 9)**
Non-identical				Open Double monotonic [16] **$n = 2$ (Theorem 1)** **Boolean (Theorem 7)**

Barman et al. developed a pseudo-polynomial[3] time algorithm for finding EF1 allocations of goods that are also Pareto optimal, and showed that there always exists an allocation that is EF1 and fractionally Pareto optimal.

Much less is known when the utility functions are non-additive or non-monotone. Caragiannis et al. [21] considered allocation problems in which a set of divisible or indivisible goods or chores has to be allocated among several agents. Aziz et al. [6] focused on additive cardinal utility functions, and presented computational hardness results as well as polynomial-time algorithms for testing Pareto optimality for subclasses of utility functions. Barman and Krishnamurthy [11] considered additive and submodular utilities for goods, and provided fairness guarantees in terms of the maximin share, i.e., the maximum value that an agent can ensure for herself by partitioning the goods into n bundles, and receiving a minimum valued bundle. Benabbou et al. [13] considered the allocation of indivisible goods to agents that have monotone, submodular, non-additive utility functions. They showed that utilitarian socially optimal (hence Pareto optimal), leximin, and maximum Nash welfare allocations are all EF1 if in addition the utility functions have binary marginal gain.

Bogomolnaia et al. [17] compared the performance of the Competitive and the Egalitarian division rules for problems involving divisible goods and chores. Aziz et al. [7] proposed a similar scenario in the indivisible setting, and gave a generalization of the decentralized round robin algorithm that finds an EF1 allocation when the utilities are additive. Bhaskar et al. [16] showed that an EF1 allocation exists for the so-called doubly monotonic utilities,[4] that is, each agent i can partition S into sets G_i and C_i such that for any $X \subseteq S$, we have $u_i(X) \leq u_i(X+s)$ for $s \in G_i$ and $u_i(X) \geq u_i(X+s)$ for $s \in C_i$. Garg et al. [32] considered EF1 allocations with the notion of Pareto-optimality, and gave a strongly polynomial-time algorithm for computing an EF1+PO allocation when agents have at most two utility values for the chores. Nevertheless, the existence of an EF1 allocation in the non-monotone, non-additive setting is still open even for identical utility functions. For a brief summary of results on EF1 allocations, see Table 1.

EFX allocations Caragiannis et al. [22] introduced envy-freeness up to any good (EFX) as a strictly stronger fairness criterion than EF1, and provided a connection to pairwise maximum share guarantee for additive utilities over indivisible goods. Plaut and Roughgarden [40] investigated a stronger variant of EFX, referred to as EFX_0 in [35], for monotone utilities in the setting of goods.[5] They proved that an EFX_0 solution always exists when the utility functions are identical. This result yields a protocol that produces an EFX_0 allocation of goods for two players with general and possibly distinct utilities. Barman and Krishnamurthy [11] extended their result by proving the existence of EFX_0 allocations for additive utility functions when all agents have the same preference toward goods. Chaudhury et al. [23] verified the existence of an EFX_0 solution for three agents with monotone, additive, non-identical utilities. For monotone submodular utilities with binary marginal gains, Benabbou et al. [13] noted that EF1 implies EFX, thus their results guarantee the existence of EFX allocations (that are Pareto optimal as well) for any such instance. However, they showed that even an EF1 and utilitarian optimal allocation may violate the EFX_0 condition. Amanatidis et al. [5] studied 2-value instances, that is, when there are at most two possible values for the goods, and proved that any allocation that maximizes the Nash welfare is EFX_0. They proposed an algorithm called *Match & Freeze* for finding an EFX_0 solution that is based on repeatedly computing maximum matchings and freezing certain agents. Furthermore, they gave an algorithm for instances where the utilities of each agent take values in an interval such that the ratio between the maximum and the minimum value is at most 2. Mahara [39] verified the existence of EFX allocations when each agent has one of two additive utilities. Berger et al. [14] extended the result of [23] for the case of nice-cancelable utilities. The case of three agents with two general monotone utilities and one MMS-feasible utility, which generalizes nice-cancelable utilities, was settled by Akrami et al. [1].

In general, most of the results on EFX_0 allocations of goods cannot be straightforwardly translated into analogous results on allocations of chores. Therefore only a few results are known for the setting of mixed items when both goods and chores are present. A utility function is tertiary if $u(s) \in \{-1, 0, +1\}$ for every $s \in S$. It is not difficult to see that for tertiary utilities EF1 implies EFX, thus the *Double Round Robin Algorithm* of [7] provides such a solution. For the same setting, a different algorithm that provides a solution

[3] A pseudo-polynomial time algorithm has running time polynomial in the numeric value but not necessarily in the length of the input.

[4] Double monotonic utilities are also referred to as separable or itemwise monotone in the literature.

[5] EFX_0 differs from EFX in requiring that any envy should disappear even when leaving out an item with utility 0 from the envied agent's bundle. In other words, agent i cannot envy agent j if there exists $s \in \pi(j)$ with $u_i(s) = 0$.

Table 2
Landscape of results on different variants of EFX allocations, where bold letters represent the results obtained in the present paper. A utility function u is called *Boolean* if $u(X) \in \{0,1\}$ for every $X \subseteq S$, and *negative Boolean* if $-u$ is Boolean.

Utilities	Additive		Non-additive	
	Monotone	Non-monotone	Monotone	Non-monotone
Identical	Exists [40] (goods) Exists [30] (chores)	Exists [30,33]	Exists [40] (goods) **Exists (Theorem 4) (chores)**	**Not exists (Theorem 2)** **Boolean (Theorem 7)** **Neg. Boolean (Theorem 9)**
Non-identical	Open $n = 3$ [23] (goods) Tertiary [7]	Open $n = 2$ [30,33] Tertiary [7]	Open $n = 2$ [40] (goods) $(0,1)$-SUB [13] **$n = 2$ (Corollary 6) (chores)**	**Not exists (Theorem 2)** **Boolean (Theorem 7)**

which is Pareto optimal as well was given by Alexandrov and Walsh [2]. They also showed that an EFX allocation can be found in polynomial time for identical additive utilities. However, these results seem to be difficult to extend to the EFX_0 case. Chen and Liu [25] analyzed the fairness of leximin solutions in allocation of indivisible chores. Their model contained itemwise monotone utility functions, a concept which is equivalent to the double monotonic setting discussed earlier. They verified that a leximin solution is EFX for combinations of goods and chores for agents with identical itemwise monotone utilities. However, their notion of EFX solutions is slightly different from ours.[6]

Gourvès et al. [33] and Freeman et al. [29] studied *equitable* (EQ) allocations[7] and their relaxations, *equitable up to one good* (EQ1) and *equitable up to any good* (EQX). Freeman et al. [30] introduced analogous notions of (approximate) equitability of chores. By combining the results of [33] and [30], an algorithm for finding an EQX allocation of mixed items for additive utilities is at hand. When agents have identical utilities, an allocation satisfies EFX if and only if it satisfies EQX, thus this immediately implies the existence of an EFX allocation of mixed items for additive identical utilities. By the protocol in [40], this result guarantees EFX allocations of mixed items for two agents with additive (not necessarily identical) utilities. Li et al. [36] showed that EFX allocations always exist when the agents' utilities share an identical ordering of chores. Gafni et al. [31] proved an analogous result for leveled utilities, that is, when a set of chores with a higher cardinality is always more burdensome than a set with a lower cardinality. Yin and Mehta [44] studied the existence of EFX allocation of chores for three agents when two of them have additive utilities with further restrictions, while the third agent has a general monotone utility function.

Despite all the efforts made, the existence of an EFX/EFX_0 allocation remains an intriguing open question for at least four agents with distinct, monotone, additive utility functions. For a summary of results on EFX allocations, see Table 2.

1.2. Our results

We study the existence or non-existence of EF1 and EFX allocations for non-monotone utility functions. Our first main result (Theorem 1) is a polynomial-time algorithm that finds an EF1 allocation for two agents with arbitrary (i.e., non-monotone, non-additive, non-identical) utility functions.

Prior work on EFX/EFX_0 allocations has mainly concentrated on the cases when all items are either goods or chores and accordingly, the notion of EFX solutions was not defined for the mixed setting. Such utility functions appear naturally in applications. As an example, consider a conference where items correspond to submissions, and agents correspond to possible reviewers. A reviewer might prefer to check certain papers over other papers, and possibly to avoid writing a report on some of them. Even more, the preferences of the reviewers can be completely different: while checking a submission could be a nightmare for one, another might be happy to read it.

It turns out that finding the right generalization for non-additive, non-monotone utility functions is not straightforward. In Section 2, we propose four possible extensions possessing different characteristics, called EFX_0^0, EFX_-^0, EFX_0^+, and EFX_-^+[8]; here EFX_0^0 implies both EFX_-^0 and EFX_0^+, and both EFX_-^0 and EFX_0^+ imply EFX_-^+. The definitions differ in whether items with a marginal value of 0 are taken into account or not, both on the side of the envious and the envied agents. Unfortunately, Theorem 1 does not generalize even to EFX_-^+ allocations. We show that this is so even when the utility functions are identical (Theorem 2).

In the setting of goods (all utilities are non-negative), it is known [40] that EFX_0 allocations always exist for monotone identical utility functions. Equitable allocations (EQ) and relaxations (EQ1 and EQX) of chores were studied in [30]. It was shown there that when all utilities are additive and identical then EFX_-^+ allocations always exist. We strengthen these results by showing that, in the setting of chores as well, EFX_0^+ allocations always exist for monotone identical utility functions (Theorem 4), and that such an allocation can be computed in pseudo-polynomial time. By using the cut-and-choose-based protocol of Plaut and Roughgarden [40],

[6] In [25], an allocation π is called EFX if for any pair i, j of agents such that i envies j, $u_i(\pi(i)) \geq u_i(\pi(j) - s)$ for all $s \in \pi(j) \cap G_i$ and $u_i(\pi(i)) \geq u_i(\pi(j) + s)$ for all $s \in \pi(i) \cap C_i$. It is worth mentioning that for additive utilities, this definition is closely related to our notion of EFX-ness.

[7] An allocation π is *equitable* if $u_i(\pi(i)) = u_j(\pi(j))$ for all agents i, j.

[8] We remark that EFX_-^+ and EFX_-^0, when restricted to monotone, additive utilities, correspond to the usual notion of EFX and EFX_0 solutions, respectively.

our result guarantees an EFX_0^+ allocation of chores for two agents with monotone non-identical utility functions (Corollary 6). Note that a monotone utility function u in the setting of chores is monotone non-increasing, i.e., $X \subseteq Y$ implies $u(X) \geq u(Y)$.

As a further step towards understanding the general case, we consider special subclasses of utility functions: *Boolean* and *negative Boolean utilities*. Boolean utilities are non-monotone, non-additive $\{0,1\}$-valued functions, while a function u is negative Boolean if $-u$ is Boolean. In other words, one can think of Boolean functions as yes/no functions for bundles of items, not only for single items. Such utility functions are particularly interesting as they model simple yes/no preferences for goods and chores, respectively. The two classes share a lot of similarities, but the proofs of the corresponding results are significantly different: while the Boolean case is an easy observation (Theorem 7), the negative Boolean case requires a careful approach that only works for the case of identical utility functions (Theorem 9).

1.3. Techniques: old and new

EF1 allocations In the setting of goods and monotone utilities, EF1 allocations always exist [37]. The algorithm in [37] assigns goods to bundles one-by-one, while maintaining the property that the partial allocation constructed so far is EF1. At a general step of the algorithm, if no unenvied agent exists, then there exists a cycle of envy among agents, and bundles are shifted along such cycles until no cycle of envy remains. Otherwise the next good is added to the bundle of an unenvied agent. The key observation is that although such a step might result in new envies, those can be eliminated by removing the good she just received, so the resulting partial allocation remains EF1.

In Section 3.1, we prove that EF1 allocations always exist for $n = 2$ with arbitrary utilities. Our approach resembles the standard *moving-knife procedure*. More specifically, we show that for any ordering $s_1, \dots, s_m$ of all the items in the entire set S, there is an EF1 allocation $\pi = \langle \pi(1), \pi(2) \rangle$ such that $\{\pi(1), \pi(2)\} = \{\{s_1, \dots, s_t\}, \{s_{t+1}, \dots, s_m\}\}$ for some $0 \leq t \leq m$. This is shown by a combinatorial argument on prefixes and suffixes of the ordered set $\langle s_1, \dots, s_m \rangle$. We show an example in Section 3.2 that an EFX allocation need not exist for 2 agents with non-monotone, non-additive utilities - even when the utility functions are identical.

EFX allocations As mentioned earlier, Plaut and Roughgarden [40] showed the existence of EFX_0 allocations for monotone (non-decreasing) utilities in the setting of goods and identical utility functions. They construct such an allocation using the so-called leximin++ solution that selects the allocation which maximizes the minimum utility, and subject to this, maximizes the size of the bundle with minimum utility, subject to this, maximizes the second minimum utility, subject to this, maximizes the size of the bundle with the second minimum bundle, and so on. Our proof of existence of EFX_0^+ allocations for monotone (non-increasing) utilities in the setting of chores is different from this and is similar to the method used in [24] where another proof of the above result from [40] was given - this method also leads to a pseudo-polynomial time algorithm to find an EFX_0^+ allocation.

Our construction resembles the one in [37] to show the existence of EF1 allocations. Interestingly, the construction in [37] never *breaks up* any bundle obtained in a partial allocation whereas the construction in [24] and ours may need to – this is because we need to maintain an EFX allocation which is more demanding than an EF1 allocation. Thus items allocated to agents in earlier rounds may go back to the pool of unallocated items in some later round. However we show a potential function (same as the one in [24]) that improves as our algorithm progresses – thus our algorithm always converges. This result is given in Section 4.

The solutions for Boolean and negative Boolean utilities are based on different approaches. While the Boolean case is an easy observation but the resulting algorithm is not efficient, the negative Boolean case is solved via a non-trivial improvement step. Our results for these special cases are given in Sections 5.1 and 5.2, respectively. We hope that the tools used for these special cases will prove useful in other situations as well. Finally, the main results and the most important open problems are summarized in Section 6.

1.4. Other related work

Let us further mention several results that are closely related to fair divisions of indivisible items. Aziz et al. [8] considered an assignment problem in which agents have ordinal preferences over objects and these objects are allocated to the agents in a fair manner. They introduced several proportionality and envy-freeness concepts for discrete assignments, and gave polynomial-time algorithms to check whether a fair assignment exists for several of these fairness notions. In [18], Brams et al. discussed a problem setting with two agents who have strict rankings over an even number of indivisible items. They proposed algorithms to find balanced allocations of these items that maximize the minimum rank of the items that the agents receive, and are envy-free and Pareto optimal, if such allocations exist. In [41], Plaut and Roughgarden provided a thorough description of the communication complexity of computing a fair allocation with indivisible goods for every combination of fairness notion, utility function class, and number of players.

Another line of research considered group envy-freeness instead of pairs of agents. Berliant et al. [15] generalized envy-freeness for equal-sized groups of agents. Conitzer et al. [26] introduced the concept of group fairness, which implies most existing notions of individual fairness. They further proposed two relaxations similar to EF1, and showed that certain local optima of the Nash welfare function satisfy both relaxations and can be computed in pseudo-polynomial time by local search. However, [26] assumed that all the items are goods, that is, items for which agents have positive utility. While the notion of group fairness and group envy-freeness can be extended to the case when chores are also present, the same does not hold for the relaxations. Aziz et al. [7] proposed fairness concepts that are suitable to handle the case of goods and for chores as well. In the same spirit, Aziz and Rey [10] defined several variants and relaxations of group fairness and group envy-freeness when both goods and chores are present.

2. Preliminaries

2.1. Basic notation

Throughout the paper, N denotes a set of n *agents* and S denotes a set of m indivisible *items*. For simplicity, we will denote a subset of items by simply enumerating its elements without separating them, e.g. 124 stands for the set $\{1,2,4\}$.

Each agent $i \in N$ has a *utility function* $u_i : 2^S \to \mathbb{R}$ that represents agent i's preferences over the subsets of items. We assume that the utility functions are given with evaluation oracles which, for any input set $X \subseteq S$, output $u_i(X)$. The complexity of an algorithm is then measured by the total number of oracle calls and conventional elementary steps. We always assume that the empty set has value 0, that is, $u_i(\emptyset) = 0$ for $i \in N$. We say that the utility functions are *identical* if $u_i(X) = u_j(X)$ for every $X \subseteq S$ and $i, j \in N$, and *non-identical* if this condition does not necessarily hold. A utility function u is called *additive* if $u(X) = \sum_{s \in X} u(s)$ for every subset $X \subseteq S$, and it is *monotonely non-decreasing* if $u(X) \leq u(Y)$ whenever $X \subseteq Y$; similarly, it is *monotonely non-increasing* if $u(X) \geq u(Y)$ whenever $X \subseteq Y$. For simplicity, we will use *non-additive* and *non-monotone* as a shorthand for *not necessarily additive* and *not necessarily monotone*, respectively.

The *marginal utility* of an item $s \in S \setminus X$ towards a subset $X \subseteq S$ is denoted by $u(s|X)$ and is defined as $u(s|X) = u(s+X) - u(X)$. For a set $X \subseteq S$ and agent i, we denote by $S_i^+(X) = \{s \in X \mid u_i(s|X-s) > 0\}$, $S_i^-(X) = \{s \in X \mid u_i(s|X-s) < 0\}$, and $S_i^0(X) = \{s \in X \mid u_i(s|X-s) = 0\}$ the sets of items in X whose deletion decreases, increases, and does not change the utility of agent i on set X, respectively. We omit the index i when the utility functions are identical.

An *allocation* of S is a function $\pi : N \to 2^S$ assigning to each agent a (possibly empty) subset of items that altogether give a partition of S, that is, $\pi(i) \cap \pi(j) = \emptyset$ for distinct $i, j \in N$ and $\bigcup_{i \in N} \pi(i) = S$. It is worth emphasizing that all items must be allocated. We will refer to the set $\pi(i)$ as the *bundle of agent* i. Agent i *envies* agent j if $u_i(\pi(i)) < u_i(\pi(j))$.

2.2. Fairness in the non-monotone, non-additive setting

There are several ways to characterize fairness, probably the most natural one being *envy-freeness*, that requires that no agent envies another agent:

(EF) For any $i, j \in N$ inequality $u_i(\pi(i)) \geq u_i(\pi(j))$ holds.

Budish [19] provided a relaxation of envy-freeness by introducing the concept of *envy-freeness up to one good* in the context of monotone allocations. Aziz et al. [7] extended the definition to the non-monotone case by requiring that an agent's envy can be eliminated by removing some item either from her own bundle or the envied one:

(EF1) For any $i, j \in N$ at least one of the following holds:

(i) $u_i(\pi(i)) \geq u_i(\pi(j))$
(ii) $u_i(\pi(i) - s) \geq u_i(\pi(j) - s)$ for some $s \in \pi(i) \cup \pi(j)$.

Note that EF1 is strictly weaker than EF.

As a less permissive relaxation of envy-freeness, Caragiannis et al. [22] introduced the notion of *envy-freeness up to any good* in the context of goods and monotone utilities. According to their definition, an allocation is EFX if for any pair i, j of agents, agent i may envy agent j, however this envy would vanish upon removing any good from j's bundle which is positively valued by agent i:

(EFX) For any $i, j \in N$ at least one of the following holds:

(i) $u_i(\pi(i)) \geq u_i(\pi(j))$
(ii) $u_i(\pi(i)) \geq u_i(\pi(j) - s)$ for all $s \in S_i^+(\pi(j))$.

Plaut and Roughgarden [40] introduced a stronger variant called EFX_0, where the envy should vanish upon removing any good from $S_i^+(\pi(j)) \cup S_i^0(\pi(j))$. For non-monotone, additive utility functions with goods and chores Aziz et al. [7] proposed to call an allocation EFX if for any pair i, j of agents, $u_i(\pi(i) - s) \geq u_i(\pi(j) - s)$ for any $s \in S_i^+(\pi(j)) \cup S_i^-(\pi(i))$. Note that the latter definition does not require anything to hold for items $s \in S$ with 0 marginal utility value.

Extending the definition to non-monotone, non-additive utilities is not immediate. We introduce four variants called EFX_0^0, EFX_-^0, EFX_0^+, and EFX_-^+. Here EFX_0^0 implies both EFX_-^0 and EFX_0^+, and both EFX_-^0 and EFX_0^+ imply EFX_-^+. That is, EFX_-^0 and EFX_0^+ are independent in some sense.

(EFX_0^0) For any $i, j \in N$ at least one of the following holds:

(i) $u_i(\pi(i)) \geq u_i(\pi(j))$
(ii) $u_i(\pi(i)) \geq u_i(\pi(j) - s)$ for every $s \in S_i^+(\pi(j)) \cup S_i^0(\pi(j))$,
$u_i(\pi(i) - s) \geq u_i(\pi(j))$ for every $s \in S_i^-(\pi(i)) \cup S_i^0(\pi(i))$, and $S_i^+(\pi(j)) \cup S_i^0(\pi(j)) \cup S_i^-(\pi(i)) \cup S_i^0(\pi(i)) \neq \emptyset$.

It is not difficult to see that EFX_0^0 is strictly weaker than EF, but it is strictly stronger than EF1. However, EFX_0^0 is too much to ask for: with such a definition, the simple example with two agents and two goods with identical additive utilities $u(1) = 1$ and $u(2) = 0$ has no EFX_0^0 allocation. Thus we introduce two, slightly weaker variants.

(EFX_-^0) For any $i, j \in N$ at least one of the following holds:

(i) $u_i(\pi(i)) \geq u_i(\pi(j))$
(ii) $u_i(\pi(i)) \geq u_i(\pi(j) - s)$ for every $s \in S_i^+(\pi(j)) \cup S_i^0(\pi(j))$, $u_i(\pi(i) - s) \geq u_i(\pi(j))$ for every $s \in S_i^-(\pi(i))$, and $S_i^+(\pi(j)) \cup S_i^0(\pi(j)) \cup S_i^-(\pi(i)) \neq \emptyset$.

(EFX_0^+) For any $i, j \in N$ at least one of the following holds:

(i) $u_i(\pi(i)) \geq u_i(\pi(j))$
(ii) $u_i(\pi(i)) \geq u_i(\pi(j) - s)$ for every $s \in S_i^+(\pi(j))$, $u_i(\pi(i) - s) \geq u_i(\pi(j))$ for every $s \in S_i^-(\pi(i)) \cup S_i^0(\pi(i))$, and $S_i^+(\pi(j)) \cup S_i^-(\pi(i)) \cup S_i^0(\pi(i)) \neq \emptyset$.

EFX_-^0 and EFX_0^+ are symmetric, therefore these definitions represent in a certain sense dual concepts. Nevertheless, we will see that results for one of them do not automatically carry over to the other. Note that EFX_0^0 implies both EFX_-^0 and EFX_0^+.

Finally, let us introduce a further weaker condition that is the easiest to work with.

(EFX_-^+) For any $i, j \in N$ at least one of the following holds:

(i) $u_i(\pi(i)) \geq u_i(\pi(j))$
(ii) $u_i(\pi(i)) \geq u_i(\pi(j) - s)$ for every $s \in S_i^+(\pi(j))$, $u_i(\pi(i) - s) \geq u_i(\pi(j))$ for every $s \in S_i^-(\pi(i))$, and $S_i^+(\pi(j)) \cup S_i^-(\pi(i)) \neq \emptyset$.

Clearly, EFX_-^+ is implied by both EFX_-^0 and EFX_0^+. For additive, monotone utility functions, EFX introduced in [22] and EFX_0 introduced in [40] are identical to EFX_-^+ and EFX_-^0, respectively. We further note that if $s \in S_i^0(X)$ then $u_i(X - s) = u_i(X)$. Therefore agent i cannot envy agent j in an EFX_0^0 or EFX_-^0 allocation if $S_i^0(\pi(j)) \neq \emptyset$, and similarly, agent i cannot envy agent j in an EFX_0^0 or EFX_0^+ allocation if $S_i^0(\pi(i)) \neq \emptyset$. In this sense, parts (i) and (ii) of definitions EFX_0^0, EFX_-^0, and EFX_0^+ describe non-disjoint situations.

In any of the above cases, if agent i envies agent j with respect to allocation π, i.e., $u_i(\pi(i)) < u_i(\pi(j))$, then we refer to this envy as an *EFX envy* if i and j satisfy condition (ii) of the corresponding definition, otherwise it is called a *non-EFX envy*.

2.3. Relation to EFX and EFX_0

Although the notion of EFX_0^0 allocations seems to be reasonable, such an allocation does not necessarily exist even in very simple examples. EFX_-^0 allocations provide a natural extension of the EFX_0 property, and EFX_0^+ serves as a symmetric counterpart. Finally, EFX_-^+ provides a generalization of EFX allocations.

Let us explain why the non-emptiness condition in the (ii) part of the definitions is necessary. Consider an example with two agents and two items with identical utilities, namely $u(\emptyset) = 0$, $u(1) = u(2) = 2$ and $u(12) = 1$. Then for the allocation $\pi(1) = \emptyset$ and $\pi(2) = 12$, agent 1 envies agent 2. Then all of the sets $S_1^-(\pi(1))$, $S_1^0(\pi(1))$, $S_1^0(\pi(2))$, and $S_1^+(\pi(2))$ are empty, thus condition (ii) without the non-emptiness assumption is a tautology in all cases. Nevertheless, the allocation does not seem to be fair towards agent 1 in any way.

The peculiar conditions for the non-emptiness of the corresponding sets have not arisen in previous works for two reasons. On the one hand, earlier results mainly focused on monotone (non decreasing) utilities. But most importantly, they mostly considered additive utilities, and in such cases the non-emptiness conditions are redundant. Indeed, if $u_i(\pi(i)) < u_i(\pi(j))$ and u_i is additive, then at least one of the sets $S_i^-(\pi(i))$ and $S_i^+(\pi(j))$ is non-empty.

3. Non-monotone and non-additive utilities

In this section we consider arbitrary utilities. Somewhat surprisingly, EF1 allocations always exist for two agents and can be found in polynomial time even in this general setting; our algorithm is presented in Section 3.1. The existence of an EFX allocation with arbitrary but identical utilities is discussed in Section 3.2.

3.1. *EF1* allocations

So far, the existence of an EF1 allocation for general (i.e., non-monotone, non-additive) utility functions is still open, even for identical utilities. Our first result answers this question positively for the special case of two agents – the case of three of more agents remains an intriguing open problem.

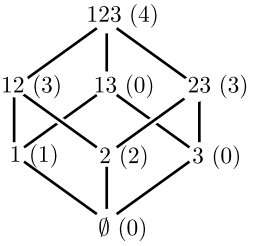

Fig. 1. Illustration of Theorem 2. The lattice of subsets of $S = \{1,2,3\}$, together with the corresponding utility value (in parentheses).

Theorem 1. *There always exists an EF1 allocation for two agents with arbitrary (not necessarily monotone or additive) utility functions. Such an allocation can be computed in linear time, assuming that both utility functions are given by an evaluation oracle.*

Proof. For an arbitrary ordering $1,\dots,m$ of the items, let $F_i = \{1 \dots i\}$ and $L_i = \{i+1 \dots m\}$ denote the sets of the first i and last $m-i$ items, respectively, for $i = 0,\dots,m$, where $F_0 = L_m = \emptyset$.

Assume that $u_i(F_j) = u_i(L_j)$ for some $i \in \{1,2\}$ and $j \in \{0,\dots,m\}$. If $u_{3-i}(F_j) \le u_{3-i}(L_j)$, then let $\pi(i) = F_j$ and $\pi(3-i) = L_j$, else let $\pi(i) = L_j$ and $\pi(3-i) = F_j$. By the assumption $u_i(F_j) = u_i(L_j)$, π is an envy-free allocation. Similarly, if there exists an index $j \in \{0,\dots,m\}$ such that $u_i(F_j) \ge u_i(L_j)$ and $u_{3-i}(F_j) \le u_{3-i}(L_j)$ for some $i \in \{1,2\}$, then $\pi(i) = F_j$, $\pi(3-i) = L_j$ is an envy-free allocation again.

From now on we assume that neither of the above two cases holds. Thus $u_1(S) \neq 0$ where S is the entire set of m items. We distinguish two cases.

Case 1. $u_1(S) > 0$.

As $u_1(F_0) = u_1(L_m) = u_1(\emptyset) = 0$ and $u_1(F_m) = u_1(L_0) = u_1(S) > 0$, there exists an index $j \in \{0,\dots,m-1\}$ such that $u_1(F_j) < u_1(L_j)$ and $u_1(F_{j+1}) > u_1(L_{j+1})$. By our assumption, $u_2(F_j) < u_2(L_j)$ and $u_2(F_{j+1}) > u_2(L_{j+1})$ also hold.

Subcase 1.1 $u_1(F_j) \le u_1(L_{j+1})$: set $\pi(1) = L_{j+1}$ and $\pi(2) = F_{j+1}$.

Subcase 1.2 $u_1(F_j) > u_1(L_{j+1})$: set $\pi(1) = F_j$ and $\pi(2) = L_j$

In both subcase 1.1 and subcase 1.2, agent 1 envies agent 2's bundle, but this envy can be eliminated by deleting the $(j+1)$-th item from agent 2's bundle. This is because $u_1(L_{j+1}) \ge u_1(F_j)$ in subcase 1.1 and $u_1(F_j) > u_1(L_{j+1})$ in subcase 1.2.

Case 2. $u_1(S) < 0$.

As $u_1(F_0) = u_1(L_m) = u_1(\emptyset) = 0$ and $u_1(L_0) = u_1(F_m) = u_1(S) < 0$, there exists an index $j \in \{0,\dots,m-1\}$ such that $u_1(F_j) > u_1(L_j)$ and $u_1(F_{j+1}) < u_1(L_{j+1})$. By our assumption, $u_2(F_j) > u_2(L_j)$ and $u_2(F_{j+1}) < u_2(L_{j+1})$ also hold.

Subcase 2.1 $u_1(F_j) \ge u_1(L_{j+1})$: set $\pi(1) = F_{j+1}$ and $\pi(2) = L_{j+1}$.

Subcase 2.2 $u_1(F_j) < u_1(L_{j+1})$: set $\pi(1) = L_j$ and $\pi(2) = F_j$.

In both subcase 2.1 and subcase 2.2, agent 1 envies agent 2's bundle, but this envy can be eliminated by deleting the $(j+1)$-th item from agent 1's bundle. This is because $u_1(F_j) \ge u_1(L_{j+1})$ in subcase 2.1 and $u_1(L_{j+1}) > u_1(F_j)$ in subcase 2.2.

This concludes the proof of the theorem. □

3.2. An interesting example

The difference between EF1 and EFX allocations is well illustrated by the fact that, in contrast to the EF1 case, an EFX^+_- allocation might not exist even for two agents with identical utilities.

Theorem 2. *There need not exist an EFX^+_- allocation for two agents with non-monotone, non-additive, identical utility functions.*

Proof. Let $S = \{1,2,3\}$ be a set of three items and let $u : 2^S \to \mathbb{R}$ be defined by $u(\emptyset) = 0$, $u(1) = 1$, $u(2) = 2$, $u(3) = 0$, $u(12) = 3$, $u(13) = 0$, $u(23) = 3$, and $u(123) = 4$ (see Fig. 1). We claim that there is no EFX^+_- allocation for two agents.

As the utility functions are identical, we may assume that the first agent receives more items. If all the items are allocated to agent 1, then $u_1(\pi(1)) = u(123) = 4$ and $u_2(\pi(2)) = u(\emptyset) = 0$, hence agent 2 envies agent 1. However, this is a non-EFX envy as $1,3 \in S^+(123)$, but deleting either item 1 or 3 from 123 results in a set with positive utility value.

If $\pi(1) = 12$ and $\pi(2) = 3$, then $u_1(\pi(1)) = u(12) = 3$ and $u_2(\pi(2)) = u(3) = 0$, hence agent 2 envies agent 1. This is a non-EFX envy as $1 \in S^+(12)$, but $2 = u(2) > u(3) = 0$. If $\pi(1) = 23$ and $\pi(2) = 1$, then $u_1(\pi(1)) = u(23) = 3$ and $u_2(\pi(2)) = u(1) = 1$, hence agent 2 envies agent 1. This is a non-EFX envy as $3 \in S^+(23)$, but $2 = u(2) > u(1) = 1$. Finally, if $\pi(1) = 13$ and $\pi(2) = 2$, then $u_1(\pi(1)) = u(13) = 0$ and $u_2(\pi(2)) = u(2) = 2$, hence agent 1 envies agent 2. This is a non-EFX envy as $3 \in S^-(13)$, but $1 = u(1) < u(2) = 2$. □

Remark 3. Although the utility function defined in the proof of Theorem 2 is non-monotone, it is as close to being monotone as possible in the sense that it can be made monotone by increasing its value on a single set (namely 13) by one. Note that an EF1 allocation exists: just set $\pi(1) = 13$ and $\pi(2) = 2$, then the envy of agent 1 can be eliminated by deleting 2 from $\pi(2)$.

Algorithm 1 Finding an EFX_0^+ allocation of chores for monotone, identical utility functions.

Input: Set N of agents, set S of items, monotone utility function $u : 2^S \to \mathbb{R}_-$.
Output: An EFX_0^+ allocation.
Let $P := S$.
Set $\pi(i) := \emptyset$ for all $i \in N$.
while P is not empty **do**
  Pick an unallocated chore $s \in P$.
  Pick a happiest agent k.
  if $(\pi(1), \dots, \pi(k) + s, \dots, \pi(n))$ is EFX_0^+ **then**
    $\pi(k) \leftarrow \pi(k) + s$
    $P \leftarrow P - s$
  else
    Let $Z \subset \pi(k) + s$ be inclusionwise minimal s.t. $u(Z) < u(\pi(\ell))$ for some $\ell \in N$.
    $\pi(k) \leftarrow Z$
    $P \leftarrow P \cup (\pi(k) + s) \setminus Z$
return π.

4. EFX_0^+ allocations of chores with identical utilities

Envy-freeness and its relaxations have been extensively studied in the setting of indivisible goods. A closely related problem is the fair division of indivisible *chores*. In this section, by 'chores' we mean monotone non-increasing utility functions, i.e., $X \subseteq Y$ implies $u_i(X) \geq u_i(Y)$.

Our goal is to find an EFX_0^+ allocation π of S. Note that for monotone non-increasing utilities, this requires that

(⋆) for any pair of agents i, j, $u_i(\pi(i) - s) \geq u_i(\pi(j))$ for all $s \in \pi(i)$.

We now show that such an allocation of chores always exists when (i) $n = 2$ or (ii) all n agents have identical utilities. First we show the following result.

Theorem 4. *When all agents have monotone (not necessarily additive) identical utility functions, an EFX_0^+ allocation of chores always exists. Such an allocation can be computed with pseudo-polynomial number of utility queries.*

Proof. We present Algorithm 1 that constructs an EFX_0^+ allocation of the set S of chores. The joint utility function of the agents is denoted by u. The algorithm runs in rounds and maintains a pool P of unallocated chores and an allocation that is EFX_0^+ on $S \setminus P$. Initially, we set $P = S$. Interestingly, chores once allocated to some agent may go back to the pool P in a later round.

Consider any round and let $\pi = \langle \pi(1), \dots, \pi(n) \rangle$ be the EFX_0^+ allocation at the start of this round and let $\pi' = \langle \pi'(1), \dots, \pi'(n) \rangle$ be the EFX_0^+ allocation at the end of this round. Let P (resp., P') be the set of unallocated chores at the start (resp., end) of this round. We will ensure that at least one of the following two conditions is satisfied:

(i) $|P'| < |P|$ and $\sum_{i \in N} u(\pi'(i)) \leq \sum_{i \in N} u(\pi(i))$, i.e., the number of allocated chores increases in this round and *utilitarian welfare* (sum of utilities of all agents) does not increase,
(ii) $\sum_{i \in N} u(\pi'(i)) < \sum_{i \in N} u(\pi(i))$, i.e., utilitarian welfare strictly decreases in this round.

In other words, in each round of our algorithm, we either decrease utilitarian welfare or we increase the number of allocated chores without increasing utilitarian welfare. This will allow us to show that our algorithm always terminates. We now describe a single round in detail.

Since all agents have the same utilities, the agents can be ordered in terms of the utilities of their bundles. So at any point in time, there is at least one agent who can be called a *happiest* agent: one who does not envy any other agent. In each round, we pick any unallocated chore s and add s to the bundle of a happiest agent (call this agent k). There are two cases:

Case 1. Suppose $\langle \pi(1), \dots, \pi(k) + s, \dots, \pi(n) \rangle$ is EFX_0^+. Then set $\pi'(k) = \pi(k) + s$ and $\pi'(i) = \pi(i)$ for $i \neq k$. Observe that the allocation $\pi' = \langle \pi'(1), \dots, \pi'(n) \rangle$ satisfies condition (i) given above.

Case 2. Suppose $\langle \pi(1), \dots, \pi(k) + s, \dots, \pi(n) \rangle$ is not EFX_0^+. By observation **(⋆)**, $u(\pi(k) + s - s') < u(\pi(\ell))$ for some $s' \in \pi(k) + s$ and $\ell \in N$. Thus we can find an inclusionwise minimal subset $\pi'(k) \subset \pi(k) + s$ such that $\pi'(k)$ is an envious bundle, i.e., $u(\pi'(k)) < u(\pi(\ell))$ for some $\ell \in N$. Since $\pi'(k)$ is a minimal envious subset of $\pi(k) + s$, for every $X \subset \pi'(k)$, we have $u(X) \geq u(\pi(i))$ for all $i \in N$.

Now the chores in $(\pi(k) + s) \setminus \pi'(k)$ are thrown back into the pool P, i.e., $P' = P \cup (\pi(k) + s) \setminus \pi'(k)$. Let $\pi'(i) = \pi(i)$ for all $i \neq k$.

Claim 5. *The allocation $\pi' = \langle \pi'(1), \dots, \pi'(n) \rangle$ is an EFX_0^+ allocation.*

Proof. We need to show that $u(\pi'(i) - r) \geq u(\pi'(j))$ for all $i, j \in N$ and $r \in \pi'(i)$. Since $\pi'(t) = \pi(t)$ for all $t \neq k$ and π is EFX_0^+, for all i, j such that neither i nor j is k, this holds.

When $i = k$: Since $\pi'(k)$ is a minimal envious subset of $\pi(k) + s$, we have $u(\pi'(k) - r) \geq u(\pi(j)) = u(\pi'(j))$ for all $r \in \pi'(k)$ and $j \neq k$.

When $j = k$: We claim that $u(\pi'(k)) < u(\pi(k))$. Indeed, $u(\pi'(k)) < u(\pi(\ell))$ for some $\ell \in [n]$ since $\pi'(k)$ is envious, and $u(\pi(\ell)) \leq u(\pi(k))$ for all $\ell \in [n]$ since k is a happiest agent in π. Thus for any $i \neq k$, we have:

$$u(\pi'(i) - r) = u(\pi(i) - r) \geq u(\pi(k)) > u(\pi'(k)) \ \forall r \in \pi'(i),$$

where the inequality $u(\pi(i) - r) \geq u(\pi(k))$ follows from π being EFX_0^+. □

Since $u(\pi'(k)) < u(\pi(k))$ and $\pi'(t) = \pi(t)$ for all $t \neq k$, $\sum_{i\in[n]} u(\pi'(i)) < \sum_{i\in[n]} u(\pi(i)))$. Thus the allocation $\pi' = \langle \pi'(1), \ldots, \pi'(n)\rangle$ satisfies condition (ii) given above.

Note that there can be at most m consecutive rounds where condition (i) holds. After that, either all chores are allocated or there is a round where condition (ii) holds, i.e., utilitarian welfare strictly decreases and this cannot reduce beyond $-n \cdot u(S)$. Let $\Delta = \min\{|u(X) - u(Y)| \mid X, Y \subseteq S \text{ and } u(X) \neq u(Y)\}$ be the minimum difference between distinct utilities. After $mn|u(S)|/\Delta$ rounds, the utilitarian welfare has to be $-n \cdot u(S)$ and within the subsequent m rounds, all chores have to get allocated. Hence our algorithm always terminates. Thus an EFX_0^+ allocation of chores always exists when all agents have monotone identical utility functions.

Let us now bound the number of utility queries made in each round. Let ℓ be the *second happiest* agent with respect to the allocation π, i.e., $u(\pi(\ell))$ is the largest in $\{u(\pi(i)) \mid i \in N \setminus \{k\}\}$. In order to test if $(\pi(1), \ldots, \pi(k) + s, \ldots, \pi(n))$ is EFX_0^+ or not, we need to check if $u(\pi(k) + s - s') \geq u(\pi(\ell))$ for all $s' \in \pi(k)$. We do this as follows: start with $Z = \pi(k) + s$. For every $s' \in Z$ do:

- if $u(Z - s') \geq u(\pi(\ell))$ then Z remains the same; else update $Z = Z - s'$.

Thus whenever we find a chore $s' \in Z$ that satisfies $u(Z - s') < u(\pi(\ell))$, we update $Z = Z - s'$. If Z gets updated, then the monotonicity of u implies that for every s' in the final set Z, we have $u(Z - s') \geq u(\pi(\ell))$.

Since we query exactly once for each $s' \in \pi(k) + s$ to check if $u(Z - s') \geq u(\pi(\ell))$ or not, the number of utility queries made in each round is at most m. So either $(\pi(1), \ldots, \pi(k) + s, \ldots, \pi(n))$ is EFX_0^+ or we find an inclusionwise minimal subset Z of $\pi(k) + s$ such that $u(Z) < u(\pi(\ell))$. The total number of utility queries made during the entire algorithm is $O(m^2 n|u(S)|/\Delta)$. Assuming there is an oracle to answer utility queries in unit time, our algorithm runs in pseudo-polynomial time. □

The cut-and-choose protocol of Plaut and Roughgarden [40] implies the following corollary.

Corollary 6. *There always exists an EFX_0^+ allocation of chores for two agents with monotone (not necessarily additive) utility functions.*

Proof. Let u_1 and u_2 be the utility functions of agent 1 and agent 2, respectively. By Theorem 4, there exists an EFX_0^+ allocation when the utility function of both agents is u_1. Take such an allocation; this defines a partition of S into two parts S_1 and S_2. Now let the second agent choose among S_1 and S_2 based on her preferences. Clearly, agent 2 will have no envy as she chooses the set with better utility with respect to u_2. On the other hand, even if agent 1 envies agent 2, this is an EFX_0^+ envy due to the construction of the allocation. □

5. Boolean and negative Boolean utility functions

In this section we consider special cases of non-identical utility functions: Boolean and negative Boolean utilities.

5.1. Boolean utilities

Recall that a utility function u is called Boolean if $u(X) \in \{0, 1\}$ for $X \subseteq S$.

Theorem 7. *When all agents have Boolean utility functions, EFX_-^0 allocations always exist.*

Proof. We construct an EFX_-^0 solution by assigning bundles to agents one-by-one using Algorithm 2. After each step, we will refer to unassigned items and to agents without bundles as *remaining* items and agents, respectively.

At a general step of the algorithm, we take an inclusionwise minimal subset X of the remaining items that has utility value 1 for at least one of the remaining agents, and we assign X to one of the agents i with $u_i(X) = 1$. If no such set exists, then we pick an arbitrary agent i and set X to be the empty set. Then we delete the members of X from S and i from N. When only one agent remains, we assign the remaining set of items to her.

We claim that the solution thus obtained is an EFX_-^0 allocation. We may assume that the order of the agents in which their bundles get fixed is $1, \ldots, n$. Note that the sequence $u_1(\pi(1)), \ldots, u_n(\pi(n))$ is monotone decreasing, that is, it consists of a sequence of 1's followed by a sequence of 0's (any of these two parts can be empty). Therefore if agent i envies j, then $j < i$, $u_i(\pi(j)) = 1$, and $u_i(\pi(i)) = 0$. However, by the choice of $\pi(j)$ in Step 3 of the algorithm, $u_i(\pi(j) - s) = 0$ for every $s \in \pi(j)$. Moreover, $S_i^+(\pi(j)) \neq \emptyset$ and the statement follows. □

Algorithm 2 Finding an EFX_{-}^{0} allocation for Boolean utility functions.

Input: Set N of agents, set S of items, utility functions $u_i : 2^S \to \{0,+1\}$.
Output: An EFX_{-}^{0} allocation.
1: **while** $|N| \geq 2$ **do**
2: **if** there exist $i \in N$ and $X \subseteq S$ s.t. $u_i(X) = 1$ **then**
3: Let $X \subseteq S$ be inclusionwise minimal s.t. $u_i(X) = 1$ for some $i \in N$. Let i be an agent with $u_i(X) = 1$. Set $\pi(i) := X$.
4: $N \leftarrow N - i$
5: $S \leftarrow S - X$
6: **else**
7: Arbitrarily pick $N' \subseteq N$ with $|N'| = |N| - 1$ and set $\pi(i) := \emptyset$ for each $i \in N'$.
8: $N \leftarrow N \setminus N'$
9: For the single remaining agent j in N, set $\pi(j) := S$.
10: **return** π

Note that an EFX_{0}^{+} allocation does not necessarily exist for Boolean allocations. To see this, consider the instance with three items and two agents with identical utilities defined as follows: $u(X) = 0$ if $|X| \leq 1$ and $u(X) = 1$ otherwise. It is not difficult to verify that no allocation satisfies the conditions of EFX_{0}^{+}.

Remark 8. The proof of Theorem 7 leads to a construction that is not efficient, as Step 3 of Algorithm 2 asks for an inclusionwise minimal set with utility value 1 for at least one of the remaining agents. The complexity of finding such a set depends on how the utility functions are given, but the difficulty is well illustrated by the fact that the problem is equivalent to determining a satisfying assignment of a Boolean function using a minimum number of true variables. Therefore the proposed algorithm does not work for finding an EFX_{-}^{0} allocation. Showing a lower bound on the complexity of finding an EFX_{-}^{0} allocation in this setting remains an interesting open problem.

One of the most common representations of Boolean functions are *conjunctive normal forms* (CNFs), the conjunctions of clauses which are elementary disjunctions of literals. A CNF is called *pure Horn* if every clause in it contains exactly one positive literal, and a Boolean function is pure Horn if it admits a pure Horn CNF representation. Pure Horn functions form a fundamental subclass of Boolean functions admitting interesting structural and computational properties. Among others, SAT is solvable for this class in linear time [27].

If each utility function u_i is represented by a pure Horn CNF, then a set X satisfying the conditions of Step 3 can be determined in polynomial time with the help of the so-called *forward chaining procedure*. For further details, we refer the interested reader to [34].

5.2. Negative Boolean utilities

Recall that a utility function u is called negative Boolean if $-u$ is Boolean. Somewhat surprisingly, Boolean and negative Boolean show a different behavior, though at first sight one would expect those to be identical up to translation by $+1$ or multiplication by -1. This breakdown of duality is caused by the fact that negative Boolean utilities represent chores, for which standard techniques (such as the *round-robin* or *envy-cycle elimination algorithms*) are no longer applicable; see the recent survey by Liu et al. [38] for further explanation.

Theorem 9. *When all agents have identical negative Boolean utility functions, EFX_{0}^{+} allocations always exist. Such an allocation can be computed in polynomial time, assuming that the utility function is given by an evaluation oracle.*

Proof. The algorithm is presented as Algorithm 3. The joint utility function of the agents is denoted by u. We start with an arbitrary allocation, say, $\pi(1) = S$ and $\pi(i) = \emptyset$ otherwise. In a general step of the algorithm, we pick a pair $i, j \in N$ of agents such that there is a non-EFX envy from i to j. As we are considering EFX_{0}^{+} allocations and the utility function is negative Boolean, this means that there exists an item $s \in \pi(i)$ such that $-1 = u(\pi(i)) = u(\pi(i) - s) < u(\pi(j)) = 0$. The algorithm moves such an item from the bundle of agent i to that of agent j.

It suffices to show that the algorithm terminates after a polynomial number of steps. Recall that the numbers of agents and items are denoted by n and m, respectively. Define

$$\varphi(\pi) = m \cdot \sum_{\substack{i \in N \\ u(\pi(i))=-1}} 1 + \sum_{\substack{i \in N \\ u(\pi(i))=0}} |\pi(i)|.$$

In other words, φ counts m times the number of agents with utility value -1, plus the total number of items in bundles having utility value 0.

Claim 10. *$\varphi(\pi)$ strictly increases throughout the algorithm.*

Proof. Let π' denote the allocation obtained from π by moving s from $\pi(i)$ to $\pi(j)$, that is, $\pi'(i) = \pi(i) - s$, $\pi'(j) = \pi(j) + s$, $\pi'(k) = \pi(k)$ otherwise. By $u(\pi(i)) = -1$, we have $\pi(i) \neq \emptyset$ and so $|\pi(j)| \leq m - 1$.

Algorithm 3 Finding an EFX_0^+ allocation for identical negative Boolean utility functions.

Input: Set N of agents, set S of items, utility function $u : 2^S \to \{0, -1\}$.
Output: An EFX_0^+ allocation.
1: Set $\pi(1) := S$ and $\pi(i) = \emptyset$ for all $i > 1$.
2: **while** π is not EFX_0^+ **do**
3: Let $i, j \in N$ s.t. $u(\pi(i)) < u(\pi(j))$ and the envy is non-EFX.
4: Choose $s \in \pi(i)$ such that $u(\pi(i) - s) < u(\pi(j))$.
5: $\pi(i) \leftarrow \pi(i) - s$
6: $\pi(j) \leftarrow \pi(j) + s$
7: **return** π

If $u(\pi'(j)) = -1$, then

$$\begin{aligned}
\varphi(\pi') &= m \cdot \sum_{\substack{i \in N \\ u(\pi'(i))=-1}} 1 + \sum_{\substack{i \in N \\ u(\pi'(i))=0}} |\pi'(i)| \\
&\geq m \cdot \left(\left(\sum_{\substack{i \in N \\ u(\pi(i))=-1}} 1 \right) + 1 \right) + \left(\sum_{\substack{i \in N \\ u(\pi(i))=0}} |\pi(i)| - (m-1) \right) \\
&= m \cdot \sum_{\substack{i \in N \\ u(\pi(i))=-1}} 1 + \sum_{\substack{i \in N \\ u(\pi(i))=0}} |\pi(i)| + 1 \\
&= \varphi(\pi) + 1,
\end{aligned}$$

while if $u(\pi'(j)) = 0$, then

$$\begin{aligned}
\varphi(\pi') &= m \cdot \sum_{\substack{i \in N \\ u(\pi'(i))=-1}} 1 + \sum_{\substack{i \in N \\ u(\pi'(i))=0}} |\pi'(i)| \\
&= m \cdot \sum_{\substack{i \in N \\ u(\pi(i))=-1}} 1 + \sum_{\substack{i \in N \\ u(\pi(i))=0}} |\pi(i)| + 1 \\
&= \varphi(\pi) + 1.
\end{aligned}$$

This concludes the proof of the claim. □

As φ takes integer values upper bounded by $m \cdot n$, Claim 10 implies that the algorithm terminates after a polynomial number of steps, and the theorem follows. □

Note that an EFX_-^0 allocation does not necessarily exist for negative Boolean allocations. To see this, consider the instance with three items and two agents with identical utilities defined as follows: $u(X) = 0$ if $|X| \leq 1$ and $u(X) = -1$ otherwise. It is not difficult to verify that no allocation satisfies the conditions of EFX_-^0.

Remark 11. A natural adaptation of Algorithm 2 to the negative Boolean setting would be to always choose an inclusionwise minimal subset of the remaining items in Step 3 that has -1 utility value. However, such an approach has no control over the set that is allocated to the last agent, and so a non-EFX envy might be present.

6. Conclusions

The present paper focused on the concept of envy-freeness and its relaxations, envy-freeness up to one item and envy-freeness up to any item. Concerning EF1 allocations, we presented a polynomial-time algorithm for finding one for two agents with arbitrary utility functions. We extended the notion of EFX allocations to non-monotone, non-additive utilities, and settled the existence or non-existence of such solutions in various settings. We showed that an EFX_0^+ allocation of chores always exists for monotone identical utility functions. For the classes of Boolean and identical negative Boolean utilities, we verified the existence of EFX_-^0 and EFX_0^+ allocations, respectively.

Tables 1 and 2 show that the existence or non-existence of a fair solution is still open in many cases. Among them, we would like to draw attention to the following open problems that seem to be particularly interesting.

Question 12. *Does there always exist an EF1 allocation for an arbitrary number of agents with non-monotone, non-additive, identical utility functions?*

Question 13. *Does there always exist an* EFX^{+}_{-} *allocation for an arbitrary number of agents with monotone, additive, non-identical utility functions?*

Question 14. *Does there always exist an* EFX^{+}_{0} *allocation for an arbitrary number of agents with monotone (not necessarily identical) utility functions?*

Question 15. *Does there always exist an* EFX^{0}_{-} *allocation for an arbitrary number of agents with monotone identical utility functions?*

Question 16. *Does there always exist an* EFX^{+}_{0} *allocation for an arbitrary number of agents with (not necessarily identical) negative Boolean utility functions?*

CRediT authorship contribution statement

Kristóf Bérczi: Methodology, Writing – original draft, Writing – review & editing. **Erika R. Bérczi-Kovács:** Methodology, Writing – original draft, Writing – review & editing. **Endre Boros:** Methodology, Writing – original draft, Writing – review & editing. **Fekadu Tolessa Gedefa:** Methodology, Writing – original draft, Writing – review & editing. **Naoyuki Kamiyama:** Methodology, Writing – original draft, Writing – review & editing. **Telikepalli Kavitha:** Methodology, Writing – original draft, Writing – review & editing. **Yusuke Kobayashi:** Methodology, Writing – original draft, Writing – review & editing. **Kazuhisa Makino:** Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Erika Bérczi-Kovács was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP-23-5 New National Excellence Program of the Ministry for Culture and Innovation. Naoyuki Kamiyama was supported by JST, PRESTO Grant Number JPMJPR1753, Japan. Yusuke Kobayashi was supported by JSPS, KAKENHI grant numbers JP18H05291, JP19H05485, and JP20K11692, Japan. Telikepalli Kavitha was supported by the Department of Atomic Energy, Government of India, under project no. RTI4001.

The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant no. LP2021-1/2021, by the Hungarian National Research, Development and Innovation Office – grant no. FK128673, by the Ministry of Innovation and Technology of Hungary – grant no. ELTE TKP 2021-NKTA-62, by Dynasnet European Research Council Synergy project – grant no. ERC-2018-SYG 810115, and by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

References

[1] H. Akrami, N. Alon, B.R. Chaudhury, J. Garg, K. Mehlhorn, R. Mehta, Efx allocations: simplifications and improvements, arXiv preprint, arXiv:2205.07638, 2022.
[2] M. Aleksandrov, T. Walsh, Greedy algorithms for fair division of mixed Manna, arXiv preprint, arXiv:1911.11005, 2019.
[3] M. Aleksandrov, T. Walsh, Online fair division: a survey, Proc. AAAI Conf. Artif. Intell. 34 (9) (2020) 13557–13562.
[4] G. Amanatidis, H. Aziz, G. Birmpas, A. Filos-Ratsikas, B. Li, H. Moulin, A.A. Voudouris, X. Wu, Fair division of indivisible goods: a survey, arXiv preprint, arXiv:2208.08782, 2022.
[5] G. Amanatidis, G. Birmpas, A. Filos-Ratsikas, A. Hollender, A.A. Voudouris, Maximum Nash welfare and other stories about EFX, Theor. Comput. Sci. 863 (2021) 69–85.
[6] H. Aziz, P. Biró, J. Lang, J. Lesca, J. Monnot, Optimal reallocation under additive and ordinal preferences, in: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, 2016, pp. 402–410.
[7] H. Aziz, I. Caragiannis, A. Igarashi, T. Walsh, Fair allocation of indivisible goods and chores, Auton. Agents Multi-Agent Syst. 36 (1) (2022) 1–21.
[8] H. Aziz, S. Gaspers, S. Mackenzie, T. Walsh, Fair assignment of indivisible objects under ordinal preferences, Artif. Intell. 227 (2015) 71–92.
[9] H. Aziz, B. Li, H. Moulin, X. Wu, Algorithmic fair allocation of indivisible items: a survey and new questions, ACM SIGecom Exch. 20 (1) (2022) 24–40.
[10] H. Aziz, S. Rey, Almost group envy-free allocation of indivisible goods and chores, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 39–45.
[11] S. Barman, S.K. Krishnamurthy, Approximation algorithms for maximin fair division, ACM Trans. Econ. Comput. 8 (1) (2020) 1–28.
[12] S. Barman, S.K. Krishnamurthy, R. Vaish, Finding fair and efficient allocations, in: Proceedings of the 2018 ACM Conference on Economics and Computation, 2018, pp. 557–574.
[13] N. Benabbou, M. Chakraborty, A. Igarashi, Y. Zick, Finding fair and efficient allocations when valuations don't add up, in: International Symposium on Algorithmic Game Theory, Springer, 2020, pp. 32–46.

[14] B. Berger, A. Cohen, M. Feldman, A. Fiat, Almost full EFX exists for four agents, Proc. AAAI Conf. Artif. Intell. 36 (2022) 4826–4833.
[15] M. Berliant, W. Thomson, K. Dunz, On the fair division of a heterogeneous commodity, J. Math. Econ. 21 (3) (1992) 201–216.
[16] U. Bhaskar, A. Sricharan, R. Vaish, 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, 2021.
[17] A. Bogomolnaia, H. Moulin, F. Sandomirskiy, E. Yanovskaia, Dividing bads under additive utilities, Soc. Choice Welf. 52 (3) (2019) 395–417.
[18] S.J. Brams, D.M. Kilgour, C. Klamler, Maximin envy-free division of indivisible items, Group Decis. Negot. 26 (1) (2017) 115–131.
[19] E. Budish, The combinatorial assignment problem: approximate competitive equilibrium from equal incomes, J. Polit. Econ. 119 (6) (2011) 1061–1103.
[20] I. Caragiannis, N. Gravin, X. Huang, Envy-freeness up to any item with high Nash welfare: the virtue of donating items, in: Proceedings of the 2019 ACM Conference on Economics and Computation, 2019, pp. 527–545.
[21] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, The efficiency of fair division, Theory Comput. Syst. 50 (4) (2012) 589–610.
[22] I. Caragiannis, D. Kurokawa, H. Moulin, A.D. Procaccia, N. Shah, J. Wang, The unreasonable fairness of maximum Nash welfare, ACM Trans. Econ. Comput. 7 (3) (2019) 1–32.
[23] B.R. Chaudhury, J. Garg, K. Mehlhorn, EFX exists for three agents, in: Proceedings of the 21st ACM Conference on Economics and Computation, 2020, pp. 1–19.
[24] B.R. Chaudhury, T. Kavitha, K. Mehlhorn, A. Sgouritsa, A little charity guarantees almost envy-freeness, in: Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, 2020, pp. 2658–2672.
[25] X. Chen, Z. Liu, The fairness of leximin in allocation of indivisible chores, arXiv preprint, arXiv:2005.04864, 2020.
[26] V. Conitzer, R. Freeman, N. Shah, J.W. Vaughan, Group fairness for the allocation of indivisible goods, Proc. AAAI Conf. Artif. Intell. 33 (2019) 1853–1860.
[27] W.F. Dowling, J.H. Gallier, Linear-time algorithms for testing the satisfiability of propositional Horn formulae, J. Log. Program. 1 (3) (1984) 267–284.
[28] D.K. Foley, Resource Allocation and the Public Sector, Yale University, 1966.
[29] R. Freeman, S. Sikdar, R. Vaish, L. Xia, Equitable allocations of indivisible goods, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 280–286.
[30] R. Freeman, S. Sikdar, R. Vaish, L. Xia, Equitable allocations of indivisible chores, in: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, 2020, pp. 384–392.
[31] Y. Gafni, X. Huang, R. Lavi, I. Talgam-Cohen, Unified fair allocation of goods and chores via copies, arXiv preprint, arXiv:2109.08671, 2021.
[32] J. Garg, A. Murhekar, J. Qin, Fair and efficient allocations of chores under bivalued preferences, Proc. AAAI Conf. Artif. Intell. 36 (2022) 5043–5050.
[33] L. Gourvès, J. Monnot, L. Tlilane, Near fairness in matroids, in: Proceedings of the Twenty-First European Conference on Artificial Intelligence, 2014, pp. 393–398.
[34] P.L. Hammer, A. Kogan, Optimal compression of propositional Horn knowledge bases: complexity and approximation, Artif. Intell. 64 (1) (1993) 131–145.
[35] M. Kyropoulou, W. Suksompong, A.A. Voudouris, Almost envy-freeness in group resource allocation, Theor. Comput. Sci. 841 (2020) 110–123.
[36] B. Li, Y. Li, X. Wu, Almost (weighted) proportional allocations for indivisible chores, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 122–131.
[37] R.J. Lipton, E. Markakis, E. Mossel, A. Saberi, On approximately fair allocations of indivisible goods, in: Proceedings of the 5th ACM Conference on Electronic Commerce, 2004, pp. 125–131.
[38] S. Liu, X. Lu, M. Suzuki, T. Walsh, Mixed fair division: a survey, arXiv preprint, arXiv:2306.09564, 2023.
[39] R. Mahara, Existence of EFX for two additive valuations, arXiv preprint, arXiv:2008.08798, 2020.
[40] B. Plaut, T. Roughgarden, Almost envy-freeness with general valuations, SIAM J. Discrete Math. 34 (2) (2020) 1039–1068.
[41] B. Plaut, T. Roughgarden, Communication complexity of discrete fair division, SIAM J. Comput. 49 (1) (2020) 206–243.
[42] H.R. Varian, Equity, envy, and efficiency, J. Econ. Theory 9 (1) (1974) 63–91.
[43] T. Walsh, Fair division: the computer scientist's perspective, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4966–4972.
[44] L. Yin, R. Mehta, On the envy-free allocation of chores, arXiv preprint, arXiv:2211.15836, 2022.