Nearly Equitable Allocations beyond Additivity and Monotonicity

Siddharth Barman[1], **Umang Bhaskar**[2], **Yeshwant Pandit**[2], **Soumyajit Pyne**[2]

[1] Indian Institute of Science, Bangalore
[2] Tata Institute of Fundamental Research, Mumbai
barman@iisc.ac.in, umang@tifr.res.in, soumyajit.pyne@tifr.res.in, yeshwant.pandit@tifr.res.in

Abstract

Equitability (EQ) in fair division requires that items be allocated such that all agents value the bundle they receive equally. With indivisible items, an equitable allocation may not exist, and hence we instead consider a meaningful analog, EQx, that requires equitability up to any item. EQx allocations exist for monotone, additive valuations. However, if (1) the agents' valuations are not additive or (2) the set of indivisible items includes both goods and chores (positively and negatively valued items), then prior to the current work it was not known whether EQx allocations exist or not.

We study both the existence and efficient computation of EQx allocations. (1) For monotone valuations (not necessarily additive), we show that EQx allocations always exist. Also, for the large class of weakly well-layered valuations, EQx allocations can be found in polynomial time. Further, we prove that approximately EQx allocations can be computed efficiently under general monotone valuations. (2) For non-monotone valuations, we show that an EQx allocation may not exist, even for two agents with additive valuations. Under some special cases, however, we show existence and efficient computability of EQx allocations. This includes the case of two agents with additive valuations where each item is either a good or a chore, and there are no mixed items.

1 Introduction

In the problem of fair division, a central planner (principal) is tasked with *fairly* partitioning a set of items among interested agents. If the items are indivisible, which is our focus, each item must be allocated integrally to an agent. Every agent i has a valuation function v_i that specifies agent i's value for each subset of items. Here, an item x could be a 'good', if every agent always values it positively, a 'chore', if every agent always values it negatively, or 'mixed', if across agents the value for item x can be both positive and negative.

What constitutes a fair allocation of items has no single answer. Over the years, various notions have been studied in depth (Moulin 2004). Possibly the most prominent among them are *envy-freeness* and *equitability*. An allocation is said to be envy-free if each agent prefers her own bundle of items to the bundle allocated to anyone else (Foley 1966). An allocation is said to be equitable if agents' values for their own bundles are the same and, hence, the agents are equally content (Dubins and Spanier 1961). If the agents have identical valuations, then the two notions coincide.

Envy-freeness has received significant attention in classic fair division literature. It is known, for example, that for additive valuations over *divisible* goods, an allocation that maximizes the Nash social welfare (the product of the agents' values for their allocated bundles) is also envy-free (Varian 1974).

The widely used platform Spliddit.org implements multiple methods to provide solutions for common fair division problems (Goldman and Procaccia 2015). The platform uses envy-freeness, in particular, as a fairness criterion for relevant applications.

Notably, equitability is a simpler construct to reason about, since it requires fewer comparisons. To test an allocation for equitability, we only need each agent's value for her own bundle, rather than every agent's value for every bundle.

Perhaps for this reason, equitable solutions are important in practical applications of fair division. Experimental studies have noted that, in specific fair-division settings, users tend to prefer equitable allocations over other notions of fairness (Herreiner and Puppe 2009).[1] Further, in bargaining experiments, equitability often plays a significant role in determining the outcome (Herreiner and Puppe 2010). Also in Spliddit.org, for fairly dividing rent among housemates, equitability was noted to be important as a refining criterion, following envy-freeness. The latest rent-division algorithm used in Spliddit.org computes solutions that satisfy this supporting objective (Gal et al. 2017).[2]

The real-world significance of equitability is further highlighted by the case of divorce settlements. The two legal means of dividing property in the United States are *community property* and *equitable distribution* (Kagan 2021). In the community property rule, holdings are divided equally among the divorcing couple and, hence, the rule induces an envy-free division. Equitable distribution, on the other hand, takes into account various factors (such as the employability and financial needs of each party) for dividing the as-

[1] These prior works refer to equitability as inequality aversion.

[2] Specifically, rent divisions that maximize the minimum value, called maximin solutions, are nearly equitable. The latest algorithm implemented in Spliddit.org finds rent divisions that satisfy envy-freeness and are also maximin.

	Existence	Polynomial-Time Algorithm
Monotone, Beyond Additive (Section 3)	✓ Theorem 2	✓ Well-Layered: Theorem 3 ✓ Approximate EQx: Theorem 5
Nonmonotone Additive (Section 4)	✗ Subjective Valuations: Theorem 6 ✓ Identical Chores: Theorem 9 ✓ Single Chore: Theorem 10	✓ Two agents, objective valuations: Theorem 7

Table 1: Our Results

sets, and, in particular, makes inter-personal comparisons of value. Most states in the US follow equitable distribution, i.e., the courts divide assets and liabilities based on equitability. The definition of equitability in this situation is somewhat intuitive. However, the evidence clearly suggests that equitability is an important concept in practical situations.

Motivated by such considerations, the current work studies equitability (EQ), with the focus on allocating indivisible items. In the discrete fair division context, simple examples (with two agents and a single indivisible item) demonstrate that exactly equitable allocations may not exist. Hence, recent works have focused on relaxations. A compellingly strong relaxation is obtained by requiring that any existing inequality in agents' values is switched by the (hypothetical) removal of any good or chore in an appropriate manner. This relaxation is called equitability up to any item, EQx. Specifically, an allocation is said to be EQx if, whenever agent i has a lower value for her bundle, say A_i, than some other agent j for her bundle, say A_j (i.e., $v_i(A_i) < v_j(A_j)$), the removal of any positively-valued item (good) from A_j or the removal of any negatively-valued item (chore) from A_i ensures that the inequality is (weakly) reversed. Similarly, an allocation is EFx if, whenever agent i has higher value for agent j's bundle than her own (i.e., $v_i(A_i) < v_i(A_j)$), this preference can be weakly reversed by removing a good from A_j or a chore from A_i.

Given the relevance of equitability, a fundamental question in discrete fair division is whether EQx allocations always exist. For the specific case of additively valued goods, EQx allocations are known to exist; this result is obtained via a greedy algorithm (Gourvès, Monnot, and Tlilane 2014).[3] Beyond this setting, however, this question has not been addressed in the literature. Our work addresses this notable gap. In particular, moving beyond monotone additive valuations, the current work establishes novel existential and algorithmic guarantees for EQx allocations. For general monotone valuations, for example, our work shows the universal existence of EQx allocations.

Our Results and Techniques. Under monotone, nondecreasing valuations, we establish (in Section 3):

- EQx allocations always exist, and can be computed in pseudo-polynomial time (Theorem 2). Our algorithm is based on a modification of an Add-and-Fix algorithm, proposed earlier for EFx under *identical* monotone nonincreasing valuations (Barman, Narayan, and Verma 2023).
- Under *weakly well-layered* (WWL) valuations, EQx allocations can be computed in polynomial time (Theorem 3). WWL functions are an encompassing and natural class of valuations that include gross substitutes, weighted matroid-rank functions, budget-additive, well-layered, and cancelable valuations (Goldberg, Høgh, and Hollender 2023).
- For any $\varepsilon \in (0, 1)$, a $(1-\varepsilon)$-approximately EQx allocation can be computed in time polynomial in $1/\varepsilon$ and the input size (Theorem 5).

Finding an EFx allocation is known to be hard (Plaut and Roughgarden 2020a). Since the negative result holds even for two agents, with *identical* submodular valuations, the hardness also applies to EQx. This observation signifies that our polynomial-time algorithm for WWL valuations is the best possible, in the sense that such a positive result is unlikely for submodular valuations in general.

We then address nonmonotone additive valuations (Section 4). In comparison to monotone valuations, the results here are mixed and highlight a complicated landscape.

- For agents with subjective valuations—when mixed items are allowed—EQx allocations may not exist, even for just two agents with normalized valuations (Theorem 6).[4] This negative result necessitates focusing on objective valuations, wherein each item is exclusively a good or a chore.

For agents with *identical* additive valuations, Aziz and Rey (2020) develop an efficient algorithm for finding EFx (and, hence, EQx) allocations. By the well-known cut-and-choose protocol, this gives an EFx algorithm for two nonidentical agents. However, cut-and-choose does not work for EQx and, hence, there was no known EQx algorithm for two nonidentical agents. Our next result addresses this gap.

- For two agents with additive objective valuations, we develop a polynomial-time algorithm for finding EQx allocations (Theorem 7).

Many instances of fair division, in fact, consist of just two agents (such as in divorce settlement, or in the experiments of Gal et al. 2017), hence our result for two agents is of practical significance.

- For n agents with additive objective valuations where each chore c has the same value for the agents (i.e., $v_i(c) = v_j(c)$ for all agents i and j), we show—using the leximin++ ordering—that EQx allocations always exist (Theorem 9).
- For n agents with additive objective valuations and a single chore, we show that EQx allocations exist (Theorem 10).

The last result is technically challenging. It is obtained via a local search algorithm, that resolves EQx violations,

[3] EQx is termed as *near jealousy-freeness* in (Gourvès, Monnot, and Tlilane 2014).

[4] Valuations are *normalised* if, for the set M of all the goods, $v_i(M)$ is equal for all agents i.

whenever they arise, in a specific order. The analysis is subtle and needs to keep track of multiple progress measures, including the leximin++ value. Table 1 summarizes our results. Due to lack of space, all missing proofs are given in the full version of the paper (Barman et al. 2023).

The existence of EQx allocation under objective, additive valuations stands as an interesting, open question.

Additional Related Work. Equitability was first studied in the divisible setting of cake cutting. Here, equitable allocations for n agents exist (Dubins and Spanier 1961; Cechlárová, Doboš, and Pillárová 2013; Chèze 2017) and, in particular, do not require additivity (Aumann and Dombb 2015). However, no finite algorithm can find an exactly equitable allocation (Procaccia and Wang 2017), though approximately equitable allocations can be computed in near-linear time (Cechlárová and Pillárová 2012).

For indivisible items, EQx allocations were first shown to exist for monotone additive valuations (Gourvès, Monnot, and Tlilane 2014; Freeman et al. 2020). This existential guarantee was obtained via an efficient greedy algorithm. For additive valuations that are strictly positive, allocations that are both EQx and Pareto optimal are known to exist (Freeman et al. 2019).

When the agents have identical valuations, EFx and EQx allocations coincide. Hence, under identical valuations, existential guarantees obtained for EFx allocations extend to EQx as well. In particular, for identical monotone (nondecreasing) valuations, EFx allocations were shown to exist using the leximin++ construct (Plaut and Roughgarden 2020a). This work also showed that even for two agents with identical submodular valuations, finding an EFx allocation requires an exponential number of queries. This problem is also PLS-complete (Goldberg, Høgh, and Hollender 2023). For objective identical valuations, the existence of EFx allocations was shown through a modification of the leximin construct (Chen and Liu 2020). For additive identical valuations, EFx existence, as well as efficient computation, was obtained by Aziz and Rey 2020.

Finally, a number of recent papers have also studied the loss of efficiency for (near) equitable and near equitable allocations (Caragiannis et al. 2012; Aumann and Dombb 2015; Freeman et al. 2019, 2020; Sun, Chen, and Doan 2023; Bhaskar et al. 2023).

2 Notation and Preliminaries

A fair division instance $(N, M, \mathcal{V})$ consists of a set $N = \{1, 2, \ldots, n\}$ of agents, a set M of indivisible items, with $m = |M|$, and a valuation function $v_i \in \mathcal{V}$ for each agent $i \in N$. The valuation $v_i : 2^M \to \mathbb{Z}$ specifies agent i's value for every subset of the items. We will assume, throughout, that $v_i(\emptyset) = 0$, and all the agents' values are integral.[5]

For notational convenience, for single items $x \in M$, we use $v_i(x)$ and $v_i(S \cup x)$ to denote $v_i(\{x\})$ and $v_i(S \cup \{x\})$, respectively. Let $V_{\max} := \max_{i \in [n]} v_i(M)$.

A valuation v is monotone nondecreasing if $v(S \cup x) \geq v(S)$ for all subsets $S \subseteq M$ and all items $x \in M$. Similarly, valuation v is monotone nonincreasing if $v(S \cup x) \leq v(S)$ for all $S \subseteq M$ and $x \in M$. Function v is additive if $v(S) = \sum_{x \in S} v(x)$, for all subsets $S \subseteq M$. We will say that a fair division instance has objective valuations if for each item $x \in M$ (i) either $v_i(S \cup x) \geq v_i(S)$ for all agents i and subsets $S \subseteq M$, (ii) or $v_i(S \cup x) \leq v_i(S)$ for all agents i and subsets $S \subseteq M$. Under (i), the item x is referred to as a good, and when case (ii) holds (and the inequality is strict for some agent i and subset S), we say that x is a chore. We will mainly address objective valuations and, hence, every item is unequivocally either a good or a chore.[6] We will typically use g to denote a good and c to denote a chore. Under objective valuations v_i, for each good g the value $v_i(g) \geq 0$ and for each chore c we have $v_i(c) \leq 0$.

An allocation $\mathcal{A} := (A_1, \ldots, A_n)$ is a partition of items M into n pairwise disjoint subsets. Here, subset of items $A_i \subseteq M$ is assigned to agent i (also called agent i's bundle). At times (such as when analyzing the interim allocations obtained by our algorithms), we may also consider partial allocations wherein not all items are assigned among the agents, i.e., for the pairwise disjoint bundles we have $\cup_{i=1}^{n} A_i \subsetneq M$. Given an allocation $\mathcal{A}$, we say an agent p is poorest if $v_p(A_p) = \min_{i \in N} v_i(A_i)$, and agent r is richest if $v_r(A_r) = \max_{i \in N} v_i(A_i)$.

Equitability and Envy-Freeness. An allocation $\mathcal{A}$ is equitable if $v_i(A_i) = v_j(A_j)$ for all agents $i, j \in N$. An allocation is said to be envy-free if $v_i(A_i) \geq v_i(A_j)$ for all agents $i, j \in N$. Hence, equity requires that all agents have equal value, while envy-freeness requires that each agent values her bundle more than that of any other agent. As mentioned, even in simple examples of a single good and two agents both equitable and envy-free allocations do not exist. Hence, we consider relaxations of these notions.

Specifically, an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is said to be equitable up to any item (EQx) if
(1) For every pair of agents i, j and for each good $g \in A_j$, we have $v_j(A_j \setminus \{g\}) \leq v_i(A_i)$, and
(2) For every pair of agents i, j and for each chore $c \in A_i$, we have $v_i(A_i \setminus \{c\}) \geq v_j(A_j)$.

That is, in an EQx allocation, for each agent, the removal of any good assigned to her makes her a poorest agent, and the removal of any chore assigned to her must make her a richest agent.

Our results for monotone nondecreasing valuations (when all items are goods) are detailed in Section 3. Section 4 presents our results for nonmonotone additive valuations.

3 Monotone Valuations

In this section, all items have nonnegative marginal values. That is, $v_i(S \cup x) \geq v_i(S)$ for all items $x \in M$, agents $i \in N$, and subsets $S \subseteq M$. Hence, all items are goods in this section. We assume a standard value-oracle access to the

[5] The integrality assumption holds without loss of generality for rational values, since we can multiply the rational numbers by the product of their denominators to get integral ones. Note that under such a scaling, the size of the input only increases polynomially.

[6] The main exception here is the example given in Theorem 6, which shows that if the valuations are not objective, then EQx allocations may fail to exist.

valuations, that given any agent $i \in N$ and subset $S \subseteq M$, returns $v_i(S) \in \mathbb{Z}_{\geq 0}$.

For monotone valuations, we establish strong positive results towards the existence of EQx allocations. Our primary result shows that, for general monotone valuations, EQx allocations *always* exist, and for a broad class of valuations—termed *weakly well-layered* (WWL) valuations—such fair allocations can be found in polynomial time.

Definition 1 ((Goldberg, Høgh, and Hollender 2023)). *A valuation function $v : 2^M \to \mathbb{Z}_{\geq 0}$ is said to be* weakly well-layered *if for any set $M' \subseteq M$ the sets $S_0, S_1, \dots$ obtained by the greedy algorithm (that is, $S_0 = \emptyset$ and $S_i = S_{i-1} \cup \{x_i\}$, where $x_i \in \arg\max_{x \in M' \setminus S_{i-1}} v(S_{i-1} \cup x)$, for $i \leq |M'|$) are optimal, in the sense that $v(S_i) = \max_{S \subseteq M' : |S| = i} v(S)$ for all $i \leq |M'|$.*

WWL valuations intuitively capture valuations where optimal sets can be obtained via a greedy algorithm. Several interesting and widely-studied classes of valuations are weakly well-layered, including gross substitutes (which include weighted matroid rank functions), budget-additive, well-layered, and cancelable valuations; Figure 1 in (Goldberg, Høgh, and Hollender 2023) is helpful in visualizing the relation between these classes. Notably, submodular functions are *not* weakly well-layered. This is an affirming observation, since it is known that, even for two agents with identical submodular valuations, obtaining an EFx (and, hence, EQx) allocation is PLS-complete *and* requires exponentially many value queries.

Our results are obtained through a modification and careful analysis of the Add-and-Fix algorithm, earlier used for obtaining EFx under *identical* cost functions (i.e., when all items are chores) (Barman, Narayan, and Verma 2023).

The modified version of Add-and-Fix uses a greedy selection criterion; see Algorithm 1. In each iteration, the algorithm identifies a poorest agent $p \in N$ in the current allocation. Then, agent p selects goods greedily from the unassigned ones. That is, from among the unassigned goods, p iteratively selects goods with *maximum marginal value*, until it is no longer the poorest agent. This is the Add phase in the algorithm (Lines 4 to 6). If after adding these goods, the allocation obtained is not EQx, this must be because of the goods assigned to agent p. In the Fix phase (Lines 7 and 8), violating goods are iteratively removed from agent p's bundle, until the allocation is EQx.

After each iteration, either the value of agent p increases, or the algorithm terminates (Claim 1). We differ from the original Add-and-Fix in two aspects: the original algorithm chose a richest agent in each iteration, since it dealt with chores, whereas we select a poorest agent. Secondly, and crucially for our results for efficient computation, the original algorithm assigned an arbitrary chore to the richest agent, while we select goods with maximum marginal contribution to the poorest agent. Also, note that the EFx guarantee in (Barman, Narayan, and Verma 2023) is obtained for identical costs functions, whereas the EQx result here holds for (monotone) non-identical valuations.

In our proofs, an iteration of the outer while-loop is called an outer iteration. Note that in every outer iteration, the allocation to every agent, other than p, remains unchanged. We will use the following claim.

Algorithm 1: Greedy Add-and-Fix

```
Input: Fair division instance $(N, M, \mathcal{V})$ with value-oracle access to monotone, nondecreasing valuations.
Output: EQx allocation $\mathcal{A}$.
1: Initialize bundles $A_i = \emptyset$ for all agents $i$, and initialize $U = M$ as the set of unassigned goods.
2: while $U \neq \emptyset$ do
3:     Let $p \in \arg\min_{i \in N} v_i(A_i)$ and $p' \in \arg\min_{i \in N \setminus \{p\}} v_i(A_i)$.
       {$p$ and $p'$ are a 'poorest' and 'second poorest' agent.}
       {Add Phase: Lines 4 to 6}
4:     while $v_p(A_p) \leq v_{p'}(A_{p'})$ and $U \neq \emptyset$ do
5:         Let $g^* \in \arg\max_{g \in U}(v_p(A_p \cup g) - v_p(A_p))$.
6:         Update $A_p \leftarrow A_p \cup \{g^*\}$ and $U \leftarrow U \setminus \{g^*\}$.
       {Fix Phase: Lines 7 and 8}
7:     while there exists $\widehat{g} \in A_p$ such that $v_p(A_p \setminus \{\widehat{g}\}) > v_{p'}(A_{p'})$ do
8:         Update $A_p \leftarrow A_p \setminus \{\widehat{g}\}$ and $U \leftarrow U \cup \{\widehat{g}\}$.
9: return Allocation $\mathcal{A} = (A_1, \dots, A_n)$.
```

Claim 1. *After each outer iteration, (i) either the value of the selected agent p strictly increases such that p is no longer the poorest agent (and the values of the other agents remain unchanged), or (ii) all the remaining unassigned goods are allocated to agent p and the algorithm terminates.*

Theorem 2. *Given any fair division instance with monotone valuations, Algorithm 1 computes an EQx allocation in pseudo-polynomial time.*

Proof. We first show that the algorithm terminates in pseudo-polynomial time. Claim 1 implies that the number of outer iterations is at most $\sum_{i \in N} v_i(M) \leq nV_{\max}$, where $V_{\max} := \max_i v_i(M)$. Each outer iteration consists of an Add phase (in which at most m goods are included in agent p's bundle) and a Fix phase (in which at most m goods are removed from agent p's bundle). Each execution of these phases requires at most m calls to the value oracle to find the required goods g^* and $\widehat{g}$. Hence, it follows that the algorithm terminates in $O(m^2 n V_{\max})$ time.

Next, we show that the allocation computed by the algorithm is indeed EQx; our proof is via induction on the number of outer iterations. Initially, the allocation is empty, which is trivially EQx.

For the inductive step, fix any outer iteration and let p be the poorest agent selected in that iteration. Write $\mathcal{A} = (A_1, \dots, A_n)$ for the allocation at the beginning of the outer iteration and $\mathcal{B} = (B_1, \dots, B_n)$ for the allocation obtained after the outer iteration. Note that $B_i = A_i$ for all agents $i \neq p$. This observation and the induction hypothesis imply that any EQx violation must involve agent p. Further, Claim 1 gives us $v_p(B_p) \geq v_p(A_p)$.

To show that allocation $\mathcal{B}$ is EQx, we need to show that for any agent $i \in N$, the removal of any good $g \in B_i$ makes i a poorest agent. This condition holds—via the induction hypothesis—for all agents $i \neq p$; recall that $B_i = A_i$ for all $i \neq p$, and $v_p(B_p) \geq v_p(A_p)$.

For agent p, note that after the completion of the Fix phase, the removal of any good from p's bundle reduces its value to at most $v_{p'}(A_{p'}) = v_{p'}(B_{p'})$. Since p was the poorest and p' was the second poorest agent in allocation $\mathcal{A}$, this implies that the removal of any good from B_p would make agent p the poorest agent in $\mathcal{B}$ as well: $v_p(B_p \setminus \{g\}) \leq v_j(B_j)$ for all $j \in N$ and each good $g \in B_p$.

The theorem stands proved. □

3.1 Weakly Well-Layered Valuations

The following theorem asserts that for WWL valuations, Algorithm 1 computes an EQx allocation in polynomial time.

Theorem 3. *Given any fair division instance in which all the agents have monotone, weakly well-layered valuations, Algorithm 1 computes an EQx allocation in polynomial time.*

Proof. The monotonicity of agents's valuations ensures that the allocation returned by Algorithm 1 is EQx (Theorem 2). Hence, it remains to prove that, under weakly well-layered (WWL) valuations, the algorithm terminates in polynomial time. Towards this, we will show that, in fact, the Fix phase never executes when all the valuations are WWL. Hence, in every outer iteration of Algorithm 1 the number of unassigned goods strictly decreases, and the algorithm terminates in polynomial time.

We will show that the Fix phase never executes via an inductive argument. In the base case (i.e., in the very first outer iteration) we have $A_i = \emptyset$ for all agents i. Now, for the first iteration, write S to denote the subset of goods assigned to the selected agent p in the Add phase. Further, let g^* denote the last good assigned in the Add phase. Then, by the loop-execution condition in Line 4, we have $v_p(S \setminus \{g^*\}) = 0$, since all other agents have value 0. Further, given that v_p is WWL and the set $S \setminus \{g^*\}$ is populated greedily, any set of goods of cardinality $|S| - 1$ has value 0. Hence, upon the removal of any good g from S, agent p has value 0 for the remaining subset, since it has size $|S|-1$. Therefore, the Fix phase (Line 7) will not execute in the first outer iteration.

For the inductive step, fix an outer iteration. Let $\mathcal{A} = (A_1, \ldots, A_n)$ be the allocation at the beginning of the iteration and $\overline{U} = M \setminus (\cup_i A_i)$ be the set of unassigned goods. For the poorest agent $\underline{p}$ selected in the iteration, consider the bundle A_p and write $\overline{S} \subseteq \overline{U}$ to denote the subset of goods assigned to p in the Add phase of the iteration. In addition, let $\overline{g} \in \overline{S}$ be the last good assigned in the Add phase. The following Claim asserts that all strict subsets $T \subsetneq \left(A_p \cup \overline{S}\right)$ have value $v_p(T) \leq v_p\left(\left(A_p \cup \overline{S}\right) \setminus \{\overline{g}\}\right)$.

Claim 4. *For weakly well-layered valuation v_p we have* $\left(A_p \cup \overline{S}\right) \setminus \{\overline{g}\} \in \arg\max_{X \subsetneq A_p \cup \overline{S}} v_p(X)$.

We use Claim 4 to complete the inductive step. The execution criterion of the Add phase (Line 4) implies that before good $\overline{g}$ was included in the agent p's bundle its value was at most $v_{p'}(A_{p'})$, i.e., $v_p\left(\left(A_p \cup \overline{S}\right) \setminus \{g\}\right) \leq v_{p'}(A_{p'})$. Hence, via Claim 4, for every strict subset $T \subsetneq A_p \cup \overline{S}$ we have $v_p(T) \leq v_{p'}(A_{p'})$. The execution condition for the Fix phase will hence not be satisfied. This completes the proof of the theorem. □

3.2 Approximate EQx Allocations

As mentioned previously, for general monotone valuations, computing an EQx allocation is a PLS-hard problem. Complementing this hardness result, this section establishes that an approximately EQx allocation can be computed efficiently. In particular, for parameter $\varepsilon \in [0, 1]$, an allocation $\mathcal{A}$ is said to be an $(1 - \varepsilon)$-EQx allocation if for every pair of agents $i, j \in N$ and for each good $g \in A_i$ we have $(1-\varepsilon)\, v_i(A_i \setminus \{g\}) \leq v_j(A_j)$. Hence, in an $(1-\varepsilon)$-EQx allocation, removing any good from any agent i's bundle brings down i's value to below $\frac{1}{1-\varepsilon}$ times the minimum. Also, note that $\varepsilon = 0$ corresponds to an exact EQx allocation.

We modify Algorithm 1 to obtain an approximately EQx allocation. We replace Lines 4 and 7 in Algorithm 1 with their approximate versions as follows:[7]

4: **while** $(1 - \varepsilon)\, v_p(A_p) \leq v_{p'}(A_{p'})$ **and** $U \neq \emptyset$ **do**...
7: **while** there exists $\widehat{g} \in A_p$ such that $(1-\varepsilon) v_p(A_p \setminus \{\widehat{g}\}) > v_{p'}(A_{p'})$ **do**...

The following theorem provides our main approximation guarantee under monotone valuations. The proof of this result is similar to that of Theorem 2 and is omitted.

Theorem 5. *Given parameter $\varepsilon \in (0, 1)$ and any fair division instance with monotone valuations, a $(1 - \varepsilon)$-EQx allocation can be computed in $O\left(\frac{m^2 n}{\varepsilon} \log V_{\max}\right)$ time.*

4 Nonmonotone Valuations

We now present our results for nonmonotone valuations. We will focus primarily on additive valuations, and will show that even in this case, EQx allocations may not exist. Furthermore, EQx allocations can be hard to compute even if they do exist.

It is known that an EQx allocation can be computed in polynomial time if all the agents have identical, additive valuations (Aziz and Rey 2020). Complementing this positive result, we next show that an EQx allocation may not exist among agents that have nonidentical (and nonmonotone) valuations. Our negative result holds even for two agents with additive, normalized valuations. The fair division instance demonstrating this nonexistence of EQx allocations is given in Table 2. In particular, the instance highlights that if there are items with positive value for one agent and negative value for another—i.e., the valuations are *subjective* rather than objective—then an EQx allocation may not exist. Further, in such instances, it is NP-hard to determine if an EQx allocation exists.

Theorem 6. *An EQx allocation may not exist for two agents with nonmonotone, additive, normalised valuations. Further, in such instances, it is* NP*-hard to determine whether an EQx allocation exists or not.*

[7] Recall that Lines 4 and 7 in Algorithm 1 check the execution condition for the Add and Fix phases, respectively.

	x_1	x_2	x_3
Agent 1	$+1$	-1	$+100$
Agent 2	-1	$+1$	$+100$

Table 2: Example showing nonexistence of EQx allocations in Theorem 6.

Algorithm 2: Two-Way Greedy Algorithm

Input: Instance $(N, M, \mathcal{V})$ with two agents and additive objective valuations.
Output: EQx allocation $\mathcal{A}$.

```
Initialize $\mathcal{A} = (\emptyset, \dots, \emptyset)$ and item set $U := M$.
while $U \neq \emptyset$ do
    Write $r \in \arg\max_{i \in N} v_i(A_i)$ and let $p$ be the other agent ($p \neq r$).
    Set item $g^* \in \arg\max_{g \in U \cap G} v_p(g)$ and item $c^* \in \arg\min_{c \in U \cap C} v_r(c)$. {$g^*$ is most valuable good for $p$ and $c^*$ is least valuable chore for $r$.}
    if $|v_p(g^*)| > |v_r(c^*)|$ then
        Update $A_p \leftarrow A_p \cup \{g^*\}$ and $U \leftarrow U \setminus \{g^*\}$.
    else
        Update $A_r \leftarrow A_r \cup \{c^*\}$ and $U \leftarrow U \setminus \{c^*\}$.
return Allocation $\mathcal{A} = (A_1, \dots, A_n)$.
```

The example for nonexistence is given in Table 2. The computational hardness is established (in the full version of the paper) via a reduction from PARTITION.

Given that under subjective valuations EQx allocations are not guaranteed to exist, we will focus on objective additive valuations in the remainder of this section. We will use C to denote the set of chores and G to denote the set of goods. Since valuations are objective, $M = G \cup C$. Further, given an allocation $\mathcal{A}$, we say agent i has a goods violation if, for some good $g \in A_i$, we have $v_i(A_i \setminus \{g\}) > \min_k v_k(A_k)$. Similarly, agent i has a chores violation if $v_i(A_i \setminus \{c\}) < \max_k v_k(A_k)$ for some chore $c \in A_i$.

4.1 Two Agents

Theorem 7. *Given any fair division instance $(N, M, \mathcal{V})$ with two agents that have additive objective valuations, Algorithm 2 computes an EQx allocation in polynomial time.*

Proof Sketch. The theorem is established via an inductive argument. A key property utilized in the analysis is that, if, in the iteration, item g^* is assigned to agent p, then any chore c assigned previously (i.e., in an earlier iteration) to the other agent r must have greater absolute value. Similarly, if c^* is assigned to agent r in the considered iteration, any good g assigned previously to the other agent p must have greater absolute value. Using these bounds we show that the EQx criterion is maintained as an invariant.

4.2 Identical Chore Valuations

We now show that EQx allocations exist when the agents have additive, objective valuations and they value the chores c identically, i.e., $v_i(c) = v_j(c)$ for all agents $i, j \in N$.

As noted, C denotes the set of chores, and G is the set of goods. For each chore $c \in C$, we will write $v_c < 0$ to denote the common value of the chore among the agents.

For intuition for the proof, consider an allocation $\mathcal{A} = (A_1, \dots, A_n)$ that is not EQx. There are two possibilities. There could be a chores violation – there exist agents i and j and a chore $c \in A_i$ such that $v_i(A_i \setminus \{c\}) < v_j(A_j)$. Here c is called the violating chore. Or, there could be a goods violation – there exist agents i and j and a good $g \in A_j$ such that $v_i(A_i) < v_j(A_j \setminus \{g\})$. Here g a violating good.

Our proof is based on the observation that transferring c to A_j in case of a chores violation and g to A_i under a goods violation leads to a lexicographic improvement in the tuple of values. The lexicographic order here additionally incorporates the sizes of the assigned bundles and the agents' indices for tie-breaking.

Specifically, for any allocation $\mathcal{X} = (X_1, \dots, X_n)$, we define permutation (ordering) $\sigma^{\mathcal{X}} \in \mathbb{S}_n$ over the n agents such that
(i) Agents i with lower values, $v_i(X_i)$, receive lower indices in $\sigma^{\mathcal{X}}$.
(ii) Among agents with equal values, agents i with lower bundle sizes, $|A_i|$, receive lower indices in $\sigma^{\mathcal{X}}$, and
(iii) Then, among agents with equal values and equal number of items, we order by index i.

Plaut and Roughgarden 2020b used a similar idea in their proof of the existence of EFx allocations for *identical* valuations. They introduced the $\preceq_{++}$ comparison operator, which we detail in Algorithm 3. Unlike their work, we use the $\preceq_{++}$ operator to obtain existential guarantees for EQx allocations even when the valuations are nonidentical over the goods (but identical for chores).

If two agents have the same value and the same number of items, then Plaut and Roughgarden's results hold for any arbitrary but consistent tie-breaking between agents. In the definition of $\preceq_{++}$ (see Algorithm 3), we use the index of each agent $i \in [n]$ as a tie-breaker to compare two agents that are otherwise identical (have the same value for their own bundles and the same number of items).

Algorithm 3: $\preceq_{++}$ comparison operator

Input: Allocations $\mathcal{A}$ and $\mathcal{B}$.
Output: **True** if $\mathcal{A} \preceq_{++} \mathcal{B}$, else return **False**.

```
Let $\sigma^{\mathcal{A}}$ and $\sigma^{\mathcal{B}}$ be the defined permutations of the agents associated with allocations $\mathcal{A}$ and $\mathcal{B}$, respectively.
for all $\ell \in [n]$ do
    Set $i = \sigma^{\mathcal{A}}(\ell)$ and set $j = \sigma^{\mathcal{B}}(\ell)$. {$i$ and $j$ are the $\ell$th agents in the permutations.}
    if $v_i(A_i) \neq v_j(B_j)$ then
        return $v_i(A_i) < v_j(B_j)$.
    else if $|A_i| \neq |B_j|$ then
        return $|A_i| < |B_j|$.
    else if $i \neq j$ then
        return $i < j$.
```

Theorem 8 (Theorem 4.1, (Plaut and Roughgarden 2020b))**.** *The comparison operator $\preceq_{++}$ induces a total order on the* *set of all allocations.*

We say an allocation $\mathcal{A}$ is a leximin++ allocation if $\mathcal{B} \preceq_{++} \mathcal{A}$ for all allocations $\mathcal{B}$. Theorem 8 ensures that a leximin++ allocation is guaranteed to exist.

The following theorem then establishes the existential guarantee for EQx allocations in the case of identically-valued chores.

Theorem 9. *In a fair division instance if the agents have additive, objective valuations and they value the chores identically, then the leximin++ allocation is EQx.*

4.3 Additive Goods and a Single Chore

We now show that in instances with additive objective valuations and a single chore c, EQx allocations exist. Note first that in the presence of even a single non-identical chore, the leximin++ allocation may no longer be EQx. Instead, our proof of existence is based on a careful analysis of a version of local search, and keeping track of the movement of the single chore.

The local search algorithm (Algorithm 4) proceeds as follows. Write $\mathcal{A}$ to denote the current allocation. Further, let p be the agent that appears first in the permutation $\sigma^{\mathcal{A}}$ (as defined in the previous subsection) and r be the agent that appears last. Note that p and r are a poorest and a richest agent, respectively. If $\mathcal{A}$ is not EQx, then there must be either a goods violation or a chores violation.

If there is a goods violation in $\mathcal{A}$, then there exists an agent i and good $g \in A_i$ with the property that $v_i(A_i \setminus \{g\}) > v_p(A_p)$. In our algorithm, we resolve the goods violation by transferring good g to agent p. The algorithm resolves all goods violations, before addressing the chores violation. Now, if there is a chores violation, then for some some agent i and the chore $c \in A_i$ it holds that $v_i(A_i \setminus \{c\}) < v_r(A_r)$. The algorithm resolves the chores violation by transferring chore c to agent richest agent r. The algorithm repeats these steps—resolving all goods violations and then the chores violation—until there are no more violations.

By design, the algorithm terminates only when the maintained allocation is EQx. To prove that the algorithm indeed terminates, we need the following notation. For an allocation $\mathcal{A}$, we denote by $\kappa(\mathcal{A})$ as the agent holding the unique chore c, i.e., $c \in A_{\kappa(\mathcal{A})}$. We call the value $v^+(\mathcal{A}) := v_{\kappa(\mathcal{A})}(A_{\kappa(\mathcal{A})} \setminus c)$ as the *cutoff value* of the allocation $\mathcal{A}$ and of the agent $\kappa(\mathcal{A})$.

In case of a chores violation we have $v_{\kappa(\mathcal{A})}(A_{\kappa(\mathcal{A})} \setminus c) < v_r(A_r)$. Write $\mathcal{B}$ to denote the allocation after transferring chore c to agent r. Note that such a chore transfer increases the cutoff value: under allocation $\mathcal{A}$, the cutoff value is $v^+(\mathcal{A}) = v_{\kappa(\mathcal{A})}(A_{\kappa(\mathcal{A})} \setminus c)$ and, under the updated allocation $\mathcal{B}$, it is $v^+(\mathcal{B}) = v_r(A_r)$. In particular, $v^+(\mathcal{A}) < v^+(\mathcal{B})$. In our analysis, we will keep track of two progress measures separately: the lexicographic value of the allocation (according to the order $\preceq_{++}$ defined in Section 4.2) and the cutoff value of the allocation. We will show, in particular, that the cutoff value of the allocation is nondecreasing between successive chores violations, and strictly increases whenever a chores violation is resolved. Whenever a goods violation is resolved, we obtain a lexicographic improvement in the allocation; a chores violation may however result in a lexicographic decrease. Since both the cutoff value and the lexicographic value can only increase a finite number of times, the local search algorithm must terminate in finite time.

Algorithm 4: Algorithm to compute an EQx allocation

```
Input: Fair division instance $(N, M, \mathcal{V})$ with additive, objective valuations and a single chore $c$
Output: EQx allocation $\mathcal{A}$
Initialize $A_1 = M$ and $A_i = \emptyset$ for all agents $i \neq 1$.
while $\mathcal{A} = (A_1, \ldots, A_n)$ is not EQx do
    $p = \sigma^{\mathcal{A}}(1)$. $\{p$ is the first agent according to $\sigma^{\mathcal{A}}.\}$
    while there exists agent $i \in N$ and good $g \in A_i$ such that $v_i(A_i \setminus \{g\}) > v_p(A_p)$ do
        Update $A_p \leftarrow A_p \cup \{g\}$ and $A_i \leftarrow A_i \setminus \{g\}$.
        Update $p = \sigma^{\mathcal{A}}(1)$.
    {After resolving all goods violations, check for chores violation.}
    $r \leftarrow \sigma^{\mathcal{A}}(n)$. $\{r$ is the last agent according to $\sigma^{\mathcal{A}}.\}$
    if $v^+(\mathcal{A}) < v_r(A_r)$ then
        Update $A_{\kappa(\mathcal{A})} \leftarrow A_{\kappa(\mathcal{A})} \setminus \{c\}$ and $A_r \leftarrow A_r \cup \{c\}$.
return $\mathcal{A}$
```

Theorem 10. *Given any fair division instance with additive objective valuations and a single chore, Algorithm 4 terminates in finite time and returns an EQx allocation.*

5 Conclusion and Future Work

Our work resolves fundamental questions regarding the existence of EQx allocations. We present sweeping positive results when all the indivisible items are goods; this includes universal existence of EQx allocations under general, monotone valuations and an accompanying pseudo-polynomial time algorithm. For monotone valuations, we also provide a fully polynomial-time approximation scheme (FPTAS) for finding approximately EQx allocations. In addition, we show that under weakly well-layered valuations EQx allocations can be computed efficiently.

For mixed items (goods and chores), we show that EQx allocations may not exist. For additively-valued goods and chores, our results present a mixed picture: existence and efficient computation for two agents, and existence either when each chore has the same value among the agents, or if there is a single chore.

In fact, all our positive results hold if we interchange goods and chores. For instance, the results in Section 3 hold for monotone nonincreasing valuations (i.e., when all items are chores) as well, with suitably modified definitions. Formal statements are given in the full version of the paper (Barman et al. 2023).

A number of significant open questions remain. First, the existence of EQx allocations under objective, additive valuations remains unresolved. Second, efficient algorithms for computing EQx allocations in this case (or even for goods and a single chore, or identical chores) appear challenging. Lastly, given the practical significance, truthful mechanisms for obtaining EQx allocations may prove useful.

Acknowledgments

Siddharth Barman's research is supported by a SERB Core research grant (CRG/2021/006165). We thank Rohit Vaish for useful discussions regarding the problem, and suggestions for the paper.

References

Aumann, Y.; and Dombb, Y. 2015. The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation (TEAC)*, 3(4): 1–16.

Aziz, H.; and Rey, S. 2020. Almost Group Envy-free Allocation of Indivisible Goods and Chores. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 39–45. ijcai.org.

Barman, S.; Bhaskar, U.; Pandit, Y.; and Pyne, S. 2023. Nearly Equitable Allocations Beyond Additivity and Monotonicity. arXiv:2312.07195.

Barman, S.; Narayan, V. V.; and Verma, P. 2023. Fair Chore Division under Binary Supermodular Costs. In Agmon, N.; An, B.; Ricci, A.; and Yeoh, W., eds., *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, 2863–2865. ACM.

Bhaskar, U.; Misra, N.; Sethia, A.; and Vaish, R. 2023. The Price of Equity with Binary Valuations and Few Agent Types. *CoRR*, abs/2307.06726.

Caragiannis, I.; Kaklamanis, C.; Kanellopoulos, P.; and Kyropoulou, M. 2012. The Efficiency of Fair Division. *Theory of Computing Systems*, 50(4): 589–610.

Cechlárová, K.; Doboš, J.; and Pillárová, E. 2013. On the existence of equitable cake divisions. *Information Sciences*, 228: 239–245.

Cechlárová, K.; and Pillárová, E. 2012. On the computability of equitable divisions. *Discrete Optimization*, 9(4): 249–257.

Chen, X.; and Liu, Z. 2020. The Fairness of Leximin in Allocation of Indivisible Chores. *CoRR*, abs/2005.04864.

Chèze, G. 2017. Existence of a simple and equitable fair division: A short proof. *Mathematical Social Sciences*, 87: 92–93.

Dubins, L. E.; and Spanier, E. H. 1961. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1P1): 1–17.

Foley, D. K. 1966. *Resource allocation and the public sector*. Yale University.

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

Freeman, R.; Sikdar, S.; Vaish, R.; and Xia, L. 2020. Equitable Allocations of Indivisible Chores. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 384–392.

Gal, Y. K.; Mash, M.; Procaccia, A. D.; and Zick, Y. 2017. Which Is the Fairest (Rent Division) of Them All? *J. ACM*, 64(6): 39:1–39:22.

Goldberg, P. W.; Høgh, K.; and Hollender, A. 2023. The Frontier of Intractability for EFX with Two Agents. In Deligkas, A.; and Filos-Ratsikas, A., eds., *SAGT 2023*, volume 14238 of *Lecture Notes in Computer Science*, 290–307. Springer.

Goldman, J.; and Procaccia, A. D. 2015. Spliddit: Unleashing Fair Division Algorithms. *ACM SIGecom Exchanges*, 13(2): 41–46.

Gourvès, L.; Monnot, J.; and Tlilane, L. 2014. Near Fairness in Matroids. In *Proceedings of the 21st European Conference on Artificial Intelligence*, 393–398.

Gourvès, L.; Monnot, J.; and Tlilane, L. 2014. Near Fairness in Matroids. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *ECAI 2014*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, 393–398. IOS Press.

Herreiner, D. K.; and Puppe, C. 2010. Inequality aversion and efficiency with ordinal and cardinal social preferences—An experimental study. *Journal of Economic Behavior & Organization*, 76(2): 238–253.

Herreiner, D. K.; and Puppe, C. D. 2009. Envy freeness in experimental fair division problems. *Theory and decision*, 67: 65–100.

Kagan, J. 2021. Equitable Distribution: Definition, State Laws, Exempt Property. https://www.investopedia.com/terms/e/equitable-division.asp. Accessed: 2023-08-12.

Moulin, H. 2004. *Fair division and collective welfare*. MIT press.

Plaut, B.; and Roughgarden, T. 2020a. Almost Envy-Freeness with General Valuations. *SIAM Journal on Discrete Mathematics*, 34(2): 1039–1068.

Plaut, B.; and Roughgarden, T. 2020b. Almost Envy-Freeness with General Valuations. *SIAM J. Discret. Math.*, 34(2): 1039–1068.

Procaccia, A. D.; and Wang, J. 2017. A lower bound for equitable cake cutting. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 479–495.

Sun, A.; Chen, B.; and Doan, X. V. 2023. Equitability and welfare maximization for allocating indivisible items. *Autonomous Agents and Multi-Agent Systems*, 37(1): 8.

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