Temporal Fair Division of Indivisible Items

Edith Elkind[1], Alexander Lam[2], Mohamad Latifian[3], Tzeh Yuan Neoh[4], Nicholas Teh[5]
[1]Northwestern University, USA [2]Hong Kong Polytechnic University, Hong Kong SAR
[3]University of Edinburgh, UK [4]Harvard University, USA [5]University of Oxford, UK

Abstract

We study a fair division model where indivisible items arrive sequentially, and must be allocated immediately and irrevocably. Previous work on online fair division has shown impossibility results for achieving approximate envy-freeness under the assumption that agents have no information about future items. In contrast, we assume that the algorithm has complete knowledge of the future, and aim to ensure that the cumulative allocation at each round satisfies approximate envy-freeness, which we define as *temporal envy-freeness up to one item* (TEF1). We focus on settings where items are exclusively goods or exclusively chores. For goods, while TEF1 allocations may fail to exist, we identify several special cases where they do—two agents, two item types, generalized binary valuations, unimodal preferences—and provide polynomial-time algorithms for these cases. We also prove that determining the existence of a TEF1 allocation is NP-hard. For chores, we obtain analogous results for the special cases, but present a slightly weaker intractability result. We also show that TEF1 is incompatible with Pareto optimality, with the implication that it is intractable to find a TEF1 allocation that maximizes any p-mean welfare, even for two agents.

1 Introduction

Fair division, a topic at the intersection of economics and computer science, has been extensively studied over the years, with applications ranging from divorce settlements and inheritance disputes to load balancing [23, 56]. Typically, in fair division there is a set of agents and a set of items, and the goal is to obtain a fair allocation of items to agents. In our work, we study a model where these items are *indivisible*, so each must be wholly allocated to an agent. Moreover, the items can provide either positive utility (in which case they are called *goods*) or negative utility (in which case they are called *chores*, or *tasks*). When allocating indivisible items, a desirable and widely-studied fairness notion is *envy-freeness up to one item (EF1)*, a natural relaxation of *envy-freeness (EF)*. In an envy-free allocation, each agent values the bundle of items they receive at least as highly as every other agent's bundle. However, this desideratum is not always achievable for indivisible items (consider two agents and a single item that they both value). In contrast, in an EF1 allocation, the envy that agent A has towards another agent B can be eliminated by removing a single item from B's bundle (in case of goods) or A's bundle (in case of chores).

Most prior research studies fair division in the *offline* setting, assuming that all of the items are immediately available and ready to be allocated. However, there are various applications where the items arrive and need to be allocated on the spot in a sequential manner. For example, when the university administration places an order for lab equipment, or when a company orders new machines for its franchises, the items may arrive over time due to their availabilities and delivery logistics. In case of chores, collaborative project management may require division of tasks over time. For a variety of reasons, arriving items may have to be allocated immediately; there may not be any storage space to keep any unallocated goods, or the central decision maker may desire a non-wasteful allocation in the sense that items or tasks should not sit idle for periods of time.

These applications can be captured by an *online fair division* model, in which items arrive over time and must be immediately and irrevocably allocated, though it is assumed that each item's valuation is not known until its arrival. Prior research has found that a complete EF1 allocation of goods cannot be guaranteed under the online fair division model[1] [19]. However, this result relies on the assumption that the algorithm has no information about the future. In contrast, in our examples, the delivery services could provide the estimated delivery dates, and there may be a pre-planned timeline for the tasks. Motivated by these nuances, our work studies the *informed online fair division setting*, assuming that the algorithm can access the items' valuations and arrival order upfront.

Note that the assumption of complete information about the future trivially leads to a complete EF1 allocation at the *end* of the allocation period: simply treat the instance as an offline problem and apply any algorithm known to satisfy EF1 (e.g., [52, 9, 26]). However, this approach ignores the cumulative bundles of the items throughout the allocation period, and, consequently, agents may feel that their partial allocations are unfair for extended periods of time. Inspired by this issue, we propose *temporal EF1 (TEF1)*, an extension of EF1 to the informed online fair division setting which requires that at each round, the cumulative allocation satisfies EF1.

The main focus of our work is on achieving TEF1, so for the informed online fair division of indivisible goods or chores, we aim to answer the following existence and computational questions:

> *Which restricted settings guarantee the existence of a TEF1 allocation, and can we compute such an allocation in polynomial time in these settings? Is it computationally tractable to determine the existence[2] of a TEF1 allocation? In terms of existence and tractability, is TEF1 compatible with natural notions of efficiency?*

1.1 Our Contributions

We outline our paper's answers to these key questions as follows.

In Section 3, we show the existence of TEF1 allocations (for goods or chores) in restricted settings, such as the case of two agents, when there are two types of items, when agents have generalized binary valuations, or when they have unimodal preferences. For each of these cases, we provide an accompanying polynomial-time algorithm. For the allocation of goods, we show that determining whether there exists a TEF1 allocation is NP-hard; whereas for chores, we show that given a partial TEF1 allocation, it is NP-hard to determine if there exists a TEF1 allocation that allocates all remaining chores.

In Section 4, we investigate the compatibility of TEF1 and Pareto-optimality (PO). We show that even in the case of two agents, while a TEF1 allocation is known to exist and can be computed in polynomial time (for both goods and chores), existence is no longer guaranteed if we mandate PO as well. Moreover, we show that in this same setting, determining the existence of TEF1 and PO allocations is NP-hard. Our result also directly implies the computational intractability of determining whether there exists a TEF1 allocation that maximizes any p-mean welfare objective (which subsumes most popular social welfare objectives).

Finally, in Section 5, we consider the special case where the same set of items arrive at each round, and show that even determining whether repeating a particular allocation in two consecutive rounds can result in a TEF1 allocation is NP-hard. We complement this with a polynomial-time algorithm for computing a TEF1 allocation in this case when there are just two rounds.

[1] In fact, the maximum pairwise envy is $\Omega(\sqrt{t})$ after t rounds in the worst case.

[2] Prior work by He et al. [46] has shown that a TEF1 allocation is not guaranteed to exist for goods in the general setting with three or more agents.

1.2 Related Work

Our work is closely related to *online fair division*, whereby items arrive over time and must be irrevocably allocated to agents. The key difference is that in the standard online setting, the algorithm has completely no information on future items, whereas we assume complete future information. Moreover, the goal in online fair division models is typically to guarantee a fair allocation to agents at the end of the time horizon, rather than at every round.

As we focus on EF1, papers satisfying envy-based notions in online allocations are particularly relevant. Aleksandrov et al. [4] consider envy-freeness from both ex-ante and ex-post standpoints, giving a best-of-both-worlds style result by designing an algorithm for goods which is envy-free in expectation and guarantees a bounded level of envy-freeness. Additionally, Benadè et al. [19] find that allocating goods uniformly at random leads to maximum pairwise envy which is sublinear in the number of rounds. For further reading, we refer the reader to the surveys by Aleksandrov and Walsh [3] and Amanatidis et al. [6].

There has also been work on online fair division with *partial information* on future items. Benadè et al. [19] study the extent to which approximations of envy-freeness and Pareto efficiency can be simultaneously satisfied under a spectrum of information settings, ranging from identical agents and i.i.d. valuations to zero future information. An emerging line of work on *learning-augmented* online algorithms has an alternate approach to partial future information: the algorithms are aided by (possibly inaccurate) predictions, typically from a machine-learning algorithm. The focus is to design algorithms which perform *consistently* well with accurate predictions, and are *robust* under inaccurate predictions. These predictions could be of each agent's total utility for the entire item set [11, 12], or for a random subset of k incoming items [20].

Unlike the aforementioned papers, our work considers a completely *informed* variant of the online fair division setting, which has been studied by He et al. [46] for the allocation of goods. Similar to our paper, their objective is to ensure that EF1 is satisfied at each round, but they allow agents to swap their bundles. When multiple goods may arrive at each round, our setting also generalizes the *repeated* fair division setting, in which the same set of goods arrives at each round. For this model, Igarashi et al. [47] give results on the existence of allocations which are envy-free and Pareto optimal in the end, with the items in each round being allocated in an EF1 manner. However, they do not analyse whether the *cumulative* allocation at each round can satisfy some fairness constraint, which is the focus of our paper. Caragiannis and Narang [25] also consider a model where the same set of items appear at each round, but each agent gets exactly one item per round.

When the valuations are known upfront for the allocation of chores, the model is similar to the field of work on job scheduling. There have been numerous papers studying fair scheduling, but the fairness is typically represented by an objective function which the algorithm aims to minimize or approximate [63, 48, 17]. On the other hand, there is little work on satisfying envy-based notions in scheduling problems, but Li et al. [51] study the compatibility of EF1 and Pareto optimality in various settings. While we consider separately the cases of goods allocation and chores allocation, to the best of our knowledge, there is no prior work which studies an online fair division model with both goods and chores in the same instance under any information assumption.

Similar temporal models that study concepts of achieving fairness over time have also been recently considered in the social choice literature [5, 27, 35, 38, 37, 53, 57, 59, 50, 69].

In a contemporary and independent work, Cookson et al. [32] consider the same setting of temporal fair division but with a different approach. They only consider goods and prove positive result in the three settings: when there are only two agents, when the identical set of goods appear in each timestep, and when agents have an identical ranking over the items. They consider several fairness notions and seek to achieve different pairs of these notions per-day and overall.

2 Preliminaries

For each positive integer k, let $[k] := \{1, \ldots, k\}$. We consider the problem of fairly allocating indivisible items to agents over multiple rounds. An instance of the *informed online fair division problem* is a tuple $\mathcal{I} = \langle N, T, \{O_t\}_{t\in[T]}, \mathbf{v} = (v_1, \ldots, v_n)\rangle$, where $N = [n]$ is a set of *agents*, T is the number of *rounds*, for each $t \in [T]$ the set O_t consists of items that arrive at round t, with $O = \cup_{t\in[T]} O_t$, and for each $i \in N$ the *valuation function* $v_i : O \to \mathbb{R}$ specifies the values that agent i assigns to items in O.

We assume that agents have additive valuations, i.e., we extend the functions v_i to subsets of O by setting $v_i(S) = \sum_{o\in S} v_i(O)$ for each $S \subseteq O$. We write v instead of v_i when all agents have identical valuation functions. We refer to the vector $\mathbf{v} = (v_1, \ldots, v_n)$ as the *valuation profile*. We define the *cumulative set of items* that arrive in rounds $1, \ldots, t$ by $O^t := \bigcup_{\ell\in[t]} O_\ell$. Note that $O = O^T$.

We consider both *goods*, where $v_i(o) \geq 0$ for each $i \in N$ and $o \in O$, and *chores*, where $v_i(o) \leq 0$ for each $i \in N$ and $o \in O$. For clarity, in the goods setting we use g instead of o and refer to the items as *goods*, while in the chores setting we use c instead of o and refer to the items as *chores*.

An *allocation* $\mathcal{A} = (A_1, \ldots, A_n)$ of items in O to the agents is an ordered partition of O, i.e., $\bigcup_{i\in N} A_i = O$ and $A_i \cap A_j = \varnothing$ for all $i, j \in N$ with $i \neq j$. For $t \in [T]$, $i \in N$ we write $A_i^t = A_i \cap O^t$; then $\mathcal{A}^t = (A_1^t, \ldots, A_n^t)$ is the allocation after round t, with $\mathcal{A} = \mathcal{A}^T$. For $t < T$, we may refer to $\mathcal{A}^t$ as a *partial allocation.*

Our goal is to find an allocation that is fair after each round. The main fairness notion that we consider is *envy-freeness up to one item (EF1)*, a well-studied notion in fair division.

Definition 2.1. In a goods (resp., chores) allocation instance, an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is said to be EF1 if for each pair of agents $i, j \in N$ there exists a good $g \in A_j$ (resp., chore $c \in A_i$) such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$ (resp. $v_i(A_i \setminus \{c\}) \geq v_i(A_j)$).

To capture fairness in a *cumulative* sense, we introduce the notion of *temporal envy-freeness up to one item (TEF1)*, which requires that at every prefix of rounds the cumulative allocation of items that have arrived so far satisfies EF1.

Definition 2.2 (Temporal EF1)**.** For every $t \in [T]$, an allocation $\mathcal{A}^t = (A_1^t, \ldots, A_n^t)$ is said to be *temporally envy-free up to one item (TEF1)* if for each $\ell \in [t]$ the allocation $\mathcal{A}^\ell$ is EF1.

A key distinction between TEF1 and EF1 is that, while the EF1 property only places constraints on the final allocation, TEF1 requires envy-freeness up to one item at every round.

However, He et al. [46, Thm. 4.2] show that for goods TEF1 allocations may fail to exist; they present an example with 3 agents and 23 items, which can be generalized to $n > 3$ agents. For completeness, we include this counterexample along with an intuitive explanation in the appendix. We remark that the construction of He et al. [46] cannot be translated to the chores setting. While we conjecture that a non-existence result of this form also holds for chores, this remains an open question.

We assume that the reader is familiar with basic notions of classic complexity theory [60]. All omitted proofs can be found in the appendix.

3 On the Existence of TEF1 Allocations

As some instances do not admit TEF1 allocations, our first goal is to explore if there are restricted classes of instances for which TEF1 allocations are guaranteed to exist. In this section we identify several such settings.

To simplify the presentation, we will first demonstrate that it usually suffices to consider instances where only one item appears at each round (i.e., $T = m$ and $|O_t| = 1$ for all $t \in [T]$). Indeed, any impossibility result for this special setting also holds for the general case, and we will now argue that the converse is true as well.

Lemma 3.1. *Given an instance $\mathcal{I}$ with $|O| = m$ items, we can construct an instance $\mathcal{I}^{=1}$ with the same set of items and exactly m rounds so that $|O_t| = 1$ for each $t \in [m]$ and if $\mathcal{I}^{=1}$ admits a* TEF1 *allocation, then so does $\mathcal{I}$.*

Proof. Consider an arbitrary instance $\mathcal{I} = \langle N, T, \{O_t\}_{t\in[T]}, \mathbf{v} = (v_1, \dots, v_n)\rangle$. Renumber the items in a non-decreasing fashion with respect to the rounds, so that for any two rounds $t, r \in [T]$ with $t < r$ and items $o_j \in O_t, o_{j'} \in O_r$ it holds that $j < j'$. We construct $\mathcal{I}^{=1} = \langle N, m, \{\widetilde{O}_t\}_{t\in[m]}, \mathbf{v}\rangle$ by setting $\widetilde{O}_t = \{o_t\}$ for each $t \in [m]$. Let $\mathcal{A}$ be a TEF1 allocation for $\mathcal{I}^{=1}$. We construct an allocation $\mathcal{B}$ for instance $\mathcal{I}$ by allocating all items in the same way as in $\mathcal{A}$: if $\mathcal{A}$ allocates at item j to agent i in round r, we identify a $t \in [T]$ such that $\sum_{\ell=1}^{t-1} |O_\ell| < r \le \sum_{\ell=1}^{t} |O_\ell|$ and place j into B_i in round t. To see that $\mathcal{B}$ satisfies TEF1, note that if $\mathcal{B}^t$ violates EF1 for some $t \in [T]$, then for $r = \sum_{\ell=1}^{t} |O_\ell|$ the allocation $\mathcal{A}^r$ satisfies $A_i^r = B_i^t$ for all $i \in N$ and hence violates EF1 as well. □

In what follows, unless specified otherwise, we simplify the notation based on the transformation in the proof of Lemma 3.1: we assume that $|O_t| = 1$ for each $t \in T$ and denote the unique item that arrives in round t by o_t (or g_t, or c_t, if we focus on goods/chores).

3.1 Two Agents

He et al. [46, Thm. 3.4] put forward a polynomial-time algorithm that always outputs a TEF1 allocation for goods when $n = 2$; in particular, this implies that a TEF1 allocation is guaranteed to exist for $n = 2$. We will now extend this result to the case of chores.

Intuitively, in each round the algorithm greedily allocates the unique chore that arrives in that round to an agent that does not envy the other agent in the current (partial) allocation. A counter s keeps track of the last round in which $\mathcal{A}^s$ was envy-free; if for some round $t \in [m]$ the allocation of a chore c_t results in both agents envying each other in $\mathcal{A}^t \setminus \mathcal{A}^s$, then the agents' bundles in $\mathcal{A}^t \setminus \mathcal{A}^s$ are swapped.

Theorem 3.2. *For $n = 2$, Algorithm 1 returns a* TEF1 *allocation for chores, and runs in polynomial time.*

Next, we consider *temporal envy-freeness up to any item (TEFX)*, the temporal variant of the stronger notion of *envy-freeness up to any item (EFX)*.

Definition 3.3. In a goods (resp., chores) allocation instance, an allocation $\mathcal{A} = (A_1, \dots, A_n)$ is said to be EFX if for all pairs of agents $i, j \in N$, and all goods $g \in A_j$ (resp., chores $c \in A_i$) we have $v_i(A_i) \ge v_i(A_j \setminus \{g\})$ (resp. $v_i(A_i \setminus \{c\}) \ge v_i(A_j)$).

Definition 3.4 (Temporal EFX)**.** For every $t \in [T]$, an allocation $\mathcal{A}^t = (A_1^t, \dots, A_n^t)$ is said to be *temporal envy-free up to any item (TEFX)* if for each $\ell \le t$ the allocation $\mathcal{A}^\ell$ is EFX.

Unfortunately, TEFX allocations (for goods or chores) may not exist, even for two agents with identical valuations, and even when there are only two types of items.

Proposition 3.5. *A* TEFX *allocation for goods or chores may not exist, even for $n = 2$ with identical valuations and two types of items.*

3.2 Other Restricted Settings

The next natural question we ask is whether there are other special cases where EF1 allocation is guaranteed to exist. We answer this question affirmatively by demonstrating the existence of EF1 allocations in three special cases, each supported by a polynomial-time algorithm that returns such an allocation.

Algorithm 1 Returns a TEF1 allocation for chores when $n = 2$

Input: Set of agents $N = \{1, \ldots, n\}$, set of chores $O = \{c_1, \ldots, c_m\}$, and valuation profile $\mathbf{v} = (v_1, v_2)$
Output: TEF1 allocation $\mathcal{A}$ of chores in O to agents in N

Initialize $s \leftarrow 0$ and $\mathcal{A}^0 \leftarrow (\varnothing, \varnothing)$
for $t = 1, 2, \ldots, m$ **do**
  if $v_1(A_1^{t-1} \setminus A_1^s) \geq v_1(A_2^{t-1} \setminus A_2^s)$ **then**
    $\mathcal{A}^t \leftarrow (A_1^{t-1} \cup \{c_t\}, A_2^{t-1})$
  else
    $\mathcal{A}^t \leftarrow (A_1^{t-1}, A_2^{t-1} \cup \{c_t\})$
  end if
  if $v_1(A_1^t \setminus A_1^s) < v_1(A_2^t \setminus A_2^s)$ and $v_2(A_2^t \setminus A_2^s) < v_2(A_1^t \setminus A_1^s)$ **then**
    $\mathcal{A}^t \leftarrow (A_1^s \cup A_2^t \setminus A_2^s, A_2^s \cup A_1^t \setminus A_1^s)$
  end if
  if $v_1(A_1^t \setminus A_1^s) \geq v_1(A_2^t \setminus A_2^s)$ and $v_2(A_2^t \setminus A_2^s) \geq v_2(A_1^t \setminus A_1^s)$ **then**
    $s \leftarrow t$
  end if
end for
return $\mathcal{A} = (A_1^m, A_2^m)$

Two Types of Items

The first setting we consider is one where items can be divided into two *types*, and each agent values all items of a particular type equally. Formally, let $S_1, S_2 \subseteq O$ be a partition of the set of items, so that $S_1 \cap S_2 = \varnothing$, and $S_1 \cup S_2 = O$. Then, for any $r \in \{1, 2\}$, two items $o, o' \in S_r$, and agent $i \in N$, we have that $v_i(o) = v_i(o')$.

Settings with only two types of items/tasks arise naturally in various applications, such as distributing food and clothing donations from a charity, or allocating cleaning and cooking chores in a household. This preference restriction has been studied for chores in offline settings [10, 42], and we remark that agents may have distinct valuations for up to $2n$ different items, unlike the extensively studied *bi-valued* preferences [33, 41] which involve only two distinct item values.

We show that for this setting, a TEF1 allocation for goods or chores always exists and can be computed in polynomial time. Intuitively, the algorithm treats the two item types independently: items of the first type are allocated in a round-robin manner from agent 1 to n, while items of the second type are allocated in reverse round-robin order from agent n to 1. Then, our result is as follows.

Theorem 3.6. *When there are two types of items, a* TEF1 *allocation for goods or chores exists and can be computed in polynomial time.*

Generalized Binary Valuations

The next setting we consider is one where agents have *generalized binary valuations* (also known as *restricted additive valuations* [1, 24]). This class of valuation functions generalizes both identical and binary valuations, which are both widely studied in fair division [45, 61, 65]. Formally, we say that agents have *generalized binary valuations* if for every agent $i \in N$ and item $o_j \in O$, $v_i(o_j) \in \{0, p_j\}$, where $p_j \in \mathbb{R} \setminus \{0\}$.

We show that for this setting, a TEF1 allocation can be computed efficiently, with the following result.

We remark that the resulting allocation also satisfies *Pareto-optimality* (Definition 4.1).

Theorem 3.7. *When agents have generalized binary valuations, a* TEF1 *allocation for goods or chores exists and can be computed in polynomial time.*

Unimodal Preferences

The last setting that we consider is the class of *unimodal preferences*, which consists of the widely studied *single-peaked* and *single-dipped* preference structures in social choice [22, 7] and cake cutting [67, 21]. We adapt these concepts for the online fair division setting with a single item at each timestep.

Definition 3.8. A valuation profile $\mathbf{v}$ is *single-peaked* if for each agent $i \in N$, there is an item o_{i^*} where for each $j, k \in [m]$ such that $j < k < i^*$, $v_i(o_j) \leq v_i(o_k) \leq v_i(o_{i^*})$, and for each $j, k \in [m]$ such that $i^* < j < k$, $v_i(o_{i^*}) \geq v_i(o_j) \geq v_i(o_k)$.

Definition 3.9. A valuation profile $\mathbf{v}$ is *single-dipped* if for each agent $i \in N$, there is an item o_{i^*} where for each $j, k \in [m]$ such that $j < k < i^*$, $v_i(o_j) \geq v_i(o_k) \geq v_i(o_{i^*})$, and for each $j, k \in [m]$ such that $i^* < j < k$, $v_i(o_{i^*}) \leq v_i(o_j) \leq v_i(o_k)$.

In other words, under single-peaked (resp. single-dipped) valuations, agents have a specific item o_{i^*} that they prefer (resp. dislike) the most, and prefer (resp. dislike) items less as they arrive further away in time from o_{i^*}.

Note that this restricted preference structure is well-defined for the setting of a single item arriving per round, but may not be compatible with a generalization to multiple items per round as described in Lemma 3.1 (unless the items in each round are identically-valued by agents).[3]

Unimodal preferences may arise in settings where agents place higher value on resources at the time surrounding specific events. For example, in disaster relief, the demand for food and essential supplies peaks as a natural disaster approaches, then declines once the immediate crisis passes. Similarly, in project management, the workload for team members intensifies (in terms of required time and effort) as the project nears its deadline, but significantly decreases during the final stages, such as editing and proofreading.

Unimodal preferences also generalizes other standard preference restrictions studied in fair division and voting models, such as settings where agents have *monotonic valuations* [39] or *identical rankings* [61].

We propose efficient algorithms for computing a TEF1 allocation for goods when agents have single-peaked valuations, and for chores when agents have single-dipped valuations.

Theorem 3.10. *When agents have single-peaked valuations, a* TEF1 *allocation for goods exists and can be computed in polynomial time. When agents have single-dipped valuations, a* TEF1 *allocation for chores exists and can be computed in polynomial time.*

We note that while a simple greedy algorithm performs well in the case of single-peaked valuations for goods and single-dipped valuations for chores, it fails in the reverse scenario—single-dipped valuations for goods and single-peaked valuations for chores. This is due to the fact that, in the latter case, the position of the dip or peak becomes critical and significantly complicates the way we allocate the item. We leave the existence of polynomial-time algorithm(s) for the reverse scenario as an open question.

[3] Specifically, in the multiple items per round case, if the bundles of items at each timestep are unimodally valued, the single-item per round transformation of the instance may not necessarily be unimodal.

3.3 Hardness Results for TEF1 Allocations

The non-existence of TEF1 goods allocations for $n \geq 3$ prompts us to explore whether we can determine if a given instance admits a TEF1 allocation for goods. Unfortunately, we show that this problem is NP-hard, with the following result.

Theorem 3.11. *Given an instance of the temporal fair division problem with goods and $n \geq 3$, determining whether there exists a* TEF1 *allocation is* NP*-hard.*

Proof. We reduce from the 1-IN-3-SAT problem which is NP-hard. An instance of this problem consists of a conjunctive normal form formula F with three literals per clause; it is a yes instance if there exists a truth assignment to the variables such that each clause has exactly one `True` literal, and a no instance otherwise.

Consider an instance of 1-IN-3-SAT given by the CNF F which contains n variables $\{x_1, \dots, x_n\}$ and m clauses $\{C_1, \dots, C_m\}$. We construct an instance $\mathcal{I}$ with three agents and $2n+2$ goods. For each $i \in [n]$, we introduce two goods t_i, f_i. We also introduce two additional goods s and r. Let the agents' (identical) valuations be defined as follows:

$$v(g) = \begin{cases} 5^{m+n-i} + \sum_{j\,:\,x_i \in C_j} 5^{m-j}, & \text{if } g = t_i, \\ 5^{m+n-i} + \sum_{j\,:\,\neg x_i \in C_j} 5^{m-j}, & \text{if } g = f_i, \\ \sum_{j \in [m]} 5^{j-1}, & \text{if } g = r, \\ \sum_{i \in [n]} 5^{m+i-1} + 2 \times \sum_{j \in [m]} 5^{j-1}, & \text{if } g = s. \end{cases}$$

Intuitively, for each variable index $i \in [n]$, we associate with it a unique value 5^{m+n-i}. For each clause index $j \in [m]$, we also associate with it a unique value 5^{m-j}. Note that no two indices (regardless of whether its a variable or clause index) share the same value. Then, the value of each good t_i comprises of the unique value associated with i, and the sum over all unique values of clauses C_j which x_i appears as a *positive literal* in; whereas the value of each good f_i comprises of the unique value associated with i, and the sum over all unique values of clauses C_j which x_i appears as a *negative literal* in. We will utilize this in our analysis later.

Then, we have the set of goods $O = \{s, t_1, f_1, t_2, f_2, \dots, t_n, f_n, r\}$. Note that $v(O) = v(s) + v(r) + \sum_{i \in [n]} v(t_i) + \sum_{i \in [n]} v(f_i)$. Also observe that $\sum_{i \in [n]} 5^{m+n-i} = \sum_{i \in [n]} 5^{m+i-1}$. Now, as each clause contains exactly three literals, we have

$$\sum_{i \in [n]} \sum_{j : x_i \in C_j} 5^{m-j} + \sum_{i \in [n]} \sum_{j : \neg x_i \in C_j} 5^{m-j} = 3 \times \sum_{j \in [m]} 5^{j-1}.$$

Then, combining the equations above, we get that

$$v(O) = 3 \times \sum_{i \in [n]} 5^{m+i-1} + 6 \times \sum_{j \in [m]} 5^{j-1}. \tag{1}$$

Let the goods be in the following order: $s, t_1, f_1, t_2, f_2, \dots, t_n, f_n, r$. We first prove the following result.

Lemma 3.12. *There exists a truth assignment α such that each clause in F has exactly one `True` literal if and only if there exists an allocation $\mathcal{A}$ such that $v(A_1) = v(A_2) = v(A_3)$ for instance $\mathcal{I}$.*

Proof. For the 'if' direction, consider an allocation $\mathcal{A}$ such that $v(A_1) = v(A_2) = v(A_3)$. Then, we have that $O = A_1 \cup A_2 \cup A_3$ and $v(A_1) = v(A_2) = v(A_3) = \frac{1}{3}v(O)$. Since agents have identical valuations, without loss of generality, let $s \in A_1$. Then, since $v(A_1^1) = v(s) = \frac{1}{3}v(O)$, agent 1 should not receive any more goods after s, and each remaining good should go to agent 2 or 3.

Again, without loss of generality, we let $r \in A_2$. Then since $v(A_2) = \frac{1}{3}v(O)$, we have that

$$v(A_2 \setminus \{r\}) = \left(\sum_{i\in[n]} 5^{m+i-1} + 2 \times \sum_{j\in[m]} 5^{j-1}\right) - \sum_{j\in[m]} 5^{j-1}$$
$$= \sum_{i\in[n]} 5^{m+i-1} + \sum_{j\in[m]} 5^{j-1}.$$

Note that this is only possible if for each $i \in [m]$, t_i and f_i are allocated to different agents. The reason is because the only way agent 1 can obtain the first term of the above bundle value (less good r) is if he is allocated exactly one good from each of $\{t_i, f_i\}$ for all $i \in [n]$.

Then, from the goods that exist in bundle A_2, we can construct an assignment α: for each $i \in [n]$, let $x_i = \texttt{True}$ if $t_i \in A_2$ and $x_i = \texttt{False}$ if $f_i \in A_2$. Then, from the second term in the expression of $v(A_1 \setminus \{r\})$ above, we can observe that each clause has exactly one `True` literal (because the sum is only obtainable if exactly one literal appears in each clause, and our assignment will set each of these literals to `True`).

For the 'only if' direction, consider a truth assignment α such that each clause in F has exactly one `True` literal. Then, for each $i \in [n]$, let

$$\ell_i = \begin{cases} t_i & \text{if } x_i = \texttt{True} \text{ under } \alpha, \\ f_i & \text{if } x_i = \texttt{False} \text{ under } \alpha. \end{cases}$$

We construct the allocation $\mathcal{A} = (A_1, A_2, A_3)$ where

$$A_1 = \{s\}, \quad A_2 = \{\ell_1, \dots, \ell_n, r\}, \quad \text{and} \quad A_3 = O \setminus (A_1 \cup A_2).$$

Again, observe that $\sum_{i\in[n]} 5^{m+n-i} = \sum_{i\in[n]} 5^{m+i-1}$. Also note that $v(A_1) = \frac{1}{3}v(O)$. Then, as each clause has exactly one `True` literal, $v(A_2) = \sum_{i\in[n]} 5^{m+i-1} + 2 \times \sum_{j\in[m]} 5^{j-1}$, and together with (1), we get that $v(A_3) = \frac{2}{3}v(O) - v(A_1) = v(A_1)$ and hence $v(A_1) = v(A_2) = v(A_3)$, as desired. □

Now consider another instance $\mathcal{I}'$ that is similar to $\mathcal{I}$, but with an additional 21 goods $\{g_1, \dots, g_{21}\}$. Let agents' valuations over these new goods be defined as follows:

v	g_1	g_2	g_3	g_4	g_5	g_6	g_7
1	90	80	70	100	100	100	15
2	90	70	80	100	100	100	95
3	80	90	70	100	100	100	25
	g_8	g_9	g_{10}	g_{11}	g_{12}	g_{13}	g_{14}
1	10000	11000	12000	20000	20000	20000	20000
2	10000	11000	12000	20000	20000	20000	20000
3	10000	11000	12000	20000	20000	18500	20000
	g_{15}	g_{16}	g_{17}	g_{18}	g_{19}	g_{20}	g_{21}
1	20000	20000	20000	20000	20000	19010	18005
2	20000	20000	20000	12000	12000	19085	14106
3	20000	20000	20000	20000	20000	19010	19496

Then, we have the set of goods $O' = O \cup \{g_1, \dots, g_{21}\}$. Let the goods be in the following order: $s, t_1, f_1, t_2, f_2, \dots, t_n, f_n, r, g_1, \dots, g_{21}$. We now present the final lemma that will give us our result.

Lemma 3.13. *If there exists a partial allocation $\mathcal{A}^{2n+2}$ over the first $2n+2$ goods such that $v(A_1^{2n+2}) = v(A_2^{2n+2})$, then there exists a* TEF1 *allocation $\mathcal{A}$. Conversely, if there does not exist a partial allocation $\mathcal{A}^{2n+2}$ over the first $2n+2$ goods such that $v(A_1^{2n+2}) = v(A_2^{2n+2})$, then there does not exists a* TEF1 *allocation $\mathcal{A}$.*

We use a program as a gadget to verify the lemma (see the full version of the paper), leveraging its output to support its correctness. Specifically, if there exists a partial allocation $\mathcal{A}^{2n+2}$ over the first $2n+2$ goods such that $v(A_1^{2n+2}) = v(A_2^{2n+2})$, then our program will show the existence of a TEF1 allocation by returning all such TEF1 allocations. If there does not exist such a partial allocation, our program essentially does an exhaustive search to show that a TEF1 allocation does not exist. This lemma shows that there exists a TEF1 allocation over O' if and only if $v(A_1^{2n+2}) \neq v(A_2^{2n+2})$, and by Claim 3.12, this implies that a TEF1 allocation over O' exists if and only if there is a truth assignment α such that each clause in F has exactly one `True` literal. □

We note that the above approach cannot be extended to show hardness for the setting with chores. Nevertheless, we are able to show a similar, though weaker, intractability result for the case of chores in general. The key difference is that we assume that we can start from any partial TEF1 allocation.

Theorem 3.14. *For every $t \in [T]$, given any partial* TEF1 *allocation $\mathcal{A}^t$ for chores, deciding if there exists an allocation $\mathcal{A}$ that is* TEF1 *is* NP*-hard.*

4 Compatibility of TEF1 and Efficiency

In traditional fair division, many papers have focused on the existence and computation of fair and efficient allocations for goods or chores, with a particular emphasis on simultaneously achieving EF1 and *Pareto-optimality (PO)* [13, 26]. In this section, we explore the compatibility between TEF1 and PO. We begin by defining PO as follows.

Definition 4.1 (Pareto-optimality)**.** We say that an allocation $\mathcal{A}$ is *Pareto-optimal (PO)* if there does not exist another allocation $\mathcal{A}'$ such that for all $i \in N$, $v_i(A'_i) \geq v_i(A_i)$, and for some $j \in N$, $v_j(A'_j) > v_j(A_j)$. If such an allocation $\mathcal{A}'$ exists, we say that $\mathcal{A}'$ *Pareto-dominates* $\mathcal{A}$.

Observe that for any $\mathcal{A}$ that is PO, any partial allocation $\mathcal{A}^t$ for $t \leq [T]$ is necessarily PO as well. We demonstrate that PO is incompatible with TEF1 in this setting, even under very strong assumptions (of two agents and two types of items), as illustrated by the following result.

Proposition 4.2. *For any $n \geq 2$, a* TEF1 *and* PO *allocation for goods or chores may not exist, even when there are two types of items.*

Despite this non-existence result, one may still wish to obtain a TEF1 and PO outcome when the instance admits one. However, the following results show that this is not computationally tractable.

Theorem 4.3. *Determining whether there exists a* TEF1 *allocation that is* PO *for goods is* NP*-hard, even when $n = 2$.*

Theorem 4.4. *Determining whether there exists a* TEF1 *allocation that is* PO *for chores is* NP*-hard, even when $n = 2$.*

The proof of the above result essentially implies that even determining whether an instance admits a TEF1 and *utilitarian-maximizing* (i.e., sum of agents' utilities) allocation is computationally intractable, since a utilitarian-welfare maximizing allocation is necessarily PO. In fact, for the case of goods, we can make a stronger statement relating to the general class of *p-mean welfares*, defined as follows.[4]

Definition 4.5. Given $p \in (-\infty, 1]$ and an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ of goods, the *p-mean welfare* is $\left(\frac{1}{n}\sum_{i \in N} v_i(A_i)^p\right)^{1/p}$.

[4]Note that we cannot say the same for chores as when agents' valuations are negative, the p-mean welfare may be ill-defined.

In the context of fair division, p-mean welfare has been traditionally and well-studied for the setting with goods [14, 28], although it has recently been explored for chores as well [34]. Importantly, p-means welfare captures a spectrum of commonly studied fairness objectives in fair division. For instance, setting $p = 1$ (resp. $p = -\infty$) would correspond to the utilitarian (resp. egalitarian) welfare. Setting $p \to 0$ corresponds to maximizing the geometric mean, which is also known as the Nash welfare [26].

Then, from our construction in the proof of Theorem 4.3 (for goods), we have that an allocation is TEF1 and PO if and only if it also maximizes the p-mean welfare, for all $p \in (-\infty, 1]$, thereby giving us the following corollary.

Corollary 4.6. *For all $p \in (-\infty, 1]$, determining whether there exists a* TEF1 *allocation that maximizes p-mean welfare is* NP*-hard, even when $n = 2$.*

5 Multiple Items per Round

We now revisit the setting where multiple items may arrive at each round. While Lemma 3.1 reduces this case to the setting where a single item arrives per round, there are restricted variants of our problem that are not preserved by this reduction. We will now consider two such variants: $T = 2$ and repeated allocation.

We begin by showing that when there are two rounds, a TEF1 allocation can be computed efficiently.

Theorem 5.1. *When $T = 2$, a* TEF1 *allocation for goods or chores exists and can be computed in polynomial time.*

For the remainder of Section 5, we consider the *repeated* setting (also studied by Igarashi et al. [47] and Caragiannis and Narang [25]), where the sets $O_1, \dots, O_T$ are identical. Formally, for each $t \in T$ we have $O_t = \{o_1^t, \dots, o_k^t\}$, and $v_i(o_j^t) = v_i(o_j^r)$ for all $t, r \in [T]$ and all $i \in N$, $j \in [k]$. Note that this property of the instance is not preserved by our reduction from many items per round to a single item per round.

In general, it remains an open question whether a TEF1 allocation exists for this setting. However, we can show that, perhaps surprisingly, it is NP-hard to determine whether there exists a TEF1 allocation that allocates the items in the *same way* at *every round*. We say that an allocation $\mathcal{A}$ is *repetitive* if for each $i \in N$, $j \in [k]$ and all $t, r \in [T]$ we have $o_j^t \in A_i^t \setminus A_i^{t-1}$ if and only if $o_j^r \in A_i^r \setminus A_i^{r-1}$. Then we have the following result.

Theorem 5.2. *Determining whether there exists a repetitive allocation $\mathcal{A} = (A_1, \dots, A_n)$ that is* TEF1 *is* NP*-complete both for goods and for chores. The hardness result holds even if $T = 2$ and agents have identical valuations.*

Proof. It is immediate that this problem is in NP: we can guess a repetitive allocation, and check whether it is TEF1. Both for goods and for chores, we reduce from the NP-hard problem Multiway Number Partitioning [44]. An instance of this problem is given by a positive integer κ and a multiset $S = \{s_1, \dots, s_\mu\}$ of μ non-negative integers whose sum is κW; it is a yes-instance if S can be partitioned into κ subsets such that the sum of integers in each subset is W, and a no-instance otherwise.

Consider an instance of Multiway Number Partitioning given by a positive integer κ and a multiset $S = \{s_1, \dots, s_\mu\}$ of μ non-negative integers that sum up to κW.

We first prove the result for goods. We construct an instance with $\kappa + 1$ agents and $\mu + 1$ goods in each round: $O_1 = \{g_1^1, \dots, g_{\mu+1}^1\}$ and $O_2 = \{g_1^2, \dots, g_{\mu+1}^2\}$. The agents have an identical valuation function v defined as follows: $v(g_j^1) = v(g_j^2) = s_j$ if $j \in [\mu]$, and $v(g_{\mu+1}^1) = v(g_{\mu+1}^2) = 2W$. We will now prove that there exists a repetitive TEF1 allocation $\mathcal{A}$ if and only if the set S can be partitioned into κ subsets with equal sums (of W each).

For the 'if' direction, consider a κ-way partition $\mathcal{P} = \{P_1, \ldots, P_\kappa\}$ of S with $\sum_{s \in P_i} s = W$ for each $i \in [\kappa]$. We construct allocations $\mathcal{A}^1$ and $\mathcal{A}^2$ by allocating the goods corresponding to the elements of subset P_i to agent i for $i \in [\kappa]$; the goods $g^1_{\mu+1}$ and $g^2_{\mu+1}$ are allocated to agent $\kappa + 1$. Then, in $\mathcal{A}^1$, for each agent $i \in [\kappa]$ we have $v(A^1_i) = \sum_{s \in P_i} s = W$, and $v(A^1_{\kappa+1}) = v(g^1_{\mu+1}) = 2W$. It is easy to verify that $\mathcal{A}^1$ is EF1: no agent $i \in [\kappa]$ envies another agent $j \in [\kappa] \setminus \{i\}$, as they have the same bundle value, and agent i's envy towards agent $\kappa + 1$ can be removed by dropping $g^1_{\mu+1}$ from $A^1_{\kappa+1}$. Also, agent $\kappa + 1$ does not envy the first κ agents: she values her bundle at $2W$ and the bundles of $i \in [\kappa]$ at W.

Moreover in $\mathcal{A}^2$ each agent $i \in [\kappa]$ values the bundles $A^2_1, \ldots, A^2_\kappa$ at $2W$ and hence does not envy any of the first κ agents; her envy towards $\kappa + 1$ can be eliminated by dropping $g^1_{\mu+1}$ from $A^2_{\kappa+1}$. On the other hand, agent $\kappa + 1$ values her bundle at $4W$ and all other bundles at $2W$, so she does not envy the first κ agents.

For the 'only if' direction, suppose we have a repetitive allocation $\mathcal{A}^2$ that satisfies TEF1. Since agents have identical valuation functions, we can assume without loss of generality that agent $\kappa + 1$ receives goods $g^1_{\mu+1}$ and $g^2_{\mu+1}$ in rounds 1 and 2. For agent $i \in [\kappa]$ not to envy $\kappa+1$ in $\mathcal{A}^2$ after we drop one item from A^2_{k+1}, it has to be the case that $v_i(A^2_i) \geq 2W$. As this holds for all $i \in [\kappa]$ and $\sum_{j \in [\mu]} s_j = \kappa W$, this is only possible if there is a κ-way partition of S such that each subset sums up to W.

The proof for chores is similar, and can be found in the appendix. □

6 Conclusion

In this work, we studied the informed online fair division of indivisible items, with the goal of achieving TEF1 allocations. For both goods and chores, we showed the existence of TEF1 allocations in four special cases and provided polynomial-time algorithms for each case. Additionally, we showed that determining whether a TEF1 allocation exists for goods is NP-hard, and presented a similar, though slightly weaker, intractability result for chores. We further established the incompatibility between TEF1 and PO, which extends to an incompatibility with p-mean welfare. Finally, we explored the special case of multiple items arriving at each round.

Numerous potential directions remain for future work, including revisiting variants of the standard fair division model. Examples include studying the existence (and polynomial-time computability) of allocations satisfying a temporal variant of the weaker *proportionality up to one item* property (as defined by Conitzer et al. [30]), which would be implied by EF1; studying group fairness considerations in the temporal setting [2, 8, 18, 31, 49, 62]; considering the more general class of *submodular* valuations [43, 55, 66, 68]; considering the *house allocation* model where each agent gets a single item [29, 40], which was partially explored by Micheel and Wilczynski [54]; or even looking at more general settings with additional size constraints [16, 15, 36]. Another promising direction is to examine the number of approximate TEF1 allocations that exist in order to identify additional special cases [58, 64]. It would also be interesting to extend our results, which hold for the cases of goods and chores separately, to the more general case of mixed manna(see, e.g., [9]). In fact, with an appropriate modification of the instance, we can extend Theorem 3.2 to show that a TEF1 allocation exists in the mixed manna setting when there are two agents (see the appendix).

Acknowledgments

Most of this work was done when Edith Elkind was at the University of Oxford. Edith Elkind and Mohamad Latifian were supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grants EP/X038548/1 and EP/Y003624/1. We would like to thank the anonymous reviewers.

References

[1] Hannaneh Akrami, Rojin Rezvan, and Masoud Seddighin. An EF2X allocation protocol for restricted additive valuations. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 17–23, 2022.

[2] Martin Aleksandrov and Toby Walsh. Group envy freeness and group pareto efficiency in fair division with indivisible items. In *Proceedings of the 41st*, pages 57–72, 2018.

[3] Martin Aleksandrov and Toby Walsh. Online fair division: A survey. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13557–13562, 2020.

[4] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. Online fair division: Analysing a food bank problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2540–2546, 2015.

[5] Shiri Alouf-Heffetz, Laurent Bulteau, Edith Elkind, Nimrod Talmon, and Nicholas Teh. Better collective decisions via uncertainty reduction. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 24–30, 2022.

[6] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence*, 322:103965, 2023.

[7] Kenneth J. Arrow. *Social Choice and Individual Values.* Yale University Press, 2012.

[8] Haris Aziz and Simon Rey. Almost group envy-free allocation of indivisible goods and chores. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 39–45, 2020.

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

[10] Haris Aziz, Jeremy Lindsay, Angus Ritossa, and Mashbat Suzuki. Fair allocation of two types of chores. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 143–151, 2023.

[11] Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. Online nash social welfare maximization with predictions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–19, 2022.

[12] Siddhartha Banerjee, Vasilis Gkatzelis, Safwan Hossain, Billy Jin, Evi Micha, and Nisarg Shah. Proportionally fair online allocation of public goods with predictions. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 20–28, 2023.

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

[14] Siddharth Barman, Umang Bhaskar, Anand Krishna, and Ranjani G. Sundaram. Tight approximation algorithms for p-mean welfare under subadditive valuations. In *Proceedings of the 28th Annual European Symposium on Algorithms (ESA)*, pages 11:1–11:17, 2020.

[15] Siddharth Barman, Arindam Khan, Sudarshan Shyam, and K. V. N. Sreenivas. Guaranteeing envy-freeness under generalized assignment constraints. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, pages 242–269, 2023.

[16] Siddharth Barman, Arindam Khan, Sudarshan Shyam, and K.V.N. Sreenivas. Finding fair allocations under budget constraints. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5481–5489, 2023.

[17] Sanjoy K Baruah. Fairness in periodic real-time scheduling. In *Proceedings 16th IEEE Real-Time Systems Symposium*, pages 200–209. IEEE, 1995.

[18] Nawal Benabbou, Mithun Chakraborty, Edith Elkind, and Yair Zick. Fairness towards groups of agents in the allocation of indivisible items. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 95–101, 2019.

[19] Gerdus Benadè, Aleksandr M Kazachkov, Ariel D Procaccia, Alexandros Psomas, and David Zeng. Fair and efficient online allocations. *Operations Research*, 72(4):1438–1452, 2024.

[20] Ziyad Benomar and Vianney Perchet. Non-clairvoyant scheduling with partial predictions. *arXiv preprint arXiv:2405.01013*, 2024.

[21] Bhavook Bhardwaj, Rajnish Kumar, and Josue Ortega. Fairness and efficiency in cake-cutting with single-peaked preferences. *Economics Letters*, 190:109064, 2020.

[22] Duncan Black. On the rationale of group decision-making. *Journal of Political Economy*, 56(1): 23–34, 1948.

[23] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[24] Franklin Camacho, Rigoberto Fonseca-Delgado, Ramón Pino Pérez, and Guido Tapia. Generalized binary utility functions and fair allocations. *Mathematical Social Sciences*, 121:50–60, 2023.

[25] Ioannis Caragiannis and Shivika Narang. Repeatedly matching items to agents fairly and efficiently. *Theoretical Computer Science*, 981:114246, 2024.

[26] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation*, 7(3):12:1–12:32, 2019.

[27] Nikhil Chandak, Shashwat Goel, and Dominik Peters. Proportional aggregation of preferences for sequential decision making. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9573–9581, 2024.

[28] Bhaskar Ray Chaudhury, Jugal Garg, and Ruta Mehta. Fair and efficient allocations under subadditive valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5269–5276, 2021.

[29] Davin Choo, Yan Hao Ling, Warut Suksompong, Nicholas Teh, and Jian Zhang. Envy-free house allocation with minimum subsidy. *Operations Research Letters*, 54:107103, 2024.

[30] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 629–646, 2017.

[31] Vincent Conitzer, Rupert Freeman, Nisarg Shah, and Jennifer Vaughan. Group fairness for the allocation of indivisible goods. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1853–1860, 2019.

[32] Benjamin Cookson, Soroush Ebadian, and Nisarg Shah. Temporal fair division. *arXiv preprint arXiv:2410.23416*, 2024.

[33] Soroush Ebadian, Dominik Peters, and Nisarg Shah. How to fairly allocate easy and difficult chores. In *21st International Conference on Autonomous Agents and Multiagent Systems*, 2022.

[34] Owen Eckart, Alexandros Psomas, and Paritosh Verma. On the fairness of normalized p-means for allocating goods and chores. *arXiv preprint arXiv:2402.14996*, 2024.

[35] Edith Elkind, Sonja Kraiczy, and Nicholas Teh. Fairness in temporal slot assignment. In *Proceedings of the 15th International Symposium on Algorithmic Game Theory (SAGT)*, pages 490–507, 2022.

[36] Edith Elkind, Ayumi Igarashi, and Nicholas Teh. Fair division of chores with budget constraints. In *Proceedings of the 17th International Symposium on Algorithmic Game Theory (SAGT)*, pages 55–71, 2024.

[37] Edith Elkind, Tzeh Yuan Neoh, and Nicholas Teh. Temporal elections: Welfare, strategyproofness, and proportionality. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pages 3292–3299, 2024.

[38] Edith Elkind, Svetlana Obraztsova, and Nicholas Teh. Temporal fairness in multiwinner voting. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 22633–22640, 2024.

[39] Edith Elkind, Svetlana Obraztsova, and Nicholas Teh. Verifying proportionality in temporal multiwinner voting. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2246–2248, 2024.

[40] Jiarui Gan, Warut Suksompong, and Alexandros A. Voudouris. Envy-freeness in house allocation problems. *Mathematical Social Sciences*, 101:104–106, 2019.

[41] Jugal Garg, Aniket Murhekar, and John Qin. Fair and efficient allocations of chores under bivalued preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5043–5050, 2022.

[42] Jugal Garg, Aniket Murhekar, and John Qin. Weighted ef1 and po allocations with few types of agents or chores. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2799–2806, 2024.

[43] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Beyond additive valuations. *Artificial Intelligence*, 303: 103633, 2022.

[44] Ronald Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429, 1969.

[45] Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. Fair division with binary valuations: One rule to rule them all. In *Proceedings of the 16th International Workshop on Internet and Network Economics (WINE)*, pages 370–383, 2020.

[46] Jiafan He, Ariel Procaccia, Alexandros Psomas, and David Zeng. Achieving a fairer future by changing the past. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 343–349, 2019.

[47] Ayumi Igarashi, Martin Lackner, Oliviero Nardi, and Arianna Novaro. Repeated fair allocation of indivisible items. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9781–9789, 2024.

[48] Sungjin Im and Benjamin Moseley. Fair scheduling via iterative quasi-uniform sampling. *SIAM Journal on Computing*, 49(3):658–680, 2020.

[49] Maria Kyropoulou, Warut Suksompong, and Alexandros Voudouris. Almost envy-freeness in group resource allocation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 400–406, 2019.

[50] Martin Lackner. Perpetual voting: Fairness in long-term decision making. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2103–2110, 2020.

[51] Bo Li, Minming Li, and Ruilong Zhang. Fair scheduling for time-dependent resources. *Advances in Neural Information Processing Systems*, 34:21744–21756, 2021.

[52] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 125–131, 2004.

[53] Andrew Mackenzie and Vilmos Komornik. Fairly taking turns. *Games and Economic Behavior*, 142: 743–764, 2023.

[54] Karl Jochen Micheel and Anaëlle Wilczynski. Fairness in repeated house allocation. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2390–2392, 2024.

[55] Luisa Montanari, Ulrike Schmidt-Kraepelin, Warut Suksompong, and Nicholas Teh. Weighted envy-freeness for submodular valuations. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9865–9873, 2024.

[56] Hervé Moulin. *Fair Division and Collective Welfare.* MIT Press, 2003.

[57] Tzeh Yuan Neoh and Nicholas Teh. Welfare maximization in perpetual voting (student abstract). In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 23597–23599, 2024.

[58] Tzeh Yuan Neoh and Nicholas Teh. Understanding efx allocations: Counting and variants. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[59] Tzeh Yuan Neoh and Nicholas Teh. Strategic manipulation in temporal voting with undesirable candidates (student abstract). In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[60] Christos H. Papadimitriou. *Computational complexity.* Academic Internet Publ., 2007. ISBN 978-1-4288-1409-7.

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

[62] Jonathan Scarlett, Nicholas Teh, and Yair Zick. For one and all: Individual and group fairness in the allocation of indivisible goods. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2466–2468, 2023.

[63] Uwe Schwiegelshohn and Ramin Yahyapour. Fairness in parallel job scheduling. *Journal of Scheduling*, 3(5):297–320, 2000.

[64] Warut Suksompong. On the number of almost envy-free allocations. *Discrete Applied Mathematics*, 284:606–610, 2020.

[65] Warut Suksompong and Nicholas Teh. On maximum weighted nash welfare for binary valuations. *Mathematical Social Sciences*, 117:101–108, 2022.

[66] Warut Suksompong and Nicholas Teh. Weighted fair division with matroid-rank valuations: Monotonicity and strategyproofness. *Mathematical Social Sciences*, 126:48–59, 2023.

[67] William Thomson. Resource-monotonic solutions to the problem of fair division when preferences are single-peaked. *Social Choice and Welfare*, 11(3):205–223, 1994.

[68] Gilad Ben Uziahu and Uriel Feige. On fair allocation of indivisible goods to submodular agents. *arXiv preprint arXiv:2303.12444*, 2023.

[69] Valentin Zech, Niclas Boehmer, Edith Elkind, and Nicholas Teh. Multiwinner temporal voting with aversion to change. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, 2024.

Appendix

A Counterexample for Goods when $n \geq 3$

As mentioned in the main text, He et al. [46, Thm. 4.2] used the following counterexample to show that a TEF1 allocation may not exist for goods when $n = 3$. Note that in this counterexample, one good arrives at each round.

v	g_1	g_2	g_3	g_{4-6}	g_7	g_8	g_9	g_{10}	g_{11-12}	g_{13}	g_{14-17}	g_{18-19}	g_{20}	g_{21-22}	g_{23}
1	0.9	0.8	0.7	1	0.15	100	110	120	200	200	200	200	200	200	200
2	0.9	0.7	0.8	1	0.95	100	110	120	200	200	200	120	200	120	200
3	0.8	0.9	0.7	1	0.25	100	110	120	200	185	200	200	200	200	200

For completeness, we briefly explain the counterexample. There are three parts to this example, which ultimately ensure that in any TEF1 allocation, after g_{22} is allocated, agent 2 envies both other agents, and one other agent envies agent 2. As a result, g_{23} cannot be allocated to any agent without violating TEF1.

The first part consists of goods g_1 to g_7. The instance is constructed such that after all of the goods in this part have been allocated in a TEF1 manner, the possible envy relations are restricted. Specifically, we have that after round 7, agent 3 cannot envy agent 1, and that agent 2 cannot envy agents 1 or 3.

The second part consists of goods g_8 to g_{16}, and builds on top of the previous envy restriction to ensure that after round 16, agent 2 is envied by either agent 1 or agent 3 in any TEF1 allocation.

The final part consists of goods g_{17} to g_{23}. Since agent 2 is envied by some agent at the start, it cannot receive good g_{17} or g_{20}, and must receive one of $\{g_{18}, g_{19}\}$ and one of $\{g_{21}, g_{22}\}$. This causes agent 2 to envy both other agents, while one of the other agents continues to envy agent 2. Therefore, TEF1 will be violated regardless of which agent receives g_{23}.

Note that this example cannot be modified to act as a counterexample for chores. We have found that in the first part, we cannot sufficiently restrict the possible envy relations. This is due to the fundamental difference in allocating goods and chores: goods cannot be allocated to an agent which is envied, whilst chores cannot be allocated to an agent which envies others.

B Omitted Proofs from Section 3

B.1 Proof of Theorem 3.2

The polynomial runtime of Algorithm 1 is easy to verify: there is only one **for** loop, with a counter that runs from 1 to m, and each operation within the loop runs in $\mathcal{O}(m)$ time. Thus, we focus on proving correctness.

For each $t \in [m]$, we define $r_t < t$ as the latest round before t such that $\mathcal{A}^{r_t}$ is EF. This implies that if $\mathcal{A}^\ell \setminus \mathcal{A}^{r_t}$ is EF1 for all $\ell = r_t, r_t + 1, \dots, t$, then $\mathcal{A}^t$ is also EF1. Therefore, it suffices to show that $\mathcal{A}^t \setminus \mathcal{A}^{r_t}$ is EF1 for each $t \in [m]$. We will prove this by induction on t.

For $t = 1$, the claim is immediate, as any allocation of a single chore is EF1. Now, suppose that $t > 1$. If $t = r_t + 1$ the allocation $\mathcal{A}^t \setminus \mathcal{A}^{r_t}$ consists of a single chore, so, again, the claim is immediate. Otherwise, $r_{t-1} = r_t$ and by the induction hypothesis it holds that $\mathcal{A}^{t-1} \setminus \mathcal{A}^{r_t}$ is EF1. Let r'_t be the earliest round ahead of r_t such that $\mathcal{A}^{r'_t} \setminus \mathcal{A}^{r_t}$ is EF (if such a round exists). We divide the remainder of the proof into two cases depending on whether a partial bundle swap (as in line 9 of the algorithm) occurs at round r'_t.

Case 1: Round r'_t does not exist or no swap at round r'_t. Suppose without loss of generality that $v_1(A_1^{t-1} \setminus A_1^{r_t}) < v_1(A_2^{t-1} \setminus A_2^{r_t})$, i.e., agent 1 envies agent 2 in $\mathcal{A}^{t-1} \setminus \mathcal{A}^{r_t}$. Then agent 2 does not envy agent 1 (otherwise we would swap the bundles, contradicting the definition of r_t), and consequently receives c_t. If agent 2 envies agent 1 after receiving c_t, this envy can be removed by removing c_t. We also know that $\mathcal{A}^t \setminus \mathcal{A}^{r_t}$ is EF1 w.r.t. agent 1 (who did not receive a chore in round t) because by our inductive assumption, $\mathcal{A}^{t-1} \setminus \mathcal{A}^{r_t}$ is EF1, concluding the proof of this case.

Case 2: Swap occurs at round r'_t. We assume that $r'_t > t$, because if $r'_t = t$, then $\mathcal{A}^t \setminus \mathcal{A}^{r_t}$ is EF and therefore EF1. For each $i \in \{t-1, t\}$, let $\mathcal{B}^i \setminus \mathcal{A}^s$ refer to the algorithm's allocation of the chores $O^i \setminus O^s$ *before* the bundle swap, and suppose without loss of generality that $v_1(B_1^{t-1} \setminus B_1^{r_t}) < v_1(B_2^{t-1} \setminus B_2^{r_t})$. We therefore must have $v_2(B_2^{t-1} \setminus B_2^{r_t}) \geq v_2(B_1^{t-1} \setminus B_1^{r_t})$ to avoid contradicting the definition of r_t. Since agent 2 is not envied by agent 1 in round $t-1$, it receives chore c_t, so we have $\mathcal{B}^t \setminus \mathcal{B}^{r_t} = (B_1^{t-1} \setminus B_1^{r_t}, (B_2^{t-1} \setminus B_2^{r_t}) \cup \{c_t\})$. This means that after the bundle swap is executed, we have $\mathcal{A}^t \setminus \mathcal{A}^{r_t} = ((B_2^{t-1} \setminus B_2^{r_t}) \cup \{c_t\}, B_1^{t-1} \setminus B_1^{r_t})$. Recall that $v_1(B_2^{t-1} \setminus B_2^{r_t}) > v_1(B_1^{t-1} \setminus B_1^{r_t})$, so c_t can be removed from agent 1's bundle to eliminate their envy towards agent 2. Also, by the inductive assumption, there exists a chore $c \in A_2^{t-1} \setminus A_2^{r_t}$ such that $v_2((A_2^{t-1} \setminus A_2^{r_t}) \setminus \{c\}) \geq v_2(A_1^{t-1} \setminus A_1^{r_t})$. Observe that $A_2^t \setminus A_2^{r_t} = A_2^{t-1} \setminus A_2^{r_t}$ and $A_1^t \setminus A_1^{r_t} = (A_1^{t-1} \setminus A_1^{r_t}) \cup \{c_t\}$. Combining this with the inductive assumption, we have that there exists a chore $c \in A_2^t \setminus A_2^{r_t}$ such that

$$\begin{aligned} v_2((A_2^t \setminus A_2^{r_t}) \setminus \{c\}) &= v_2((A_2^{t-1} \setminus A_2^{r_t}) \setminus \{c\}) \\ &\geq v_2(A_1^{t-1} \setminus A_1^{r_t}) \\ &\geq v_2((A_1^{t-1} \setminus A_1^{r_t}) \cup \{c_t\}) \\ &= v_2(A_1^t \setminus A_1^{r_t}). \end{aligned}$$

Therefore, $\mathcal{A}^t \setminus \mathcal{A}^{r_t}$ is EF1 in this case.

We have shown that $\mathcal{A}^t \setminus \mathcal{A}^{r_t}$ is EF1 regardless of whether the allocation has undergone a bundle swap, so by induction, Algorithm 1 returns a TEF1 allocation for chores.

B.2 Proof of Proposition 3.5

We first prove the result for the case of goods. Consider the instance with two agents $N = \{1, 2\}$ and three goods $O = \{g_1, g_2, g_3\}$, where agents have identical valuations: $v(g_1) = v(g_2) = 1$ and $v(g_3) = 2$. In order for the partial allocation at the end of the second round to be TEFX, each agent must be allocated exactly one of $\{g_1, g_2\}$—suppose that agent 1 is allocated g_1 and agent 2 is allocated g_2. In the third round, without loss of generality, suppose that g_3 is allocated to agent 1. Then, agent 2 will still envy agent 1 even after dropping g_1 from agent 1's bundle, as $v(A_2) = v(g_2) = 1 < v(A_1 \setminus \{g_1\}) = v(g_3) = 2$.

Next, we prove the result for chores. Consider the instance with two agents $N = \{1, 2\}$ and three chores $O = \{c_1, c_2, c_3\}$, where agents have identical valuations: $v(c_1) = v(c_2) = -1$ and $v(c_3) = -2$. In order for the partial allocation at the end of the secound round to be TEFX, each agent must be allocated exactly one of $\{c_1, c_2\}$—suppose that agent 1 is allocated c_1 and agent 2 is allocated c_2. In the third round, without loss of generality, suppose that c_3 is allocated to agent 1. Then, agent 1 will still envy agent 2 even after dropping c_1 from her own bundle, as $v(A_1 \setminus \{c_1\}) = v(c_3) = -2 < v(A_2) = v(c_2) = -1$.

B.3 Proof of Theorem 3.6

Consider the following greedy algorithm (Algorithm 2).

The polynomial runtime of the Algorithm 2 is easy to verify: there is only one **for** loop which runs in $\mathcal{O}(m)$ time, and the other operations within run in $\mathcal{O}(mn)$ time. Thus, we focus on proving correctness.

Intuitively, α and β each keep a counter of which agent should be next allocated an item of type T_1 and T_2, respectively. For this reason, for each $r \in \{1, 2\}$, we can observe that with respect to only items

Algorithm 2 Returns a TEF1 allocation for goods or chores when there are two types of items

Input: Set of agents $N = \{1, \ldots, n\}$, set of items $O = \{o_1, \ldots, o_m\}$, and valuation profile $\mathbf{v} = (v_1, \ldots, v_n)$
Output: TEF1 allocation $\mathcal{A}$ of items in O to agents in N

Initialize $\alpha \leftarrow 1$, $\beta \leftarrow n$, and $\mathcal{A}^0 \leftarrow (\varnothing, \ldots, \varnothing)$
for $t = 1, 2, \ldots, m$ **do**
  if $\alpha = n + 1$ **then**
    $\alpha \leftarrow 1$
  else if $\beta = 0$ **then**
    $\beta \leftarrow n$
  end if
  if $o_t \in T_1$ **then**
    $A^t_\alpha \leftarrow A^{t-1}_\alpha \cup \{o_t\}$, $A^t_j \leftarrow A^{t-1}_j$ for all $j \in N \setminus \{\alpha\}$, and $\alpha \leftarrow \alpha + 1$
  else
    $A^t_\beta \leftarrow A^{t-1}_\beta \cup \{o_t\}$, $A^t_j \leftarrow A^{t-1}_j$ for all $j \in N \setminus \{\beta\}$, and $\beta \leftarrow \beta - 1$
  end if
end for
return $\mathcal{A} = (A^m_1, \ldots, A^m_n)$

of type T_r, the algorithm allocates these items in a round-robin fashion. We can therefore make the following two observations:

(i) for any pair of agents $i, j \in N$, if $|A^t_i \cap T_1| > |A^t_j \cap T_1|$, then $i < j$; if $|A^t_i \cap T_2| > |A^t_j \cap T_2|$, then $i > j$; and

(ii) for any pair of agents $i, j \in N$, round $t \in [m]$, and $r \in \{1, 2\}$, we have that $\Big||A^t_i \cap T_r| - |A^t_j \cap T_r|\Big| \leq 1$.

The first observation follows from fact that the α counter is increasing in agent indices whereas the β counter is decreasing in agent indices. The second observation follows from the widely-known fact that, with respect to items of a specific type, a round-robin allocation always returns a balanced allocation, i.e., the bundle sizes of any two agents differ by no more than one.

Next, we have that for any two agents $i, j \in N$, round $t \in [m]$, and $r, r' \in \{1, 2\}$ where $r \neq r'$, if $|A^t_i \cap T_r| > |A^t_j \cap T_r|$, then $|A^t_i \cap T_{r'}| \leq |A^t_j \cap T_{r'}|$. To see this, suppose for a contradiction that there exists agents $i, j \in N$ and round $t \in [m]$ such that both $|A^t_i \cap T_1| > |A^t_j \cap T_1|$ and $|A^t_i \cap T_2| > |A^t_j \cap T_2|$. Then, observation (i) will give us $i < j$ and $i > j$ respectively, a contradiction.

For the case of goods, we have that for any pair of agents $i, j \in N$ and round $t \in [m]$, if $i < j$, then

$$v_i(A^t_i \cap T_1) \geq v_i(A^t_j \cap T_1) \tag{2}$$

because i precedes j in the round-robin allocation order, and by the well-established EF1 property of the round-robin algorithm for goods, there exists a good $g \in A^t_j \cap T_2$ such that

$$v_i(A^t_i \cap T_2) \geq v_i(A^t_j \cap T_2 \setminus \{g\}). \tag{3}$$

Combining (2) and (3), there exists a good $g \in A^t_j$ such that

$$v_i(A^t_i) = v_i(A^t_i \cap T_1) + v_i(A^t_i \cap T_2) \geq v_i(A^t_j \cap T_1) + v_i(A^t_j \cap T_2 \setminus \{g\}) = v_i(A^t_j \setminus \{g\}).$$

Moreover, if $i > j$, then

$$v_i(A_i^t \cap T_2) \geq v_i(A_i^t \cap T_2), \tag{4}$$

and there exists a good $g \in A_j^t \cap T_1$ such that

$$v_i(A_i^t \cap T_1) \geq v_i(A_j^t \cap T_1 \setminus \{g\}). \tag{5}$$

Combining (4) and (5), there exists a good $g \in A_j^t$ such that

$$v_i(A_i^t) = v_i(A_i^t \cap T_1) + v_i(A_i^t \cap T_2) \geq v_i(A_j^t \cap T_1 \setminus \{g\}) + v_i(A_j^t \cap T_2) = v_i(A_j^t \setminus \{g\}).$$

For the case of chores, we have that for any pair of agents $i, j \in N$ and round $t \in [m]$, if $i > j$, then

$$v_i(A_i^t \cap T_1) \geq v_i(A_j^t \cap T_1), \tag{6}$$

and again by the EF1 property of the round-robin algorithm for chores [9], there exists a chore $c \in A_i^t \cap T_2$ such that

$$v_i(A_i^t \cap T_2 \setminus \{c\}) \geq v_i(A_j^t \cap T_2). \tag{7}$$

Combining (6) and (7), there exists a chore $c \in A_i^t$ such that

$$v_i(A^t \setminus \{c\}) = v_i(A_i^t \cap T_1) + v_i(A_i^t \cap T_2 \setminus \{c\}) \geq v_i(A_j^t \cap T_1) + v_i(A_j^t \cap T_2) = v_i(A_j^t).$$

Moreover, if $i < j$, then

$$v_i(A_i^t \cap T_2) \geq v_i(A_j^t \cap T_2), \tag{8}$$

and there exists a chore $c \in A_i^t \cap T_1$ such that

$$v_i(A_i^t \cap T_1 \setminus \{c\}) \geq v_i(A_j^t \cap T_1). \tag{9}$$

Combining (8) and (9), there exists a chore $c \in A_i^t$ such that

$$v_i(A^t \setminus \{c\}) = v_i(A_i^t \cap T_1 \setminus \{c\}) + v_i(A_i^t \cap T_2) \geq v_i(A_j^t \cap T_1) + v_i(A_j^t \cap T_2) = v_i(A_j^t).$$

Thus, our result holds.

B.4 Proof of Theorem 3.7

We first prove the result for goods. Consider the following greedy algorithm (Algorithm 3) which iterates through the rounds and allocates each good to the agent who has the least value for their bundle.

We first show that for any $i, j \in N$ and $t \in [m]$, it holds that $v_i(A_i^t) \geq v_j(A_i^t)$. Suppose for a contradiction that there exists some $i, j \in N$ and $t \in [m]$ such that $v_i(A_i^t) < v_j(A_i^t)$. This means there exists some good $g \in A_i^t$ whereby $v_i(g) = 0$ and $v_j(g) > 0$. However, then the algorithm would not have allocated g to i, a contradiction.

Next, we will prove by induction that for every $t \in [m]$, $\mathcal{A}^t$ is TEF1. The base case is trivially true: when $t = 1$, if every agent values g_1 at 0, then allocating it to any agent will satisfy TEF1, whereas if some agent values g_1, allocating it to any agent will also be TEF1: the envy by any other agent towards this agent will disappear with the removal of g_1 from the agent's bundle (every agent's bundle will then be the empty set).

Then, we prove the inductive step. Assume that for some $k \in [m-1]$, $\mathcal{A}^k$ is TEF1. We will show that $\mathcal{A}^{k+1}$ is also TEF1. Due to the assumption, it suffices to show that for all $i, j \in N$, there exists a good $g \in A_j^{k+1}$ such that $v_i(A_i^{k+1}) \geq v_i(A_j^{k+1} \setminus \{g\})$. Consider the agent $i \in N$ that is allocated g_{k+1}.

Algorithm 3 Returns a TEF1 allocation of goods under generalized binary valuations

Input: Set of agents $N = \{1, \ldots, n\}$, set of goods $O = \{g_1, \ldots, g_m\}$, and valuation profile $\mathbf{v} = (v_1, \ldots, v_n)$
Output: TEF1 allocation of goods $\mathcal{A}$ in O to agents in N

Initialize the empty allocation $\mathcal{A}^0$ where $A_i^0 = \varnothing$ for all $i \in N$.
for $t = 1, 2, \ldots, m$ **do**
  Let $S := \{i' \in N \mid v_{i'}(g_t) > 0\}$
  if $S = \varnothing$ **then**
    Let i be any agent in N
  else
    Let $i \in \arg\min_{i' \in S} v_{i'}(A_{i'}^{t-1})$, with ties broken arbitrarily
  end if
  $A_i^t \leftarrow A_i^{t-1} \cup \{g_t\}$ and $A_j^t \leftarrow A_j^{t-1}$ for all $j \in N \setminus \{i\}$
end for
return $\mathcal{A} = (A_1^m, \ldots, A_n^m)$

We first show agent i must be unenvied before being allocated g_{k+1}. Suppose towards a contradiction this is not the case, i.e., there exists some other agent $j \neq i$ whereby $v_j(A_j^k) < v_j(A_i^k)$. Together with the fact that $v_j(A_i^k) \leq v_i(A_i^k)$ from the result above, we get that

$$v_j(A_j^k) < v_j(A_i^k) \leq v_i(A_i^k),$$

contradicting the fact that i is an agent with the minimum bundle value and thus chosen by the algorithm to receive g_{k+1}. As such, i must be unenvied before being allocated g_{k+1}, i.e., for any other agent $j \in N \setminus \{i\}$, we have that $v_j(A_j^k) \geq v_j(A_i^k)$.

Consequently, we get that

$$v_j(A_j^{k+1}) = v_j(A_j^k) \geq v_j(A_i^k) = v_j(A_i^{k+1} \setminus \{g_{k+1}\}).$$

Thus, by induction, the result holds.

Next, we prove the result for chores. Consider the following greedy algorithm (Algorithm 4) which iterates through the rounds, allocating each chore to an agent with zero value for it if possible, and otherwise, allocates the chore to an agent who does not envy any other agent.

We first show that for any $i, j \in N$ and $t \in [m]$, it holds that

$$v_i(A_i^t) \geq v_j(A_i^t). \tag{10}$$

Suppose for a contradiction that there exists some $i, j \in N$ and $t \in [m]$ such that $v_i(A_i^t) < v_j(A_i^t)$. This means there exists some chore $c \in A_i^t$ whereby $v_i(c) < 0$ and $v_j(c) = 0$. However, then the algorithm would not have allocated c to i, a contradiction.

Next, we will prove by induction that for every $t \in [m]$, $\mathcal{A}^t$ is TEF1. The base case is trivially true: when $t = 1$, if there exists an agent that values c_1 at 0, then allocating it to any such agent will satisfy TEF1, whereas if all agents values c_1 negatively, allocating it to any agent will also be TEF1: the envy by this agent towards any other agent will disappear with the removal of c_1 from the former agent's bundle (every agent's bundle will then be the empty set).

Then, we prove the inductive step. Assume that for some $k \in [m-1]$, $\mathcal{A}^k$ is TEF1. We will show that $\mathcal{A}^{k+1}$ is also TEF1, i.e., for all $i, j \in N$, there exists a chore $c \in A_i^{k+1}$ such that $v_i(A_i^{k+1} \setminus \{c\}) \geq v_i(A_j^{k+1})$.

Algorithm 4 Returns an TEF1 allocation of chores under generalized binary valuations

Input: Set of agents $N = \{1, \ldots, n\}$, set of chores $O = \{c_1, \ldots, c_m\}$, and valuation profile $\mathbf{v} = (v_1, \ldots, v_n)$
Output: TEF1 allocation of chores $\mathcal{A}$ in O to agents in N

Initialize the empty allocation $\mathcal{A}^0$ where $A_i^0 = \varnothing$ for all $i \in N$.
for $t = 1, 2, \ldots, m$ **do**
  if there exists an agent $i \in N$ such that $v_i(c_t) = 0$ **then**
    Let $i \in \{i' \in N \mid v_{i'}(c_t) = 0\}$
  else
    Let $i \in \arg\max_{i' \in N} v_{i'}(A_{i'}^{t-1})$
  end if
  $A_i^t \leftarrow A_i^{t-1} \cup \{c_t\}$ and $A_j^t \leftarrow A_j^{t-1}$ for all $j \in N \setminus \{i\}$
end for
return $\mathcal{A} = (A_1^m, \ldots, A_n^m)$

Suppose agent i is allocated the chore c_{k+1}. If $v_i(c_{k+1}) = 0$, then each agents' valuation for every other agent's bundle (including his own) remains the same, and thus $\mathcal{A}^{k+1}$ remains TEF1. If $v_i(c_{k+1}) < 0$, then we know that $v_j(c_{k+1}) < 0$ for all $j \in N$. We then proceed to show that agent i must not envy any other agent before being allocated c_{k+1}. Suppose for contradiction this is not the case, i.e., that there exists some other agent $j \neq i$ whereby $v_i(A_i^k) < v_i(A_j^k)$. Since c_{k+1} is allocated to the agent with the highest bundle, we have that $v_i(A_i^k) \geq v_j(A_j^k)$, and therefore

$$v_i(A_j^k) > v_i(A_i^k) \geq v_j(A_j^k).$$

However, this contradicts (10).

Since agent i does not envy another agent before being allocated c_{k+1}, we get that for any $j \neq i$,

$$v_i(A_i^{k+1} \setminus \{c_{k+1}\}) = v_i(A_i^k) \geq v_i(A_j^k) = v_i(A_j^{k+1}) \quad \text{and} \quad v_j(A_j^{k+1}) = v_j(A_j^k).$$

Thus, by induction, we get that $\mathcal{A}^{t+1}$ is TEF1.

B.5 Proof of Theorem 3.10

Consider the following greedy algorithm (Algorithm 5). Note that the same algorithm works for both settings for goods when valuations are single-peaked, and for chores when valuations are single-dipped.

Algorithm 5 Returns a TEF1 allocation for goods when valuations are single-peaked and chores when valuations are single-dipped

Input: Set of agents $N = \{1, \ldots, n\}$, set of items $O = \{o_1, \ldots, o_m\}$, and valuation profile $\mathbf{v} = (v_1, \ldots, v_n)$
Output: TEF1 allocation $\mathcal{A}$ of items in O to agents in N

Initialize $\mathcal{A}^0 \leftarrow (\varnothing, \ldots, \varnothing)$
for $t = 1, 2, \ldots, m$ **do**
  Let $i := \arg\min_{i \in N} |A_i^{t-1}|$, with ties broken lexicographically
  $A_i^t \leftarrow A_i^{t-1} \cup \{g_t\}$ and $A_i^t \leftarrow A_i^{t-1}$
end for
return $\mathcal{A} = (A_1^m, \ldots, A_n^m)$

The polynomial runtime of the Algorithm 5 is easy to verify: there is only one **for** loop which runs in $\mathcal{O}(m)$ time, and the other operations within run in $\mathcal{O}(n)$ time. Thus, we focus on proving correctness.

We first prove the case for goods, when valuations are single-peaked.

For each $i \in [m]$, let $g_i = o_i$, and thus $O = \{g_1, \dots, g_m\}$. We can assume that $m = \alpha n$ for some $\alpha \in \mathbb{Z}_{>0}$; otherwise we can simply add dummy goods to O until that condition is fulfilled. Then, Algorithm 5 will return $\mathcal{A}$, where for each $i \in N$, $A_i = \{g_i, g_{i+n}, \dots, g_{i+(\alpha-1)n}\}$.

For each $i \in N$ and $j \in [\alpha]$, let

- $T_j := \{g_{(j-1)n+1}, g_{(j-1)n+2}, \dots, g_{jn}\}$,
- $g'_{i,j} \in A_i \cap T_j$ be the unique good in T_j that was allocated to agent i
- $g^* := \arg\max_{g \in O} v_i(g)$ (with ties broken arbitrarily), and $g^* \in T_{i^*}$ for some $i^* \in [\alpha]$.

Then, we will show that for all $r \in [\alpha]$, $v_i(A_i^r) \geq v_i(A_j^r \setminus \{g\})$ for some $g \in A_j^r$. We split our analysis into two cases.

Case 1: $i < j$. If $r < i^*$, then since agent i's valuation for each subsequent good up to round T_r is non-decreasing, we have that for all $k \in \{2, \dots, r\}$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k-1}).$$

Consequently, we get that

$$v_i(A_i^r) \geq \sum_{k=2}^{r} v_i(g'_{i,k}) \geq \sum_{k=2}^{r} v_i(g'_{j,k-1}) = v_i(A_j^r \setminus \{g'_{j,r}\}).$$

If $r \geq i^*$, then we split our analysis into two further cases.

Case 1(a): g'_{i,i^*} appears before g^*. Then, for all $k \in \{2, \dots, i^*\}$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k-1}),$$

and for all $k \in \{i^* + 1, \dots, r\}$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k}).$$

Consequently, we get that

$$v_i(A_i^r) \geq \sum_{k=2}^{i^*} v_i(g'_{i,k}) + \sum_{k=i^*+1}^{r} v_i(g'_{i,k}) \geq \sum_{k=2}^{i^*} v_i(g'_{j,k-1}) + \sum_{k=i^*+1}^{r} v_i(g'_{j,k}) = v_i(A_j^r \setminus \{g'_{j,i^*}\}).$$

Case 1(b): g'_{i,i^*} appears after (or is) g^*. Then, for all $k \in \{2, \dots, i^* - 1\}$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k-1}),$$

and for all $k \in \{i^*, \dots, r\}$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k}).$$

Consequently, we get that

$$v_i(A_i^r) \geq \sum_{k=2}^{i^*-1} v_i(g'_{i,k}) + \sum_{k=i^*}^{r} v_i(g'_{i,k}) \geq \sum_{k=2}^{i^*-1} v_i(g'_{j,k-1}) + \sum_{k=i^*}^{r} v_i(g'_{j,k}) = v_i(A_j^r \setminus \{g'_{j,i^*-1}\}).$$

Case 2: $i > j$. If $r \leq i^*$, then since agent i's valuation for each subsequent good up to round T_r is nondecreasing, we have that for all $k \in [r]$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k}). \tag{11}$$

Consequently, we get that

$$v_i(A_i^r) \geq \sum_{k \in [r-1]} v_i(g'_{i,k}) \geq \sum_{k \in [r-1]} v_i(g'_{j,k}) \quad \text{(by (11))} = v_i(A_j^r \setminus \{g'_{j,r}\}).$$

If $r > i^*$, then we split our analysis into two further cases.

Case 2(a): g'_{i,i^*} **appears before (or is)** g^*. Then for all $k \in [i^*]$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k}),$$

and for all $k \in \{i^* + 1, \dots, r - 1\}$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k+1}).$$

Consequently, we get that

$$v_i(A_i^r) \geq \sum_{k \in [i^*]} v_i(g'_{i,k}) + \sum_{k=i^*+1}^{r-1} v_i(g'_{i,k}) \geq \sum_{k \in [i^*]} v_i(g'_{j,k}) + \sum_{k=i^*+1}^{r-1} v_i(g'_{j,k+1}) = v_i(A_j^r \setminus \{g'_{j,i^*+1}\}).$$

Case 2(b): g'_{i,i^*} **appears after** g^*. Then, for all $k \in [i^* - 1]$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k}),$$

and for all $k \in \{i^*, \dots, r - 1\}$,

$$v_i(g'_{i,k}) \geq v_i(g'_{j,k+1}).$$

Consequently, we get that

$$v_i(A_i^r) \geq \sum_{k \in [i^*-1]} v_i(g'_{i,k}) + \sum_{k=i^*}^{r-1} v_i(g'_{i,k}) \geq \sum_{k \in [i^*-1]} v_i(g'_{j,k}) + \sum_{k=i^*}^{r-1} v_i(g'_{j,k+1}) = v_i(A_j^r \setminus \{g'_{j,i^*}\}).$$

Thus, our result follows.

Next, we prove the case for chores, when valuations are single-dipped.

For each $j \in [m]$, let $o_i = c_i$, and thus $O = \{c_1, \dots, c_m\}$. We can assume that $m = \alpha n$ for some $\alpha \in \mathbb{Z}_{>0}$; otherwise we can simply add dummy chores to O until that condition is fulfilled. Then, Algorithm 5 will return $\mathcal{A}$, where for each $i \in N$, $A_i = \{c_i, c_{i+n}, \dots, c_{i+(\alpha-1)n}\}$.

For each $i \in N$ and $j \in [\alpha]$, let

- $T_j := \{c_{(j-1)n+1}, c_{(j-1)n+2}, \dots, c_{jn}\}$,
- $c'_{i,j} \in A_i \cap T_j$ be the unique chore in T_j that was allocated to agent i
- $c^* := \arg\min_{c \in O} v_i(c)$ (with ties broken arbitrarily), and $c^* \in T_{i^*}$ for some $i^* \in [\alpha]$.

Case 1: $i < j$**.** If $r \leq j^*$, then since agent i's valuation for each subsequent chore up to round T_{r-1} is nonincreasing, we have that for all $k \in [r-1]$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k}).$$

Consequently, we get that

$$v_i(A_i^r \setminus \{c'_{i,r}\}) = \sum_{k\in[r-1]} v_i(c'_{i,k}) \geq \sum_{k\in[r-1]} v_i(c'_{j,k}) \geq v_i(A_j^r).$$

If $r > j^*$, then we split our analysis into two further cases.

Case 1(a): c'_{j,j^*} **appears before (or is)** c^***.** Then for all $k \in [j^*]$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k})$$

and for all $k \in \{j^*+2, \dots, r\}$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k-1}).$$

Consequently, we get that

$$v_i(A_i^r \setminus \{c'_{i,j^*+1}\}) = \sum_{k\in[j^*]} v_i(c'_{i,k}) + \sum_{k=j^*+2}^{r} v_i(c'_{i,k}) \geq \sum_{k\in[j^*]} v_i(c'_{j,k}) + \sum_{k=j^*+2}^{r} v_i(c'_{j,k-1}) \geq v_i(A_j^r).$$

Case 1(b): c'_{j,j^*} **appears after** c^***.** Then for all $k \in [j^*-1]$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k})$$

and for all $k \in \{j^*+1, \dots, r\}$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k-1}).$$

Consequently, we get that

$$v_i(A_i^r \setminus \{c'_{i,j^*}) = \sum_{k\in[1,j^*-1]} v_i(c'_{i,k}) + \sum_{k=j^*+1}^{r} v_i(c'_{i,k}) \geq \sum_{k\in[j^*-1]} v_i(c'_{j,k}) + \sum_{k=j^*+1}^{r} v_i(c'_{j,k-1}) \geq v_i(A_j^r).$$

Case 2: $j < i$**.** If $r < j^*$, then since agent i's valuation for each subsequent chore up to round T_r is nondecreasing, we have that for all $k \in [r-1]$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k+1}).$$

Consequently, we get that

$$v_i(A_i^r \setminus \{c'_{i,r}\}) = \sum_{k\in[r-1]} v_i(c'_{i,k}) \geq \sum_{k\in[r-1]} v_i(c'_{j,k+1}) \geq v_i(A_j^r).$$

If $r \geq j^*$, then we split our analysis into two further cases.

Case 2(a): c_{j,j^*} **appears before (or is)** c^***.** Then for all $k \in [j^*-1]$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k+1})$$

and for all $k \in \{j^*+1, \dots, r\}$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k}).$$

Consequently, we get

$$v_i(A_i^r \setminus \{c'_{i,j^*}\}) = \sum_{k\in[j^*-1]} v_i(c'_{i,k}) + \sum_{k=j^*+1}^{r} v_i(c'_{i,k}) \geq \sum_{k\in[j^*-1]} v_i(c'_{j,k+1}) + \sum_{k=j^*+1}^{r} v_i(c'_{k,j}) \geq v_i(A_j^r).$$

Case 2(b): c_{j,j^*} appears after c^*. Then for all $k \in [j^* - 2]$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k+1}$$

and for all $k \in \{j^*, \ldots, r\}$,

$$v_i(c'_{i,k}) \geq v_i(c'_{j,k}).$$

Consequently, we get that

$$v_i(A_i^r \backslash \{c'_{i,j^*-1}\}) = \sum_{k \in [j^*-2]} v_i(c'_{i,k}) + \sum_{k=j^*}^{r} v_i(c'_{i,k}) \geq \sum_{k \in [j^*-2]} v_i(c'_{j,k+1}) + \sum_{k=j^*}^{r} v_i(c'_{j,k}) \geq v_i(A_j^r).$$

Thus, our result follows.

B.6 Proof of Lemma 3.13

```
from itertools import combinations
from copy import deepcopy

# If there exists a partial allocation for first 2n+2 rounds such that bundle valuations are equal
if_some_envy_exists = False

def is_ef1(allocation, agents, valuations, if_some_envy_exists, partial_alloc_envy_from, partial_alloc_envy_to):
    """
    Check if the current allocation is EF1.

    Parameters:
    - allocation: List of lists, where allocation[i] is the list of goods allocated to agent i.
    - agents: List of agent identifiers.
    - valuations: Dictionary where valuations[agent][good] gives the value of a good for an agent.
    - if_some_envy_exists: If the partial allocation for the first 2n+2 rounds is EF (i.e., equal bundle values)
    - partial_alloc_envy_from: If if_some_envy_exists is True, then which agent envies
    - partial_alloc_envy_to: If if_some_envy_exists is True, then which agent is being envied

    Returns:
    - True if allocation is EF1, False otherwise.
    """
    num_agents = len(agents)

    # Compute the value each agent has for their own bundle
    agent_own_values = []
    for agent_idx in range(num_agents):
        agent = agents[agent_idx]
        total = sum(valuations[agent][good] for good in allocation[agent_idx])
        agent_own_values.append(total)

    # Check EF1 condition for every pair of agents (i, j)
    for i in range(num_agents):
        for j in range(num_agents):
            if i == j:
                continue
            agent_i = agents[i]
            agent_j_bundle = allocation[j]
```

```python
            # Agent i's value for agent j's bundle
            lst = [valuations[agent_i][good] for good in agent_j_bundle]
            if lst:
                max_value = max(lst)
            else:
                max_value = 0
            value_i_for_j_less_one = sum(lst) - max_value
            # Agent i's own value
            value_i_own = agent_own_values[i]

            if if_some_envy_exists:
              if i == partial_alloc_envy_from:
                if j == partial_alloc_envy_to:
                  value_i_for_j_less_one += 1

            if value_i_own < value_i_for_j_less_one:
                return False
    return True

def find_ef1_allocations(agents, goods, valuations, if_some_envy_exists, partial_alloc_envy_from=0, partial_alloc_envy_to=0):
    """
    Find all allocations that are EF1 at each step of allocating goods one by one.

    Parameters:
    - agents: List of agent identifiers.
    - goods: List of goods to be allocated.
    - valuations: Dictionary where valuations[agent][good] gives the value of a good for an agent.
    - if_some_envy_exists: If the partial allocation for the first 2n+2 rounds is EF (i.e., equal bundle values)
    - partial_alloc_envy_from: If if_some_envy_exists is True, then which agent envies
    - partial_alloc_envy_to: If if_some_envy_exists is True, then which agent is being envied

    Returns:
    - List of allocations. Each allocation is a list of lists, where allocation[i] is the list of goods for agent i.
    """
    num_agents = len(agents)
    all_allocations = []

    def backtrack(current_allocation, index):
        """
        Recursive helper function to perform backtracking.

        Parameters:
        - current_allocation: Current allocation state.
        - index: Index of the next good to allocate.
        """
        if index == len(goods):
            # All goods allocated, add to results
            all_allocations.append(deepcopy(current_allocation))
            return

        current_good = goods[index]

        for agent_idx in range(num_agents):
            # Assign current_good to agent_idx
            current_allocation[agent_idx].append(current_good)

```

```
            # Check EF1 condition at this step
            if is_ef1(current_allocation, agents, valuations, if_some_envy_exists, partial_alloc_envy_from, partial_alloc_envy_to):
                # Continue to allocate the next good
                backtrack(current_allocation, index + 1)

            # Backtrack: remove the good from the agent's allocation
            current_allocation[agent_idx].pop()

    # Initialize allocation: list of empty lists for each agent
    initial_allocation = [[] for _ in agents]
    backtrack(initial_allocation, 0)

    return all_allocations

# Example Usage
if __name__ == "__main__":
    # Define agents and goods
    agents = ['A', 'B','C']
    goods = ['g1', 'g2', 'g3','g4', 'g5','g6', 'g7','g8', 'g9','g10', 'g11','g12', 'g13','g14', 'g15','g16', 'g17','g18', 'g19','g20','g21']

    # Define valuations for each agent
    valuations = {
        'A': {'g1': 90, 'g2': 80, 'g3': 70, 'g4' : 100, 'g5' : 100,'g6': 100, 'g7':15,'g8':10000, 'g9':11000,'g10':12000, 'g11':20000,'g12':20000, 'g13':20000,'g14':20000, 'g15':20000,'g16':20000, 'g17':20000,'g18':20000, 'g19':20000,'g20':19010, 'g21' :18005},
      'B': {'g1': 90, 'g2': 70, 'g3': 80, 'g4' : 100, 'g5' : 100,'g6': 100, 'g7':95,'g8':10000, 'g9':11000,'g10':12000, 'g11':20000,'g12':20000, 'g13':20000,'g14':20000, 'g15':20000,'g16':20000, 'g17':20000,'g18':12000, 'g19':12000,'g20':19085, 'g21' :14106},
      'C': {'g1': 80, 'g2': 90, 'g3': 70, 'g4' : 100, 'g5' : 100,'g6': 100, 'g7':25,'g8':10000, 'g9':11000,'g10':12000, 'g11':20000,'g12':20000, 'g13':18500,'g14':20000, 'g15':20000,'g16':20000, 'g17':20000,'g18':20000, 'g19':20000,'g20':19010, 'g21' :19496}
    }

    # Find all EF1 allocations
    if if_some_envy_exists:
        for partial_alloc_envy_from in range(3):
            for partial_alloc_envy_to in range(3):
                if partial_alloc_envy_from != partial_alloc_envy_to:
                    ef1_allocations = find_ef1_allocations(agents, goods, valuations,True, partial_alloc_envy_from, partial_alloc_envy_to)
                    # Each iteration considers different combinations of envy that exists in the partial allocation for the first 2n+2 rounds

                    # Print the allocations
                    print(f"Total EF1 allocations: {len(ef1_allocations)}\n")
    else:
        ef1_allocations = find_ef1_allocations(agents, goods, valuations, False)
        # Print the allocations
        print(f"Total EF1 allocations: {len(ef1_allocations)}\n")
    for idx, alloc in enumerate(ef1_allocations, 1):
        print(f"Allocation {idx}:")
        for agent_idx, agent in enumerate(agents):
            print(f"  {agent}: {alloc[agent_idx]}")
```

```
        print(f" {sum(valuations['A'][good] for good in alloc[0]) - sum(valuations['A'][good] for good in alloc[1])},{sum(valuations['A'][good] for good in alloc[0]) - sum(valuations['A'][good] for good in alloc[2])}")
        print(f" {sum(valuations['B'][good] for good in alloc[1]) - sum(valuations['B'][good] for good in alloc[0])},{sum(valuations['B'][good] for good in alloc[1]) - sum(valuations['B'][good] for good in alloc[2])}")
        print(f" {sum(valuations['C'][good] for good in alloc[2]) - sum(valuations['C'][good] for good in alloc[0])},{sum(valuations['C'][good] for good in alloc[2]) - sum(valuations['C'][good] for good in alloc[1])}")
        print()
```

B.7 Proof of Theorem 3.14

We reduce from the NP-hard problem Partition. An instance of this problem consists of a multiset S of positive integers; it is a yes-instance if S can be partitioned into two subsets S_1 and S_2 such that the sum of the numbers in S_1 equals the sum of the numbers in S_2, and a no-instance otherwise.

Consider an instance of Partition given by a multiset set $S = \{s_1, \dots, s_m\}$ of m positive integers. Then, we construct a set $S' = \{s'_1, \dots s'_m\}$ such that for each $j \in [m]$, $s'_j = s_m - K$ where $K := \max\{s_1, \dots, s_m\} + \varepsilon$ for some small $\varepsilon > 0$. We then scale members of S' such that they sum to -2, i.e., $\sum_{s' \in S'} s' = -2$.

Next, we construct an instance with four agents and $m + 4$ chores $O = \{b_1, b_2, b_3, b_4, c_1, \dots, c_m\}$, where agents have the following valuation profile $\mathbf{v}$ for $j \in \{1, \dots, m\}$:

$\mathbf{v}$	b_1	b_2	b_3	b_4	c_1	$\dots$	c_j	$\dots$	c_m
1	-1	0	0	0	-1	$\dots$	-1	$\dots$	-1
2	-1	-1	-1	-1	s'_1	$\dots$	s'_j	$\dots$	s'_m
3	-1	-1	-1	-1	s'_1	$\dots$	s'_j	$\dots$	s'_m
4	0	0	0	-1	-1	$\dots$	-1	$\dots$	-1

Also, suppose we are given the partial allocation $\mathcal{A}^4$ where for each $i \in \{1, 2, 3, 4\}$, chore b_i is allocated to agent i, as illustrated in the table above. Note that the partial allocation $\mathcal{A}^4$ is TEF1.

We first establish the following two lemmas. The first lemma states that after chores b_1, b_2, b_3, b_4 are allocated, in order to maintain TEF1, each remaining chore in $\{c_1, \dots, c_m\}$ cannot be allocated to either agent 1 or agent 4. The result is as follows.

Lemma B.1. *In any* TEF1 *allocation, agents* 1 *and* 4 *cannot be allocated any chore in* $\{c_1, \dots, c_m\}$.

The second lemma states that in any TEF1 allocation, the sum of values that agents 2 and 3 obtain from the chores in $\{c_1, \dots, c_m\}$ that are allocated to them must be equal. We formalize it as follows.

Lemma B.2. *In any* TEF1 *allocation, let* C_2, C_3 *be the subsets of* $\{c_1, \dots, c_m\}$ *that were allocated to agents* 2 *and* 3 *respectively. Then,* $v_2(C_2) = v_3(C_3)$.

We will now prove that there exists an allocation $\mathcal{A}$ satisfying TEF1 if and only if the set S can be partitioned into two subsets of equal sum.

For the 'if' direction, suppose $S = \{s_1, \ldots, s_m\}$ can be partitioned into two subsets S_1, S_2 of equal sum. This means that $S' = \{s'_1, \ldots, s'_m\}$ can be correspondingly partitioned into two subsets S'_1, S'_2 of equal sum (of -1 each). Let C_1, C_2 be the partition of chores in $\{c_1, \ldots, c_m\}$ with values corresponding to the partitions S'_1, S'_2 respectively. Then we allocate all chores in C_1 to agent 2 and all chores in C_2 to agent 3. By Lemma B.1, we have that agents 1 and 4 cannot envy any other agent at any round. Also, for any round $t \in [T]$ and $i, j \in \{2, 3\}$ where $i \neq j$, $v_i(A_i^t \setminus \{b_i\}) \geq -1 \geq v_i(A_j^t)$, and for all $i \in \{2, 3\}$ and $k \in \{1, 4\}$, $v_i(A_i^t \setminus \{b_i\}) \geq -1 = v_i(A_k^t)$. Thus, the allocation $\mathcal{A}$ that, for each $i \in \{1, 2, 3, 4\}$, allocates b_i to agent i and for each $j \in \{2, 3\}$, allocates C_j to agent j, is TEF1.

For the 'only if' direction, suppose we have an allocation $\mathcal{A}$ satisfying TEF1. By Lemma B.1, it must be that any chore in $\{c_1, \ldots, c_m\}$ is allocated to either agent 2 or 3. Let C_2, C_3 be the subsets of chores in $\{c_1, \ldots, c_m\}$ that are allocated to agents 2 and 3 respectively, under $\mathcal{A}$. Then, by Lemma B.2, we have that $v_2(C_2) = v_3(C_3)$. By replacing the chores with their corresponding values, we get a partition of S' into two subsets of equal sums, which in turn gives us a partition of S into two subsets of equal sum.

B.8 Proof of Lemma B.1

Consider any TEF1 allocation $\mathcal{A}$. Suppose for a contradiction that at least one of agent 1 and 4 is allocated a chore in $\{c_1, \ldots, c_m\}$. Assume without loss of generality that agent 1 was the first (if not only) agent that received such a chore.

Consider the first round $j + 4$ (for some $j \in [m]$) whereby agent 1 is allocated some chore $c_j \in \{c_1, \ldots, c_m\}$. Then,

$$v_1(A_1^{j+4} \setminus \{b_1\}) = -1 < 0 = v_1(A_4^{j+4}),$$

a contradiction to $\mathcal{A}$ being TEF1.

B.9 Proof of Lemma B.2

Consider any TEF1 allocation $\mathcal{A}$. Suppose for a contradiction that $v_2(C_2) \neq v_3(C_3)$. Since $v_2(C_2) + v_3(C_3) = \sum_{s' \in S'} s' = -2$, it means one of $\{v_2(C_2), v_3(C_3)\}$ is strictly less than -1, and the other is strictly more than -1. Without loss of generality, assume $v_2(C_2) > v_3(C_3)$, i.e., $v_3(C_3) < -1$. We get that

$$v_3(A_3 \setminus \{b_3\}) = v_3(C_3) < -1 = v_3(A_1),$$

contradicting the fact that $\mathcal{A}$ is a TEF1 allocation.

C Omitted Proofs from Section 4

C.1 Proof of Proposition 4.2

We first prove the result for goods. Consider an instance with two agents and four goods $O = \{g_1, g_2, g_3, g_4\}$, with the following valuation profile:

v	g_1	g_2	g_3	g_4
1	1.1	1.1	2	2
2	2	2	1.1	1.1

Observe that the first two goods must be allocated to different agents, otherwise TEF1 will be violated after the second good is allocated. Without loss of generality, suppose that agent 1 receives g_1 and agent 2 receives g_2. We have $v_1(g_1) < v_1(\{g_2, g_3, g_4\}) - v_1(g_3)$ and $v_2(g_2) < v_2(\{g_1, g_3, g_4\}) - v_2(g_1)$, thereby showing that EF1 will be violated if g_3 and g_4 are allocated to the same agent.

Thus, in any TEF1 allocation $\mathcal{A}$, agent 1 must receive one good from $\{g_1, g_2\}$ and one good from $\{g_3, g_4\}$. However, observe that every such allocation $\mathcal{A}$ is Pareto-dominated by the allocation where agent 2 receives bundle $\{g_1, g_2\}$ and agent 1 receives bundle $\{g_3, g_4\}$. This proof can be extended to the case of $n \geq 3$ simply by adding dummy agents who have zero value for each good, and observing that they cannot receive any item in a PO allocation. As such, a TEF1 and PO allocation cannot be guaranteed to exist, even when when there are two types of chores.

Next, we prove the result for chores. Consider an instance with $n \geq 2$ agents and $2n$ chores $O = \{c_1, \dots, c_{2n}\}$, with the following valuation profile:

$\mathbf{v}$	c_1	$\dots$	c_n	c_{n+1}	$\dots$	c_{2n}
1	-1.1	$\dots$	-1.1	-2	$\dots$	-2
2	-2	$\dots$	-2	-1.1	$\dots$	-1.1
3	-2	$\dots$	-2	-2	$\dots$	-2
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
n	-2	$\dots$	-2	-2	$\dots$	-2

In this instance, agent 1 has value -1.1 for each of the first n chores, and value -2 for the last n chores. Agent 2 has value -2 for the first n chores, and value -1.1 for the last n chores. If $n \geq 3$, then agents $3, \dots, n$ have value -2 for all chores.

Observe that each agent must receive one of the first n chores to avoid violating TEF1 within the first n rounds. We now show that each agent must also receive one of the final n chores, otherwise TEF1 will be violated. Suppose for contradiction that in the final allocation $\mathcal{A}$, some agent $i \in N$ is allocated at least two chores from $\{c_{n+1}, \dots, c_{2n}\}$. Then for each $i \in N$, let $c_i' := \arg\min_{c \in A_i} v_i(c)$. We get that

$$v_i(A_i \setminus \{c_i'\}) \leq \begin{cases} -5.1 + 2 = -3.1 & \text{if } i = 1, \\ -4.2 + 2 = -2.2 & \text{if } i = 2, \\ -6 + 2 = -4 & \text{if } i \in \{3, \dots, n\}. \end{cases} \tag{12}$$

By the pigeonhole principle, there exists some other $j \in N \setminus \{i\}$ that receives no chore from $\{c_{n+1}, \dots, c_{2n}\}$, giving us

$$v_i(A_j) = \begin{cases} -1.1 & \text{if } i = 1, \\ -2 & \text{if } i = 2, \\ -2 & \text{if } i \in \{3, \dots, n\}. \end{cases} \tag{13}$$

Consequently, agent i would envy agent j even after removing one chore from her own bundle, and TEF1 is violated. Thus, in any TEF1 allocation, each agent must receive exactly one chore from $\{c_1, \dots, c_n\}$ and exactly one chore out of $\{c_{n+1}, \dots, c_{2n}\}$.

However, any such allocation is Pareto-dominated by another allocation where agent 1 receives exactly two chores from $\{c_1, \dots, c_n\}$ and no chores from $\{c_{n+1}, \dots, c_{2n}\}$, and agent 2 receives no chores from $\{c_1, \dots, c_n\}$ and exactly two chores from $\{c_{n+1}, \dots, c_{2n}\}$. As such, a TEF1 and PO allocation cannot be guaranteed to exist, even when there are two types of chores.

C.2 Proof of Theorem 4.3

We reduce from the NP-hard problem 1-in-3-SAT. An instance of this problem consists of conjunctive normal form F with three literals per clause; it is a yes-instance if there exists a truth assignment to the variables such that each clause has exactly one `True` literal, and a no-instance otherwise.

Consider an instance of 1-in-3-SAT given by the CNF F which contains n variables $\{x_1, \dots, x_n\}$ and m clauses $\{C_1, \dots, C_m\}$.

We construct an instance $\mathcal{I}$ with two agents and $2n+1$ goods. For each $i \in [n]$, we introduce two goods t_i, f_i. We also introduce an additional good r. Let agents' (identical) valuations be defined as follows:

$$v(g) = \begin{cases} 5^{m+n-i} + \sum_{j \,:\, x_i \in C_j} 5^{m-j}, & \text{if } g = t_i, \\ 5^{m+n-i} + \sum_{j \,:\, \neg x_i \in C_j} 5^{m-j}, & \text{if } g = f_i, \\ \sum_{j \in [m]} 5^{j-1}, & \text{if } g = r. \end{cases}$$

Intuitively, for each variable index $i \in [n]$, we associate with it a unique value 5^{m+n-i}. For each clause index $j \in [m]$, we also associate with it a unique value 5^{m-j}. Note that no two indices (regardless of whether its a variable or clause index) share the same value, hence the uniqueness of the values. Then, the value for each good t_i comprises of the unique value associated with i, and the sum over all unique values of clauses C_j which x_i appears as a *positive literal* in; whereas the value for each good f_i comprises of the unique value associated with i, and the sum over all unique values of clauses C_j which x_i appears as a *negative literal* in. We will utilize this in our analysis later.

Then, we have the set of goods $O = \{t_1, f_1, t_2, f_2, \ldots, t_n, f_n, r\}$. Note that

$$v(O) = v(r) + \sum_{i \in [n]} v(t_i) + \sum_{i \in [n]} v(f_i).$$

Also observe that

$$\sum_{i \in [n]} 5^{m+n-i} = \sum_{i \in [n]} 5^{m+i-1}.$$

Now, as each clause contains exactly three literals,

$$\sum_{i \in [n]} \sum_{j: x_i \in C_j} 5^{m-j} + \sum_{i \in [n]} \sum_{j: \neg x_i \in C_j} 5^{m-j} = 3 \times \sum_{j \in [m]} 5^{j-1}.$$

Then, combining the equations above, we get that

$$v(O) = 2 \times \sum_{i \in [n]} 5^{m+i-1} + 4 \times \sum_{j \in [m]} 5^{j-1}. \tag{14}$$

Let the goods appear in the following order:

$$t_1, f_1, t_2, f_2, \ldots, t_n, f_n, r.$$

We first prove the following result.

Lemma C.1. *There exists a truth assignment α such that each clause in F has exactly one* `True` *literal if and only if there exists an allocation $\mathcal{A} = (A_1, A_2)$ such that $v(A_1) = v(A_2)$ for instance $\mathcal{I}$.*

Proof. For the 'if' direction, consider an allocation $\mathcal{A}$ such that $v(A_1) = v(A_2)$. Since agents have identical valuations, without loss of generality, let $r \in A_1$. Since $O = A_1 \cup A_2$ and $v(A_1) = v(A_2) = \frac{1}{2}v(O)$, we have that

$$v(A_1 \setminus \{r\}) = \left(\sum_{i \in [n]} 5^{m+i-1} + 2 \times \sum_{j \in [m]} 5^{j-1} \right) - \sum_{j \in [m]} 5^{j-1} = \sum_{i \in [n]} 5^{m+i-1} + \sum_{j \in [m]} 5^{j-1}.$$

Note that this is only possible if for each $i \in [m]$, t_i and f_i are allocated to different agents. The reason is because the only way agent 1 can obtain the $\sum_{i \in [n]} 5^{m+i-1}$ term of the above bundle value is if he is allocated exactly one good from each of $\{t_i, f_i\}$ for all $i \in [n]$.

Then, from the goods that exist in bundle A_1, we can construct an assignment α: for each $i \in [n]$, let $x_i = \texttt{True}$ if $t_i \in A_1$ and $x_i = \texttt{False}$ if $f_i \in A_1$. Then, from the second term in the expression of $v(A_1 \setminus \{r\})$ above, we can observe that each clause must have exactly one `True` literal.

For the 'only if' direction, consider a truth assignment α such that each clause in F has exactly one `True` literal. Then, for each $i \in [n]$, let

$$\ell_i = \begin{cases} t_i & \text{if } x_i = \texttt{True} \text{ under } \alpha, \\ f_i & \text{if } x_i = \texttt{False} \text{ under } \alpha. \end{cases}$$

We construct the allocation $\mathcal{A} = (A_1, A_2)$ where

$$A_1 = \{\ell_1, \ldots, \ell_n, r\} \quad \text{and} \quad A_2 = O \setminus A_1.$$

Again, observe that

$$\sum_{i \in [n]} 5^{m+n-i} = \sum_{i \in [n]} 5^{m+i-1}.$$

Then, as each clause has exactly one `True` literal,

$$v(A_1) = \sum_{i \in [n]} 5^{m+i-1} + 2 \times \sum_{j \in [m]} 5^{j-1},$$

and together with (14), we get that

$$v(A_2) = v(O) - v(A_1) = v(A_1),$$

as desired. □

Note that for all values of $m, n \geq 1$, and some $\varepsilon < \frac{1}{3}$,

$$5^{m+n} - 2\varepsilon > 5^{m+n-1} + \frac{5^m - 1}{4} = 5^{m+n-1} + \sum_{j \in [m]} 5^{j-1} \geq \max_{g \in O} v(g). \tag{15}$$

Now consider another instance $\mathcal{I}'$ that is similar to $\mathcal{I}$, but with an additional four goods o_1, o_2, o_3, o_4. Let the agents' valuations over these four new goods be defined as follows, for some $\varepsilon < \frac{1}{3}$:

v	o_1	o_2	o_3	o_4
1	5^{m+n}	$5^{m+n} - \varepsilon$	$5^{m+n} - \varepsilon$	5^{m+n}
2	$5^{m+n} - \varepsilon$	5^{m+n}	5^{m+n}	$5^{m+n} - \varepsilon$

Then, we have the set of goods $O' = O \cup \{o_1, o_2, o_3, o_4\}$.

Let the goods be in the following order:

$$t_1, f_1, t_2, f_2, \ldots, t_n, f_n, r, o_1, o_2, o_3, o_4.$$

If there is a partial allocation $\mathcal{A}^{2n+1}$ over the first $2n+1$ goods such that $v(A_1^{2n+1}) = v(A_2^{2n+1})$, then by giving o_1, o_4 to agent 1 and o_2, o_3 to agent 2, we obtain an allocation that is TEF1 and PO (note that any allocation for the first $2n+1$ goods will be PO, since agents have identical valuations over them).

However, if there does not exist a partial allocation $\mathcal{A}^{2n+1}$ over the first $2n+1$ goods such that $v(A_1^{2n+1}) = v(A_2^{2n+1})$, then let $\mathcal{A}^{2n+1}$ be any partial allocation of the first $2n+1$ goods that is TEF1 but $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$. We will show that if $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$, any TEF1 allocation of O' cannot be PO.

Note that in order for $\mathcal{A}^{2n+1}$ to be TEF1, we must have that for any agent $i \in \{1, 2\}$,

$$v(A_i^{2n+1}) \geq \frac{v(O) - \max_{g \in O} v(g)}{2}. \tag{16}$$

This also means that for any agent $i \in \{1, 2\}$,

$$v(A_i^{2n+1}) \leq v(O) - \frac{v(O) - \max_{g \in O} v(g)}{2} = \frac{v(O) + \max_{g \in O} v(g)}{2}. \tag{17}$$

Also observe that since $\min_{g \in O} v(g) > \varepsilon$ and $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$,

$$\left| v(A_1^{2n+1}) - v(A_2^{2n+1}) \right| > \varepsilon. \tag{18}$$

We split our analysis into two cases.

Case 1: $v(A_1^{2n+1}) > v(A_2^{2n+1})$. If we give o_1 to agent 1, since by (15), $v_2(o_1) > \max_{g \in A_1^{2n+1}} v(g)$, we get that

$$v_2(A_2^{2n+2}) = v(A_2^{2n+1}) < v(A_1^{2n+1}) = v_2(A_1^{2n+2} \setminus \{o_1\}),$$

and agent 2 will still envy agent 1 after dropping o_1 from agent 1's bundle. Thus, we must give o_1 to agent 2.

Next, if we give o_2 to agent 2, then since $v_1(o_1) > \max_{g \in O} v(g)$ and $v_1(o_1) > v_1(o_2)$, we have that

$$\begin{aligned}
v_1(A_1^{2n+3}) &= v(A_1^{2n+1}) \\
&\leq \frac{v(O) + \max_{g \in O} v(g)}{2} \quad \text{(by (17))} \\
&< \frac{v(O) - \max_{g \in O} v(g)}{2} + v_1(o_2) \quad \text{(by (15))} \\
&\leq v(A_2^{2n+1}) + v_1(o_2) \quad \text{(by (16))} \\
&= v_1(A_2^{2n+3} \setminus \{o_1\}),
\end{aligned}$$

and agent 1 will still envy agent 2 after dropping o_1 from agent 2's bundle. Thus, we must give o_2 to agent 1. However, such a partial allocation (and thus $\mathcal{A}$) will fail to be PO, as giving o_1 to agent 1 and o_2 to agent 2 instead will strictly increase the utility of both agents.

Case 2: $v(A_1^{2n+1}) < v(A_2^{2n+1})$. If we give o_1 to agent 2, since by (15), $v_1(o_1) > \max_{g \in A_2^{2n+1}} v(g)$, we get that

$$v_1(A_1^{2n+2}) = v(A_1^{2n+1}) < v(A_2^{2n+1}) = v_1(A^{2n+2} \setminus \{o_1\}),$$

and agent 1 will still envy agent 2 after dropping o_1 from agent 2's bundle. Thus, we must give o_1 to agent 1.

Next, if we give o_2 to agent 1, then since $v_2(o_2) > \max_{g \in O} v(g)$ and $v_2(o_2) > v_2(o_1)$, we have that

$$\begin{aligned}
v_2(A_2^{2n+3}) &= v(A_2^{2n+1}) \\
&\leq \frac{v(O) + \max_{g \in O} v(g)}{2} \quad \text{(by (17))} \\
&< \frac{v(O) - \max_{g \in O} v(g)}{2} + v_1(o_1) \quad \text{(by (15))} \\
&\leq v(A_1^{2n+1}) + v_1(o_1) \quad \text{(by (16))} \\
&= v_2(A_1^{2n+3} \setminus \{o_2\}),
\end{aligned}$$

and agent 2 will still envy agent 1 after dropping o_2 from agent 1's bundle. Thus, we must give o_2 to agent 2.

Now, if we give o_3 to agent 2, then since $v_1(o_3) > \max_{g \in O} v(g)$ and $v_1(o_3) = v_1(o_2)$, we have that

$$\begin{aligned} v_1(A_1^{2n+4}) &= v(A_1^{2n+1}) + v_1(o_1) \\ &< v(A_2^{2n+1}) - \varepsilon + v_1(o_1) \quad \text{(by (18))} \\ &= v(A_2^{2n+1}) + v_1(o_2) \\ &= v_1(A_2^{2n+4} \setminus \{o_3\}), \end{aligned}$$

and agent 1 will still envy agent 2 after dropping o_3 from agent 2's bundle. Thus, we must give o_3 to agent 1.

Finally, if we give o_4 to agent 1, then since $v_2(o_3) > \max_{g \in O} v(g)$ and $v_2(o_3) > v_2(o_1) = v_2(o_4)$, we have that

$$\begin{aligned} v_2(A_2) &= v(A_2^{2n+1}) + v_2(o_2) \\ &= v(A_2^{2n+1}) + 5^{m+n} \\ &\leq \frac{v(O) + \max_{g \in O} v(g)}{2} + 5^{m+n} \quad \text{(by (17))} \\ &< \frac{v(O) - \max_{g \in O} v(g)}{2} + 2 \times 5^{m+n} - 2\varepsilon \quad \text{(by (15))} \\ &\leq v(A_1^{2n+1}) + 2 \times 5^{m+n} - 2\varepsilon \quad \text{(by (16))} \\ &= v(A_1^{2n+1}) + v_2(\{o_1, o_4\}) \\ &= v_2(A_1 \setminus \{o_3\}), \end{aligned}$$

and agent 2 will still envy agent 1 after dropping o_3 from agent 1's bundle. Thus, we must give o_4 to agent 2. However, again, this is not PO as giving o_3 to agent 2 and o_4 to agent 1 will strictly increase the utility of both agents.

By exhaustion of cases, we have shown that if $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$, there does not exist a TEF1 and PO allocation over O'. Thus, a TEF1 and PO allocation over O' exists if and only if $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$. By Lemma C.1, this implies that a TEF1 and PO allocation over O' exists if and only if there is a truth assignment α such that each clause in F has exactly one `True` literal.

C.3 Proof of Theorem 4.4

We reduce from the NP-hard problem 1-in-3-SAT. An instance of this problem consists of conjunctive normal form F with three literals per clause; it is a yes-instance if there exists a truth assignment to the variables such that each clause has exactly one `True` literal, and a no-instance otherwise.

Consider an instance of 1-in-3-SAT given by the CNF F which contains n variables $\{x_1, \ldots, x_n\}$ and m clauses $\{C_1, \ldots, C_m\}$.

We construct an instance $\mathcal{I}$ with two agents and $2n+1$ chores. For each $i \in [n]$, we introduce two chores t_i, f_i. We also introduce an additional chore r. Let agents' (identical) valuations be defined as follows:

$$v(c) = \begin{cases} -5^{m+n-i} - \sum_{j \,:\, x_i \in C_j} -5^{m-j}, & \text{if } c = t_i, \\ -5^{m+n-i} - \sum_{j \,:\, \neg x_i \in C_j} 5^{m-j}, & \text{if } c = f_i, \\ - \sum_{j \in [m]} 5^{j-1}, & \text{if } c = r. \end{cases}$$

Intuitively, for each variable index $i \in [n]$, we associate with it a unique value -5^{m+n-i}. For each clause index $j \in [m]$, we also associate it with a unique number -5^{m-j}. Note that no two indices (regardless of whether its a variable or clause index) share the same value, hence the term unique value. Then, the value for each chore t_i comprises of the unique value associated with i, and the sum over all unique values of clauses C_j which x_i appears as a *positive literal* in; whereas the value for each chore f_i comprises of the unique value associated with i, and the sum over all unique values of clauses C_j which x_i appears as a *negative literal* in. We will utilize this in our analysis later.

Then, we have that the set of chores $O = \{t_1, f_1, t_2, f_2, \dots, t_n, f_n, r\}$. Note that

$$v(O) = v(r) + \sum_{i \in [n]} v(t_i) + \sum_{i \in [n]} v(f_i).$$

Also observe that

$$-\sum_{i \in [n]} 5^{m+n-i} = -\sum_{i \in [n]} 5^{m+i-1}.$$

Now, as each clause contains exactly three literals,

$$-\sum_{i \in [n]} \sum_{j: x_i \in C_j} 5^{m-j} - \sum_{i \in [n]} \sum_{j: \neg x_i \in C_j} 5^{m-j} = 3 \times -\sum_{j \in [m]} 5^{j-1}.$$

Then, combining the equations above, we get that

$$v(O) = 2 \times -\sum_{i \in [n]} 5^{m+i-1} + 4 \times -\sum_{j \in [m]} 5^{j-1}. \tag{19}$$

Let the chores appear in the following order:

$$t_1, f_1, t_2, f_2, \dots, t_n, f_n, r.$$

We first prove the following result.

Lemma C.2. *There exists a truth assignment α such that each clause in F has exactly one* `True` *literal if and only if there exists an allocation $\mathcal{A} = (A_1, A_2)$ such that $v(A_1) = v(A_2)$ for instance $\mathcal{I}$.*

Proof. For the 'if' direction, consider an allocation $\mathcal{A}$ such that $v(A_1) = v(A_2)$. Since agents have identical valuations, without loss of generality, let $r \in A_1$. Since $O = A_1 \cup A_2$ and $v(A_1) = v(A_2) = \frac{1}{2} v(O)$, we have that

$$v(A_1 \setminus \{r\}) = \left(-\sum_{i \in [n]} 5^{m+i-1} + 2 \times -\sum_{j \in [m]} 5^{j-1} \right) + \sum_{j \in [m]} 5^{j-1} = -\sum_{i \in [n]} 5^{m+i-1} - \sum_{j \in [m]} 5^{j-1}.$$

Note that this is only possible if for each $I \in [m]$, t_i and f_i are allocated to different agents. The reason is because the only way agent 1 can obtain the first term of the above bundle value (less chore r) is if she is allocated exactly one chore from each of $\{t_i, f_i\}$ for each $i \in [n]$.

Then, from the chores that exists in bundle A_1, we can construct an assignment α: for each $i \in [n]$, let $x_i =$ `True` if $t_i \in A_1$ and $x_i =$ `False` if $f_i \in A_1$. Then, from the second term in the expression of $v(A_1 \setminus \{r\})$ above, we can observe that each clause has exactly one `True` literal (because the sum is only obtainable if exactly one literal appears in each clause, and our assignment will cause each these literals to evaluate `True`.

For the 'only if' direction, consider a truth assignment α such that each clause in F has exactly one `True` literal. Then, for each $i \in [n]$, let

$$\ell_i = \begin{cases} t_i & \text{if } x_i = \texttt{True} \text{ under } \alpha, \\ f_i & \text{if } x_i = \texttt{False} \text{ under } \alpha. \end{cases}$$

We construct the allocation $\mathcal{A} = (A_1, A_2)$ where

$$A_1 = \{\ell_1, \dots, \ell_n, r\} \quad \text{and} \quad A_2 = O \setminus A_1.$$

Again, observe that

$$-\sum_{i\in[n]} 5^{m+n-i} = -\sum_{i\in[n]} 5^{m+i-1}.$$

Then, as each clause has exactly one `True` literal,

$$v(A_1) = -\sum_{i\in[n]} 5^{m+i-1} + 2 \times -\sum_{j\in[m]} 5^{j-1},$$

and together with (14), we get that

$$v(A_2) = v(O) - v(A_1) = v(A_1),$$

as desired. $\square$

Note that for all values of $m, n \geq 1$, and some $\varepsilon < \frac{1}{3}$,

$$\frac{-5^{m+n} + 2\varepsilon}{2} < -5^{m+n-1} - \frac{5^m - 1}{4} = -5^{m+n-1} - \sum_{j\in[m]} 5^{j-1} \leq \min_{c\in O} v(c). \tag{20}$$

Now, consider another instance $\mathcal{I}'$ that is similar to $\mathcal{I}$, but with an additional four chores o_1, o_2, o_3, o_4. Let agents' valuations over these four new chores be defined as follows, for some $\varepsilon < \frac{1}{3}$:

v	o_1	o_2	o_3	o_4
1	-5^{m+n}	$-5^{m+n} + \varepsilon$	$-5^{m+n} + \varepsilon$	-5^{m+n}
2	$-5^{m+n} + \varepsilon$	-5^{m+n}	-5^{m+n}	$-5^{m+n} + \varepsilon$

Then, we have the set of chores $O' = O \cup \{o_1, o_2, o_3, o_4\}$.

Let the chores be in the following order:

$$t_1, f_1, t_2, f_2, \dots, t_n, f_n, r, o_1, o_2, o_3, o_4.$$

If there is a partial allocation $\mathcal{A}^{2n+1}$ over the first $2n+1$ chores such that $v(A_1^{2n+1}) = v(A_2^{2n+1})$, then by giving o_1, o_4 to agent 1 and o_2, o_3 to agent 2, we obtain an allocation that is TEF1 and PO (note that any allocation for the first $2n+1$ chores will be PO, since agents have identical valuations over them).

However, if there does not exist a partial allocation $\mathcal{A}^{2n+1}$ over the first $2n+1$ goods such that $v(A_1^{2n+1}) = v(A_2^{2n+1})$, then let $\mathcal{A}^{2n+1}$ be any partial allocation of the first $2n+1$ goods that is TEF1 but $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$.

Note that in order for $\mathcal{A}^{2n+1}$ to be TEF1, we must have that for any agent $i \in \{1, 2\}$,

$$v(A_i^{2n+1}) - \min_{c\in O} v(c) \geq \frac{v(O)}{2}. \tag{21}$$

This also means that for any agent $i \in \{1, 2\}$,

$$v(A_i^{2n+1}) \leq v(O) - \left(\frac{v(O)}{2} + \min_{c\in O} v(c)\right) = \frac{v(O)}{2} - \min_{c\in O} v(c). \tag{22}$$

Also observe that since $\min_{g\in O} v(g) > \varepsilon$ and $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$,

$$\left|v(A_1^{2n+1}) - v(A_2^{2n+1})\right| > \varepsilon. \tag{23}$$

We split our analysis into two cases.

Case 1: $v(A_1^{2n+1}) > v(A_2^{2n+1})$**.** If we give o_1 to agent 2, since by (20), $v_2(o_1) < \min_{c \in A_2^{2n+1}} v(c)$, we get that

$$v_2(A_2^{2n+2} \setminus \{o_1\}) = v(A_2^{2n+1}) < v(A_1^{2n+1}) = v_2(A_1^{2n+2}),$$

and agent 2 will still envy agent 1 after dropping o_1 from his own bundle. Thus, we must give o_1 to agent 1.

Next, if we give o_2 to agent 1, then since $v_1(o_1) < \max_{c \in O} v(c)$ and $v_1(o_1) < v_1(o_2)$, we have that

$$\begin{aligned}
v_1(A_1^{2n+3} \setminus \{o_1\}) &= v(A_1^{2n+1}) + v_1(o_2) \\
&\leq \frac{v(O)}{2} - \min_{c \in O} v(c) + v_1(o_2) \quad \text{(by (22))} \\
&< \frac{v(O)}{2} + \min_{c \in O} v(c) \quad \text{(by (20))} \\
&\leq v(A_2^{2n+1}) \quad \text{(by (21))} \\
&= v_1(A_2^{2n+3}),
\end{aligned}$$

and agent 1 will still envy agent 2 after dropping o_2 from her own bundle. Thus, we must give o_2 to agent 2. However, such a partial allocation (and thus $\mathcal{A}$) will fail to be PO, as giving o_1 to agent 2 and o_2 to agent 1 will strictly increase the utility of both agents.

Case 2: $v(A_1^{2n+1}) < v(A_2^{2n+1})$**.** If we give o_1 to agent 1, since by (20), $v_1(o_1) \min_{c \in A_1^{2n+1}} v(c)$, we get that

$$v_1(A_1^{2n+2} \setminus \{o_1\}) = v(A_1^{2n+1}) < v(A_2^{2n+1}) = v_1(A^{2n+2}),$$

and agent 1 will still envy agent 2 after dropping o_1 from her own bundle. Thus, we must give o_1 to agent 2.

Next, if we give o_2 to agent 2, then since $v_2(o_2) < \min_{c \in O} v(c)$ and $v_2(o_2) < v_2(o_1)$, we have that

$$\begin{aligned}
v_2(A_2^{2n+3} \setminus \{o_2\}) &= v(A_2^{2n+1}) + v_2(o_1) \\
&\leq \frac{v(O)}{2} - \min_{c \in O} v(c) + v_2(o_1) \quad \text{(by (22))} \\
&< \frac{v(O)}{2} + \min_{c \in O} v(c) \quad \text{(by (20))} \\
&\leq v(A_1^{2n+1}) \quad \text{(by (21))} \\
&= v_2(A_1^{2n+3}),
\end{aligned}$$

and agent 2 will still envy agent 1 after dropping o_2 from his own bundle. Thus, we must give o_2 to agent 1.

Now, if we give o_3 to agent 1, then since $v_1(o_3) < \min_{c \in O} v(c)$ and $v_1(o_3) = v_1(o_2)$, we have that

$$\begin{aligned}
v_1(A_1^{2n+4} \setminus \{o_3\}) &= v(A_1^{2n+1}) + v_1(o_2) \\
&< v(A_2^{2n+1}) - \varepsilon + v_1(o_2) \quad \text{(by (23))} \\
&= v(A_2^{2n+1}) + v_1(o_1) \\
&= v_1(A_2^{2n+4}),
\end{aligned}$$

and agent 1 will still envy agent 2 after dropping o_3 from her own bundle. Thus, we must give o_3 to agent 2.

Finally, if we give o_4 to agent 2, then since $v_2(o_3) < \min_{c \in O} v(c)$ and $v_2(o_3) < v_2(o_1) = v_2(o_4)$, we have that

$$\begin{aligned}
v_2(A_2 \setminus \{o_3\}) &= v(A_2^{2n+1}) + v_2(\{o_1, o_4\}) \\
&= v(A_2^{2n+1}) - 2 \times 5^{m+n} + 2\varepsilon \\
&\leq \frac{v(O)}{2} - \min_{c \in O} v(c) - 2 \times 5^{m+n} + 2\varepsilon \quad \text{(by (22))} \\
&< \frac{v(O)}{2} + \min_{c \in O} v(c) - 5^{m+n} + \varepsilon \quad \text{(by (20))} \\
&\leq v(A_1^{2n+1}) + v_2(o_1) \quad \text{(by (21))} \\
&= v(A_1^{2n+1}) + v_2(o_1) \\
&= v_2(A_1),
\end{aligned}$$

and agent 2 will still envy agent 1 after dropping o_3 from his own bundle. Thus, we must give o_4 to agent 1. However, again, this is not PO as giving o_3 to agent 1 and o_4 to agent 2 will strictly increase the utility of both agents.

By exhaustion of cases, we have shown that if $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$, there does not exist a TEF1 and PO allocation over O'. Thus, a TEF1 and PO allocation over O' exists if and only if $v(A_1^{2n+1}) \neq v(A_2^{2n+1})$. By Lemma C.2, this implies that a TEF1 and PO allocation over O' exists if and only if there is a truth assignment α such that each clause in F has exactly one `True` literal.

D Omitted Proofs from Section 5

D.1 Proof of Theorem 5.1

Let $\mathcal{A}^1$ and $\mathcal{B} = \mathcal{A}^2 \setminus \mathcal{A}^1$ be the allocations of item sets O_1 and O_2 respectively. Note that while we are in the setting whereby $O_1 = O_2$, we can simply relabel item.

We first address a special case. When allocating chores, for each $t \in \{1, 2\}$ such that $|O_t| < n$ (i.e., there are less chores than agents in either round), add $n - |O_t|$ zero-valued dummy chores to O_t.

To obtain $\mathcal{A}^1$, we allocate the items in the first round in a round-robin fashion, with picking sequence $(1, \dots, n)^*$. That is, agent 1 picks their most preferred item, followed by agent 2, and so on until agent n, after which the sequence restarts. The items arriving in the second round are also allocated in a round-robin fashion to obtain $\mathcal{B}$, but with picking sequence $(n, \dots, 1)^*$. The round-robin algorithm is well-known to satisfy EF1 for both the goods and chores settings [9], so we know that $\mathcal{A}^1$ and $\mathcal{B}$ are EF1. It remains to show that $\mathcal{A}^2 = \mathcal{A}^1 \cup \mathcal{B}$ is EF1.

Consider an arbitrary pair of agents i, j. If $i < j$, then

$$v_i(A_i^1) \geq v_i(A_j^1),$$

because i precedes j in the picking sequence for allocation $\mathcal{A}$. Similarly, if $i > j$, then

$$v_i(B_i) \geq v_i(B_j).$$

Note that these inequalities hold for both goods and chores.

We now prove our result for goods. Consider an arbitrary agent i. Since $\mathcal{A}^1$ and $\mathcal{B}$ are EF1, we know that for any agent $j \neq i$, there exists a good $g_a \in A_j^1$ such that $v_i(A_i^1) \geq v_i(A_j^1 \setminus \{g_a\})$, and there exists a good $g_b \in B_j$ such that $v_i(B_i) \geq v_i(B_j \setminus \{g_b\})$. Therefore for any agent $j < i$, there exists $g_a \in A_j^1$ such that

$$v_i(A_i^2) = v_i(A_i^1 \cup B_i) \geq v_i(A_j^1) - v_i(g_a) + v_i(B_j) = v_i(A_j^2) - v_i(g_a) = v_i(A_j^2 \setminus \{g_a\}).$$

Similarly, for any $j > i$, there exists $g_b \in B_j$ such that $v_i(A_i^2) \geq v_i(A_j^2 \setminus \{g_b\})$.

We next prove our result for chores. Again consider an arbitrary agent i. Due to $\mathcal{A}^1$ and $\mathcal{B}$ satisfying EF1, for any agent $j \neq i$, there exists a chore $c_a \in A_i^1$ such that $v_i(A_i^1 \setminus \{c_a\}) \geq v_i(A_j^1)$, and there exists a chore $c_b \in B_j$ such that $v_i(B_i \setminus \{c_b\}) \geq v_i(B_j)$. Therefore for any $j < i$, there exists $c_a \in A_i^1$ such that

$$v_i(A_i^2 \setminus \{c_a\}) = v_i(A_i^1 \setminus \{c_a\}) + v_i(B_i) \geq v_i(A_j^1) + v_i(B_j) = v_i(A_j^2).$$

Similarly, for any $j > i$, there exists $c_b \in B_i$ such that $v_i(A_i^2 \setminus \{c_b\}) \geq v_i(A_j^2)$. This concludes the proof.

D.2 Proof of Theorem 5.2 (continued)

We now prove the result for the case of chores. We construct a set $S' = \{s'_1, \ldots, s'_m\}$ such that for each $j \in [m]$, $s'_j = -K + s_j$ where $K := \max\{s_1, \ldots, s_m\}$. Observe that S' contains non-positive integers. Let $W' := \frac{1}{\kappa} \sum_{j \in [m]} s'_j$

Then, we construct an instance with $\kappa + 1$ agents and $m+1$ chores in each round: $O_1 = \{c_1, \ldots, c_{m+1}\}$ and $O_2 = \{c'_1, \ldots, c'_{m+1}\}$, where agents have an identical valuation function v defined as follows:

$$v(c_j) = v(c'_j) = \begin{cases} s'_j, & \text{if } j \leq m, \\ 2W', & \text{if } j = m+1. \end{cases}$$

We will now prove that there exists a repetitive TEF1 allocation $\mathcal{A}$ if and only if the set S can be partitioned into κ subsets with equal sums (of W each).

For the 'if' direction, consider a κ-way partition $\mathcal{P} = \{P_1, \ldots, P_\kappa\}$ of S with equal sums (of W each). This means that S' can also be partitioned into κ subsets of equal sums (with the same partition $\mathcal{P}$; let the sum be W'). We construct allocations $\mathcal{A}^1$ and $\mathcal{A}^2$ such that the chores in both rounds are allocated identically, and show that $\mathcal{A}^2$ satisfies TEF1.

For each $i \in \{1, \ldots, \kappa\}$, allocate the chores corresponding to the elements of subset P_i to agent i, and the chore c_{m+1} to agent $\kappa + 1$. Then, in $\mathcal{A}^1$, for each agent $i \in [\kappa]$, $v(A_i^1) = \sum_{c \in P_i} c = W'$, and $v(A_{\kappa+1}^1) = v(\{c_{m+1}\}) = 2W'$. It is easy to verify that $\mathcal{A}^1$ is TEF1: every pair of agents $i, j \in [\kappa]$ has the same bundle value, and each agent $i \in [\kappa]$ has a higher bundle value than agent $\kappa + 1$. Also, agent $\kappa + 1$ will not envy any agent $i \in [\kappa]$ after removing chore $c_{m+1} \in A_{\kappa+1}^1$.

Next, we consider $\mathcal{A}^2$. For each agent $i \in [\kappa]$, $v(A_i^2) = 2W'$, and $v(A_{\kappa+1}^2) = 4W'$. We verify that $\mathcal{A}^2$ is TEF1: again, each pair of agents $i, j \in [\kappa]$ has the same bundle value, and each agent $i \in [\kappa]$ has a higher bundle value than agent $\kappa + 1$. Also, agent $\kappa + 1$ will not envy any agent $i \in [\kappa]$ after removing chore $c_{m+1} \in A_{\kappa+1}^2$.

For the 'only if' direction, suppose we have a repetitive allocation $\mathcal{A}^2$ which satisfies TEF1. Since agents have identical valuation functions, without loss of generality, suppose that agent $\kappa + 1$ receives chore c_{m+1} under $\mathcal{A}^1$. Then, $v(A_{\kappa+1}^2 \setminus \{c_{m+1}\}) \leq 2W'$. In order for $\mathcal{A}^2$ to be TEF1, we must have that $v(A_i^2) \leq 2W'$ for each $i \in [\kappa]$ (so that agent $\kappa + 1$ will not envy any agent $i \in [\kappa]$). This means that for each $i \in [\kappa]$, $v(A_i^1) \leq W'$, but since $\sum_{j \in [m]} s'_j = \kappa W'$, this is only possible if there is a κ-way partition of S' such that each subset has a sum of W' (i.e. there is a κ-way partition of S such that each subset has a sum of W).

E TEF1 for Mixed Manna

We first define TEF1 for mixed manna.

Definition E.1 (Temporal EF1 for mixed manna)**.** In the case of with both goods and chores, an allocation $\mathcal{A}^t = (A_1^t, \dots, A_n^t)$ is said to be *temporal envy-free up to one item (TEF1)* if for all $t' \leq t$ and $i, j \in N$, there exists an item $o \in A_i^{t'} \cup A_j^{t'}$ such that $v_i(A_i^{t'} \setminus \{o\}) \geq v_i(A_j^{t'} \setminus \{o\})$.

Then, we can extend the result of Theorem 3.2 to the more general mixed manna setting, with the following result.

Theorem E.2. *When* $n = 2$*, a* TEF1 *allocation exists in the mixed manna setting, and can be computed in polynomial time.*

Proof. For an agent $i \in \{1, 2\}$ and round $t \in [T]$, we define $S_i^t \subseteq O^t$ as the set of items that have arrived up to round t which only agent i has a positive value for. Then, for any $t \in [t]$ and $i, j \in \{1, 2\}$ where $i \neq j$, $v_i(S_i^t) \geq 0$ and $v_i(S_j^t) \leq 0$. Clearly, if some allocation $\mathcal{A}^t$ is TEF1 over $O^t \setminus (S_1^t \cup S_2^t)$, then $\mathcal{B}^t = (S_1^t \cup A_1^t, S_2^t \cup A_2^t)$ is a TEF1 allocation over O^t. Furthermore, for any $t \in [T]$ and $i, j \in \{1, 2\}$ where $i \neq j$, if there exists an item $o \in A_i^t \cup A_j^t$ such that $v_i(A_i^t \setminus \{o\}) \geq v_i(A_j^t \setminus \{o\})$, then

$$\begin{aligned} v_i(B_i^t \setminus \{o\}) &= v_i(A_i^t \setminus \{o\}) + v_i(S_i^t) \\ &\geq v_i(A_i^t \setminus \{o\}) \\ &\geq v_i(A_j^t \setminus \{o\}) \\ &\geq v_i(A_j^t \setminus \{o\}) + v_i(S_j^t) \\ &= v_i(B_j^t \setminus \{o\}), \end{aligned}$$

where the first and third inequalities are due to the fact that $v_i(S_i^t) \geq 0$ and $v_i(S_j^t) \leq 0$. It therefore suffices to assume that for each item $o \in O$, either $v_1(o) \leq 0$ and $v_2(o) \leq 0$, or $v_1(o) \geq 0$ and $v_2(o) \geq 0$, and we make this assumption for the remainder of the proof.

Let $v_i'(o) = |v_i(o)|$ for all $i \in \{1, 2\}$ and $o \in O$. Note that $v_i'(o) \geq 0$ for all $i \in \{1, 2\}$ and $o \in O$ and thus, with respect to the augmented valuations, each $o \in O$ is a good. We use Algorithm 2 in He et al. [46], which returns a TEF1 allocation for goods in polynomial time, to compute an allocation $\mathcal{B}$ which is TEF1 with respect to the augmented valuations $\mathbf{v}'$.

For a round $t \in [T]$, let $G^t, C^t \subseteq O^t$ be, respectively, the subsets of goods and chores (with respect to the original valuation profile $\mathbf{v} = (v_1, v_2)$) that have arrived up to round t. Then, for each $t \in [T]$ and $i \in \{1, 2\}$, let $G_i^t = G^t \cap B_i^t$ and $C_i^t = C^t \cap B_i^t$. We construct allocation $\mathcal{A} = (G_1^T \cup C_2^T, G_2^T \cup C_1^T)$ from $\mathcal{B}$ by swapping the agents' bundles of chores. We now show that $\mathcal{A}$ is TEF1.

Recall that all items are goods with respect to $\mathbf{v}'$. Since $\mathcal{B}$ is TEF1, we know that for any $t \in [T]$ and $i, j \in \{1, 2\}$ where $i \neq j$, there exists an item $o \in B_j^t$ such that

$$\begin{aligned} & v_i'(B_i^t) \geq v_i'(B_j^t \setminus \{o\}) \\ \implies\; & v_i(G_i^t) - |v_i(C_i^t)| \geq v_i(G_j^t) - |v_i(C_j^t)| - |v_i(o)| \\ \implies\; & v_i(G_i^t) + v_i(C_j^t) \geq v_i(G_j^t) + v_i(C_i^t) - |v_i(o)| \\ \implies\; & \begin{cases} v_i(G_i^t) + v_i(C_j^t) \geq v_i(G_j^t \setminus \{o\}) + v_i(C_i^t) & \text{if } o \in G_j^t \\ v_i(G_i^t) + v_i(C_j^t \setminus \{o\}) \geq v_i(G_j^t) + v_i(C_i^t) & \text{if } o \in C_j^t \end{cases} \\ \implies\; & v_i(G_i^t \cup C_j^t \setminus \{o\}) \geq v_i(G_i^t \cup C_j^t \setminus \{o\}) \\ \implies\; & v_i(A_i^t \setminus \{o\}) \geq v_i(A_j^t \setminus \{o\}). \end{aligned}$$

Thus, $\mathcal{A}$ is TEF1. $\square$

Edith Elkind
Northwestern University
Evanston, Illinois, USA
Email: edith.elkind@northwestern.edu

Alexander Lam
Hong Kong Polytechnic University
Hong Kong, Hong Kong SAR
Email: alexander-a.lam@polyu.edu.hk

Mohamad Latifian
University of Edinburgh
Edinburgh, UK
Email: mohamad.latifian@ed.ac.uk

Tzeh Yuan Neoh
Harvard University
Cambridge, Massachusetts, USA
Email: tzehyuan_neoh@g.harvard.edu

Nicholas Teh
University of Oxford
Oxford, UK
Email: nicholas.teh@cs.ox.ac.uk