

Existence of 2-EFX Allocations of Chores

Jugal Garg¹, Aniket Murhekar²

¹University of Illinois at Urbana-Champaign

²Northwestern University

jugal@illinois.edu, aniket2@illinois.edu

Abstract

We study the fair division of indivisible chores among agents with additive disutility functions. We investigate the existence of allocations satisfying the popular fairness notion of envy-freeness up to any chore (EFX), and its multiplicative approximations. The existence of 4-EFX allocations was recently established in (Garg, Murhekar, and Qin 2025). We improve this guarantee by proving the existence of 2-EFX allocations for all instances with additive disutilities. This approximation was previously known only for restricted instances such as bivalued disutilities (Lin, Wu, and Zhou 2025) or three agents (Afshinmehr et al. 2024).

We obtain our result by providing a general framework for achieving approximate-EFX allocations. The approach begins with a suitable initial allocation and performs a sequence of local swaps between the bundles of envious and envied agents. For our main result, we begin with an initial allocation that satisfies envy-freeness up to one chore (EF1) and Pareto-optimality (PO); the existence of such an allocation was recently established in a major breakthrough by Mahara (2025). We further demonstrate the strength and generality of our framework by giving simple and unified proofs of existing results, namely (i) 2-EFX for bivalued instances (Lin, Wu, and Zhou 2025), (ii) 2-EFX for three agents (Afshinmehr et al. 2024), (iii) EFX when the number of chores is at most twice the number of agents (Kobayashi, Mahara, and Sakamoto 2023), and (iv) 4-EFX for all instances (Garg, Murhekar, and Qin 2025). We expect this framework to have broader applications in approximate-EFX due to its simplicity and generality.

1 Introduction

Fair allocation is a fundamental problem studied extensively across the fields of computer science, economics, mathematics, and multi-agent systems. It is primarily concerned with finding *fair* allocations of resources or responsibilities to agents with heterogeneous preferences. The allocation of indivisible items has received increasing attention in recent years, as it models several important real-world problems like course allocation, inheritance division, and task assignment. These problems can be broadly categorized depending on whether the items are *goods*, i.e., provide utility or value to agents, or *chores*, i.e., provide disutility or cost to agents.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper, we study the fair division problem with indivisible chores where agents have *additive disutility* functions. That is, the disutility an agent incurs by doing a set of chores equals the sum of disutilities of chores in that set. *Envy-freeness* (EF) (Foley 1967) is one of the most fundamental notions of fairness. An allocation $X = (X_1, X_2, \dots, X_n)$ is said to be envy-free if every agent i prefers their own bundle X_i to the bundle X_j assigned to any other agent j . Unfortunately, EF allocations are not guaranteed to exist due to indivisibility — consider allocating a single chore among two agents. This motivated the need to adapt envy-freeness to the discrete setting. Among various relaxations proposed, the notion closest to EF is *envy-freeness up to any chore* (EFX). In an EFX allocation, every agent prefers their bundle to the bundle of another agent, after the removal of any chore from their own bundle. That is, X is EFX if every agent i prefers the bundle $X_i \setminus \{c\}$ to the bundle X_j of any other agent j , for *every* chore $c \in X_i$. Since EFX is a natural and compelling fairness notion in the discrete setting, investigating the existence of EFX is considered an important problem in discrete fair division (Procaccia 2020; Caragiannis et al. 2019).

However, the existence of EFX allocations remains open for general instances with additive disutilities, even when there are only three agents. Several important works have proved the existence of EFX allocations under specific restricted domains, such as two agents or when the number of chores is at most twice the number of agents (Kobayashi, Mahara, and Sakamoto 2023); see Section 1.1 for an expanded discussion of known results. Another popular approach is to investigate the existence of *approximately*-EFX allocations for general instances, without imposing any restrictions on the instance. In a λ -EFX allocation, where $\lambda \geq 1$, the disutility of each agent i after removing any chore from her own bundle is at most λ times the disutility of i for the bundle assigned to any other agent j . Naturally, $\lambda = 1$ corresponds to exactly EFX allocations, and λ -EFX allocations with smaller λ are considered fairer and closer to EFX than allocations with larger λ .

The existence of approximate-EFX allocations of chores has been investigated in several works (Zhou and Wu 2024; Christoforidis and Santorinaios 2024a; Afshinmehr et al. 2024; Garg, Murhekar, and Qin 2025; Lin, Wu, and Zhou 2025). The current best-known approximation is the exist-

tence of 4-EFX allocations for all chore division instances, as established in (Garg, Murhekar, and Qin 2025). They obtain this result by defining and proving the existence of fractional solutions called earning restricted (ER) equilibria, rounding an ER equilibrium to a desirable integral allocation, and designing involved procedures that transfer chores between agents until obtaining a 4-EFX allocation. Our main contribution is to significantly improve this approximation factor to 2, through a simpler approach.

Contribution 1. *For any chore division instance with additive disutilities, a 2-EFX allocation always exists.*

Prior to our work, the existence of 2-EFX allocations was known only under special cases, such as bivalued disutilities (Lin, Wu, and Zhou 2025) or $n = 3$ agents (Afshinmehr et al. 2024). For $n = 3$ agents, the authors remark “*our proof relies on an extensive case study; we raise the question of whether a more elegant proof exists.*” We answer this question positively, as our result subsumes and extends theirs to any number of agents, through a simpler approach. Our approach is to begin with a suitable initial allocation, and perform a series of operations that involve swapping subsets of chores between envious and envied agents, until the allocation is 2-EFX.

The desired initial allocation satisfies a weaker notion of fairness called envy-freeness up to one chore (EF1) and the efficiency notion of Pareto-optimality (PO). An allocation is said to be EF1 if every agent i prefers their bundle to that of any other agent j after the removal of *some* chore from their own bundle. Thus, an EFX allocation is EF1, but not vice versa. It is known that EF1 allocations can be computed in polynomial time (Bhaskar, Sricharan, and Vaish 2020). An allocation is considered PO if there is no other allocation in which some agent receives strictly lower disutility and no agent receives strictly higher disutility. The existence of allocations of chores that are simultaneously EF1 and PO was a challenging open problem, until its recent resolution through a remarkable breakthrough by Mahara (2025). They proved that there always exists an integral allocation X and a set of chore prices p such that (i) agents are only assigned chores that minimize their disutility-to-price ratio, and (ii) the allocation is *price*-EF1, i.e., EF1 in terms of prices. The first condition ensures the allocation is PO (see Proposition 1), and the second condition is stronger than EF1 (see Lemma 1). Using such an allocation as the starting point, we systematically swap subsets of bundles between agents to attain an 2-EFX allocation.

As noted earlier, a similar approach was utilized by (Garg, Murhekar, and Qin 2025) to obtain a 4-EFX allocation by starting with an allocation that satisfies weaker guarantees than price-EF1, which was obtained by rounding a fractional ER equilibrium. We distill the core idea of the approach and find that it has much broader applicability. Our second contribution is to develop a general, unifying framework for obtaining approximate-EFX allocations of chores.

Contribution 2. *We give a general framework for obtaining approximate-EFX allocations of chores to agents with additive preferences.*

Our framework has two components: (1) obtaining a suit-

able initial allocation X that satisfies certain properties, and (2) beginning with X , iteratively performing *chore swaps* until an approximate-EFX allocation is reached. We briefly explain the two components.

We term allocations that are suitable for finding a λ -EFX allocation as λ -EFX-friendly allocations (formally defined in Definition 2). For a λ -EFX-friendly allocation X , there exists a partition of agents into sets N_0 and N_H such that agents in N_0 are λ -EFX, while agents in N_H may not be. Intuitively, this implies that each envious agent $i \in N_H$ has a “high” disutility chore j_i . Let H be the set of such high disutility chores. We require that for all agents $i \in N$, $d_i(X_i \setminus H) \leq \lambda \cdot d_i(j)$ for all $j \in H$. That is, every agent strongly prefers their own bundle of “low” disutility chores over *any* chore in H , up to a multiplicative factor of λ .

Next, we design an algorithm (Algorithm 1) that takes as input a λ -EFX-friendly allocation X and iteratively performs *chore swaps* between an envious agent in N_H and envied agents. In each iteration, we consider an agent $i \in N_H$ who is not λ -EFX in the current allocation Y . Let ℓ be the agent i envies the most. In an (i, ℓ) chore swap, agent i picks up the entire bundle Y_ℓ of agent ℓ and transfers the chore j_i to ℓ . We prove that if we perform such swaps in a carefully chosen order, we obtain a λ -EFX allocation in at most n swaps. We describe and analyze the framework in Section 3.

Our framework is useful as it is simple, algorithmic, and provides a unified method of obtaining approximate-EFX allocations of chores. Several works on approximate-EFX (Zhou and Wu 2024; Christoforidis and Santorinaios 2024a; Afshinmehr et al. 2024; Garg, Murhekar, and Qin 2025; Lin, Wu, and Zhou 2025) rely on extensive case analysis and novel, instance-specific algorithmic techniques. By using our unified framework, we essentially reduce the problem of finding a λ -EFX allocation to that of finding a λ -EFX-friendly allocation. In addition to the state-of-the-art existence result for 2-EFX which subsumes the same result for $n = 3$ agents (Afshinmehr et al. 2024), our framework provides simple, clean, and interpretable proofs of known results in a unified manner, as listed below.

- Existence of 4-EFX allocations (Garg, Murhekar, and Qin 2025).
- Polynomial time algorithm for computing a $(2 - 1/k)$ -EFX allocation in a *bivalued* instance, where all chore disutilities are either 1 or k , for some $k \geq 1$ (Lin, Wu, and Zhou 2025).
- Polynomial time algorithm for computing an EFX allocation when the number of chores is at most twice the number of agents (Kobayashi, Mahara, and Sakamoto 2023; Garg, Murhekar, and Qin 2025).

Moreover, our framework is algorithmic. For obtaining a constant-approximate EFX allocation, we can either begin with a price-EF1 and PO allocation or by rounding the fractional earning restricted (ER) equilibrium. The proof of existence of a price-EF1 and PO allocation (Mahara 2025) crucially uses a non-constructive fixed-point argument to find the right set of chore prices, which does not lead to efficient computation. Further, it is not known if ER equilibria can be computed in polynomial time, although they can be com-

puted fast in practice using the Lemke’s scheme for solving a linear complementarity problem (LCP). Polynomial-time algorithms for either of these two problems would immediately imply a polynomial-time algorithm for computing a constant-approximate EFX allocation. Lastly, we note that our approach is simple. In particular, we can simplify certain components of the algorithm of (Garg, Murhekar, and Qin 2025) using our framework (see Remark 1).

For the above reasons, we believe that our framework will find further applications in computing approximate-EFX allocations, both for general instances and in restricted domains.

Due to space constraints, many proofs are delegated to the full version of our paper (Garg and Murhekar 2025).

1.1 Other Related Work

We mention other related work that is most relevant to EFX and EF1 and PO allocations of chores. We refer the reader to detailed surveys (Aziz et al. 2022; Amanatidis et al. 2023; Liu et al. 2024) for other work on discrete fair division.

Approximate-EFX for Chores. We first discuss results pertaining to all instances with additive disutilities. The first non-trivial result for approximate-EFX was a polynomial time algorithm for computing an $O(n^2)$ -EFX allocation for n agents (Zhou and Wu 2024). This was improved to 4-EFX in (Garg, Murhekar, and Qin 2025). For superadditive disutilities, (Christoforidis and Santorinaios 2024b) showed that EFX allocations do not always exist.

Second, we discuss exact EFX allocations for restricted domains. EFX allocations exist and can be computed in polynomial time for instances with two agents, instances where agents have an identical preference order (IDO) over the chores (Li, Li, and Wu 2022), or two types of chores (Aziz et al. 2023). Kobayashi, Mahara, and Sakamoto (2023) gave matching-based algorithms for computing EFX allocations for instances where (i) the number of chores is at most twice the number of agents, (ii) all but one agent have IDO disutility functions (Kobayashi, Mahara, and Sakamoto 2023), and (iii) there are three agents with 2-ary disutilities (Kobayashi, Mahara, and Sakamoto 2023). For bivalued instances, EFX and PO allocations are known for $n = 3$ agents (Garg, Murhekar, and Qin 2023) or when the number of chores is at most twice the number of agents (Garg, Murhekar, and Qin 2025).

Third, we discuss approximate-EFX allocations for restricted domains. For $n = 3$ agents, 2-EFX allocations were known to exist (Afshinmehr et al. 2024). For bivalued instances, Zhou and Wu (2024) showed the existence of $(n - 1)$ -EFX allocations. This was improved to 3-EFX and PO (Garg, Murhekar, and Qin 2025), and then to slightly better than 2-EFX (Lin, Wu, and Zhou 2025).

EF1 and PO for Chores. The existence of EF1 and PO allocations of chores was recently proved by (Mahara 2025) using a novel fixed-point argument. Polynomial time algorithms are known for two agents (Aziz et al. 2019), three agents (Garg, Murhekar, and Qin 2023), three types of agents (Garg, Murhekar, and Qin 2024b), bivalued disutilities (Ebadian, Peters, and Shah 2022; Garg, Murhekar, and

Qin 2022), and two types of chores (Aziz et al. 2023; Garg, Murhekar, and Qin 2023). Garg, Murhekar, and Qin (2025) proved the existence of 2EF2 and PO and $(n - 1)$ -EF1 and PO allocations for all instances. In an α -EF k allocation, for every agent i , the disutility of i from her own bundle is at most α times the disutility from another agent’s bundle after the removal of k chores from i ’s bundle. The above results are accompanied by polynomial time algorithms for constant number of agents, however a polynomial time algorithm for general number of agents remains open.

2 Preliminaries

Let $[n] = \{1, 2, \dots, n\}$, for any $n \in \mathbb{N}$.

Problem Instance. The chore division problem is to allocate a set $M = [m]$ of m indivisible chores to a set $N = [n]$ of n agents. Agent $i \in N$ has a disutility function $d_i : 2^M \rightarrow \mathbb{R}_{\geq 0}$, where $d_i(S)$ is the disutility (cost) incurred by agent i on doing the set S of chores. We assume that agents have *additive disutilities*, i.e., $d_i(S) = \sum_{j \in S} d_{ij}$, where $d_{ij} > 0$ is the disutility of chore j for agent i . An instance is said to be *bivalued* if there exist $a, b \in \mathbb{R}_{> 0}$ such that $d_{ij} \in \{a, b\}$ for all $i \in N, j \in M$. We scale the disutilities so that $d_{ij} \in \{1, k\}$ for all $i \in N, j \in M$, where $k = \max\{b/a, a/b\}$, and call such an instance a $\{1, k\}$ -bivalued instance.

Allocation. An allocation $X = (X_1, X_2, \dots, X_n)$ is a partition of the chores into n bundles, where X_i denotes the bundle allocated to agent $i \in N$. A *fractional* allocation $y \in [0, 1]^{n \times m}$ allocates chores fractionally, with y_{ij} denoting the fraction of chore j allocated to agent i . In a fractional allocation y , agent i receives disutility $d_i(y_i) = \sum_{j \in M} d_{ij} y_{ij}$. We assume allocations are integral unless specified.

Fairness and Efficiency Notions. An allocation X is:

- λ -Envy-free up to k chores (λ -EF k) if for all $i, h \in N$, ‘s $S \subseteq X_i$ with $|S| \leq k$ such that $d_i(X_i \setminus S) \leq \lambda \cdot d_i(X_h)$. An allocation is simply denoted by EF k if it is 1-EF k .
- λ -Envy-free up to any chore (λ -EFX) if for all $i, h \in N$ and $j \in X_i$, $d_i(X_i \setminus \{j\}) \leq \lambda \cdot d_i(X_h)$. An allocation is simply denoted by EFX if it is 1-EFX.
- Pareto optimal (PO) if there is no allocation Y that dominates X . An allocation Y dominates allocation X if for all $i \in N$, $d_i(Y_i) \leq d_i(X_i)$, and there exists $h \in N$ such that $d_h(Y_h) < d_h(X_h)$.
- Fractionally Pareto-optimal (fPO) if there is no fractional allocation that dominates X . An fPO allocation is clearly PO, but not vice-versa.

Competitive Equilibrium. In a Fisher market for chores, each agent i aims to earn an amount $e_i > 0$ by performing chores in exchange for payment. Each chore j has a price¹ $p_j > 0$ which specifies the payment per unit of the chore. In a fractional allocation x under prices $p = (p_1, \dots, p_m)$, the earning of agent i is $p(x_i) = \sum_{j \in M} p_j \cdot x_{ij}$.

Define the *minimum-pain-per-buck* (MPB) ratio of agent i as $\alpha_i = \min_{j \in M} d_{ij}/p_j$. Let $\text{MPB}_i = \{j \in M \mid d_{ij}/p_j =$

¹We use the term price and payment equivalently

$\alpha_i\}$ denote the set of chores which are MPB for agent i for payments p . We call (x, p) an MPB allocation if $x_i \subseteq \text{MPB}_i$ for all $i \in N$.

An allocation (x, p) is a competitive equilibrium (CE) if each agent i receives a bundle of lowest disutility subject to earning their requirement e_i , and every chore is allocated. For additive disutilities, a CE can equivalently be characterized in terms of MPB ratios: (x, p) is a CE iff (i) for all $i \in N$, $p(x_i) = e_i$, (ii) (x, p) is an MPB allocation, and (iii) for all $j \in M$, $\sum_{i \in N} x_{ij} = 1$.

Competitive equilibria are known to have desirable efficiency and fairness properties.

Proposition 1 (First Welfare Theorem (Mas-Colell, Whinston, and Green 1995)). *Let (x, p) be an MPB allocation. Then x is fPO.*

Let X be an integral allocation and p be a price vector. We let $p_{-k}(X_i) := \min_{S \subseteq X_i, |S| \leq k} p(X_i \setminus S)$ denote the earning of agent i from the bundle X_i excluding her k highest paying chores. Likewise, we let $\hat{p}(X_i) := \max_{j \in X_i} p(X_i \setminus \{j\})$ denote the earning of i from X_i excluding her least priced chore. The following notions are generalizations of price-EF1, defined by (Barman, Krishnamurthy, and Vaish 2018).

Definition 1 (Price EF k and Price EFX). An allocation (X, p) is said to be λ -price envy-free up to k chores (λ -pEF k) if for all $i, h \in N$ we have $p_{-k}(X_i) \leq \lambda \cdot p(X_h)$. Agent i λ -pEF k -envies h if $p_{-k}(X_i) > \lambda \cdot p(X_h)$.

An allocation (X, p) is said to be λ -price envy-free up any chore (λ -pEFX) if for all $i, h \in N$ we have $\hat{p}(X_i) \leq \lambda \cdot p(X_h)$. Agent i λ -pEFX-envies h if $\hat{p}(X_i) > \lambda \cdot p(X_h)$.

The next lemma is a sufficient condition for computing an λ -EF k / λ -EFX and PO allocation, and has been used for computing fair and efficient allocations for both goods and chores (Barman and Krishnamurthy 2019; Barman, Krishnamurthy, and Vaish 2018; Ebadian, Peters, and Shah 2022; Garg, Murhekar, and Qin 2022, 2023, 2024b, 2025).

Lemma 1. *Let (X, p) be an MPB allocation where X is integral.*

- (i) *If (X, p) is λ -pEF k , then X is λ -EF k and fPO.*
- (ii) *If (X, p) is λ -pEFX, then X is λ -EFX and fPO.*

Proof. Since (X, p) is an MPB allocation, Proposition 1 shows X is fPO. Let α_i be the MPB ratio of agent i in (X, p) .

- (i) If (X, p) is λ -pEF k , then for any agents $i, h \in N$:

$$\begin{aligned} \min_{S \subseteq X_i, |S| \leq k} d_i(X_i \setminus S) &= \alpha_i \cdot p_{-k}(X_i) \leq \alpha_i \cdot \lambda \cdot p(X_h) \\ &\leq \lambda \cdot d_i(X_h), \end{aligned}$$

where the first and third (in-)equalities used the MPB condition, and the second used the fact that (X, p) is λ -pEF k . Thus, X is λ -EF k .

- (ii) If (X, p) is λ -pEFX, then for any agents $i, h \in N$:

$$\begin{aligned} \max_{j \in X_i} d_i(X_i \setminus \{j\}) &= \alpha_i \cdot \hat{p}(X_i) \leq \alpha_i \cdot \lambda \cdot p(X_h) \\ &\leq \lambda \cdot d_i(X_h), \end{aligned}$$

where the first and third (in-)equalities used the MPB condition, and the second used the fact that (X, p) is λ -pEFX. Thus, X is λ -EFX. \square

Earning-Restricted Equilibrium. An earning restricted (ER) competitive equilibrium (Garg, Murhekar, and Qin 2025) imposes a restriction c_j on the total payment chore j can provide to the agents. Under these restrictions, an ER equilibrium (x, p) is a partial fractional allocation x and chore prices p where every agent earns their desired earning requirement subject to receiving a bundle of lowest disutility, and the payment from each chore is at most c_j . Formally, (x, p) is an ER equilibrium iff (i) for all $i \in N$, $p(x_i) = e_i$, (ii) (x, p) is an MPB allocation, and (iii) for all $j \in M$, $\sum_{i \in N} p_j \cdot x_{ij} = \min\{p_j, c_j\}$.

Naturally if $\sum_i e_i > \sum_j c_j$, some agent will not receive their earning requirement and an ER equilibrium cannot exist. However, (Garg, Murhekar, and Qin 2025) showed that $\sum_i e_i \leq \sum_j c_j$ is a sufficient condition for the existence of ER equilibria.

Theorem 2 (Garg, Murhekar, and Qin). *An ER Equilibrium exists if and only if $\sum_{i \in N} e_i \leq \sum_{j \in M} c_j$.*

3 General Framework for Approximate-EFX

We present a general framework for constructing approximate-EFX allocations of chores. The framework has two key components: (1) obtaining a suitable initial allocation X that satisfies certain properties, and (2) beginning with X , iteratively performing *chore swaps* until an approximate-EFX allocation is reached.

Initial Allocation. We describe the properties that the initial allocation X should satisfy.

Definition 2 (λ -EFX-friendly allocation). Let X be an allocation with $|X_i| \geq 1$ for all $i \in N$. Let $j_i \in \arg \max_{j \in X_i} d_{ij}$, and let $S_i = X_i \setminus \{j_i\}$. Then X is considered λ -EFX-friendly if there exists a partition $N = N_0 \sqcup N_H$ of the agents that satisfy the following properties:

- (i) For all $i \in N_0$, $d_i(X_i) \leq \lambda \cdot \min_{k \in N_0} d_i(X_k)$.
- (ii) For all $i \in N_0$, $d_i(X_i) \leq \lambda \cdot \min_{h \in N_H} d_i(j_h)$.
- (iii) For all $i \in N_H$, $d_i(S_i) \leq (\lambda - 1) \cdot \min_{k \in N_0} d_i(X_k)$.
- (iv) For all $i \in N_H$, $d_i(S_i) \leq (\lambda - 1) \cdot \min_{h \in N_H} d_i(j_h)$.

We observe using properties (i) and (ii) that agents in N_0 are λ -EFX in a λ -EFX-friendly allocation. Agents in N_H may not be λ -EFX due to the presence of ‘‘high’’ disutility chores from the set $H = \{j_i : i \in N_H\}$. However, for each such agent $i \in N_H$, properties (iii) and (iv) imply that the bundle S_i has lower disutility (upto a $(\lambda - 1)$ factor) than the single chore j_h of any $h \in N_H$ and the bundle X_k of any $k \in N_0$. These properties enable us to address the approximate-EFX-envy of such agents by performing chore swaps.

Chore Swaps. We formally define a *chore swap*, which aims to reduce approximate-EFX-envy among agents by swapping chores between the bundles of envious and envied agents. The idea of chore swaps was introduced in (Garg, Murhekar, and Qin 2025).

Definition 3 (Chore swap). Consider an λ -EFX-friendly allocation X and an agent $i \in N_H$ who is not λ -EFX. Let $j_i \in \arg \max_{j \in X_i} d_{ij}$. Let ℓ be the agent who i envies the most, i.e. $\ell = \arg \min\{h \in N : d_i(X_h)\}$. Then

an (i, ℓ) swap on X results in an allocation X' given by $X'_i = X_i \cup X_\ell \setminus \{j_i\}$, $X'_\ell = \{j_i\}$, and $X'_k = X_k$ for all $k \notin \{i, \ell\}$.

Thus, in an (i, ℓ) swap, the highest disutility chore j_i according to agent i in X_i is transferred to ℓ and ℓ 's entire bundle X_ℓ is given to agent i . The key benefit of a chore swap is that it locally resolves approximate-EFX-envy of agents i and ℓ . This is clearly true for agent ℓ who is assigned a single chore j_i in the allocation X' resulting from the swap. Since i is not λ -EFX in X , we have $d_i(X_\ell) < \lambda^{-1} \cdot d_i(X_i)$, indicating that X_ℓ is of sufficiently low disutility. In addition, we also know that X_ℓ is i 's least disutility bundle, and that properties (iii) and (iv) hold since X is λ -EFX-friendly. We use these observations to argue that the resulting allocation X' is λ -EFX for agent i , and moreover remains λ -EFX-friendly. In particular, we have $d_i(X'_i) \leq \lambda \cdot d_i(j_i)$. We formally prove this in Lemma 2.

A natural algorithm would then involve starting with a λ -EFX-friendly allocation Y , and repeatedly performing chore swaps until a λ -EFX allocation is obtained. However, it could happen that after an (i, ℓ) swap, agent i re-develops λ -EFX-envy after a subsequent swap. Concretely, consider a subsequent swap (h, k) that causes i to develop λ -EFX-envy. Note that i will not λ -EFX-envy h 's bundle after the swap, since i did not λ -EFX-envy k 's bundle before the swap. However, i could λ -EFX-envy k , who has the single chore j_h after the swap. The above scenario indicates that performing swaps in an arbitrary order can cause λ -EFX-envy to re-develop among agents who have participated in a swap earlier. However, we show that we can prevent this from happening by re-allocating the chores $\{j_i\}_{i \in N_H}$, and performing swaps in a particular order.

Re-allocating Chores and Ordering the Swaps. We first note that $d_i(X'_i) \leq \lambda \cdot d_i(j_i)$ holds in the allocation X' that results from an (i, ℓ) swap. The main observation is that if for every subsequent (h, k) swap, we have that $d_i(j_i) \leq d_i(j_h)$, then $d_i(X'_i) \leq \lambda \cdot d_i(j_h)$, and i will not re-develop λ -EFX-envy subsequently. Motivated by this observation, we re-allocate the chores in $H = \{j_i : i \in N_H\}$ to agents in N_H by using a simple round-robin procedure. We set aside the chores in H , arbitrarily order agents in N_H , and ask them to pick their least disutility chore from H one by one. After such a re-allocation of H , we perform (i, ℓ) chore swaps involving λ -EFX-envious agents $i \in N_H$ picked according to the same round-robin order.

3.1 Algorithm Description and Analysis

We put all the above ideas together and design Algorithm 1, which takes as input a λ -EFX-friendly allocation Y and returns a λ -EFX allocation in polynomial time.

Algorithm Description. Let $N = N_0 \sqcup N_H$ be a partition of the agents s.t. the properties in Definition 2 hold for Y . Let $j'_i \in \arg \max_{j \in Y_i} d_{ij}$, and let $H = \{j'_i : i \in N_H\}$. We re-index the agents $N = [n]$ so that $N_H = [r]$ for $r = |N_H|$, and $N_0 = [n] \setminus [r]$. Phase 1 of Algorithm 1 (Lines 4-8) re-allocates H to N_H using a round-robin procedure. In iteration $i \in [r]$, agent i picks out the least disutility chore j_i among the remaining chores in H . Phase 2 of

Algorithm 1 Computes a λ -EFX allocation from a λ -EFX-friendly allocation

Input: A λ -EFX-friendly allocation Y

Output: A λ -EFX allocation X

- 1: Let $N = N_0 \sqcup N_H$ be a partition of the agents s.t. the properties of Definition 2 hold for Y
 - 2: Let $S_i = Y_i \setminus \{j'_i\}$, where $j'_i \in \arg \max_{j \in Y_i} d_{ij}$
 - 3: Let $H \leftarrow \{j'_i : i \in N_H\}$
— Phase 1: Re-allocating H —
 - 4: Let $X_i \leftarrow Y_i$ for all $i \in N_0$; $X_i \leftarrow S_i$ for all $i \in N_H$
 - 5: Let $H' \leftarrow H$
 - 6: **for** $i = 1$ to $|N_H|$ **do**
 - 7: $j_i \leftarrow \arg \min_{j \in H'} d_{ij}$
 - 8: $H' \leftarrow H' \setminus \{j_i\}$
 - 9: $X_i \leftarrow X_i \cup \{j_i\}$
— Phase 2: Chore swaps —
 - 10: **for** $i = 1$ to $|N_H|$ **do**
 - 11: **if** i is not λ -EFX **then**
 - 12: $\ell \leftarrow \arg \min \{d_i(X_h) : h \in N\}$
 - 13: $X_i \leftarrow X_i \cup X_\ell \setminus \{j_i\}$, $X_\ell \leftarrow \{j_i\} \triangleright (i, \ell)$ swap
 - 14: **return** X
-

Algorithm 1 (Lines 9-12) performs chore swaps to eliminate λ -EFX-envy. In iteration $i \in [r]$, if agent i is not λ -EFX, we perform an (i, ℓ) swap involving the agent ℓ who is most envied by i .

Algorithm Analysis. Clearly, Algorithm 1 terminates in at most $2n$ iterations. We prove that it terminates with a λ -EFX allocation. Let X^i be the allocation just *after* iteration i of Phase 2 of Algorithm 1. Let $r = |N_H|$. We prove that the following invariants are maintained during the execution of Algorithm 1.

Lemma 2. *The following hold for each iteration $i \in [r]$ of Phase 2 of Algorithm 1.*

- (i) *Before iteration i , agents in $N_H \setminus [i - 1]$ do not participate in a swap.*
- (ii) *In iteration i , if agent i participates in an (i, ℓ) swap, then i is λ -EFX after the swap. Moreover, $d_i(X^i) \leq \lambda \cdot d_i(j_i)$ immediately after the swap.*
- (iii) *After iteration i , agents in $N_0 \cup [i]$ are λ -EFX.*

Due to space constraints, we defer the proof to the full version. With Lemma 2 in hand, it is straightforward to prove that Algorithm 1 terminates with a λ -EFX allocation.

Theorem 3. *Given as input a λ -EFX-friendly allocation, Algorithm 1 returns a λ -EFX allocation in polynomial time.*

Proof. Clearly, Algorithm 1 terminates in at most $2n$ iterations. By invariant (iii), agents in $N_0 \cup [r]$ are λ -EFX in the allocation X^r at the end of iteration r . Since $N_H = [r]$, we conclude that X^r is λ -EFX for all agents. \square

3.2 Obtaining Improved Guarantees

We show that under some conditions the same framework can lead to improved EFX-approximations, if we begin with an allocation that satisfies weaker conditions than outlined

in Definition 2. Let $L_i = \{j \in M : d_{ij} = \min_{k \in M} d_{ik}\}$ be the set of lowest disutility chores of an agent $i \in N$.

Definition 4 (Weakly λ -EFX-friendly allocation). Let X be an allocation with $|X_i| \geq 1$ for all $i \in N$. Let $j_i \in \arg \max_{j \in X_i} d_{ij}$, and let $S_i = X_i \setminus \{j_i\}$. Then X is considered *weakly λ -EFX-friendly* if there exists a partition $N = N_0 \sqcup N_H$ such that for all $i \in N_H$, $S_i \cap L_i \neq \emptyset$, and the following properties hold:

- (i) For all $i \in N_0$, $\hat{d}_i(X_i) \leq \lambda \cdot \min_{k \in N_0} d_i(X_k)$.
- (ii) For all $i \in N_0$, $\hat{d}_i(X_i) \leq \lambda \cdot \min_{h \in N_H} d_i(j_h)$.
- (iii) For all $i \in N_H$, $\hat{d}_i(S_i) \leq (\lambda - 1) \cdot \min_{k \in N_0} d_i(X_k)$.
- (iv) For all $i \in N_H$, $\hat{d}_i(S_i) \leq (\lambda - 1) \cdot \min_{h \in N_H} d_i(j_h)$.

The main difference between Defs. 2 and 4 is that in the latter, the disutilities on the left-hand side are up to the smallest chore. Similar to Theorem 3, we prove the following:

Theorem 4. *Given as input a weakly λ -EFX-friendly allocation, Algorithm 1 returns a λ -EFX allocation in polynomial time.*

We defer the proof to the full version. As we show in Theorems 7 and 8, we can obtain improved approximation factors if we begin with a weakly λ -EFX-friendly allocation Y and use Theorem 4, instead of beginning with a λ' -EFX-friendly allocation and using Theorem 3, where $\lambda < \lambda'$.

4 Applications

In this section, we give applications of the framework described in Section 3.

4.1 Existence of 2-EFX Allocations

Our main result establishes the existence of 2-EFX allocations for *all* chore division instances with additive disutilities. This substantially improves the previous best-known approximation of 4-EFX (Garg, Murhekar, and Qin 2025).

Theorem 5. *For any chore division instance with additive disutilities, a 2-EFX allocation always exists.*

In a recent breakthrough, Mahara (2025) showed the existence of EF1 and PO allocations for all chore division instances with additive disutilities. In fact, they prove the existence of an allocation X and a price vector p such that (X, p) is an MPB allocation that is pEF1. We prove that X is λ -EFX-friendly for $\lambda = 2$.

Lemma 3. *If (X, p) is an MPB allocation that is pEF1, then X is 2-EFX-friendly.*

Proof. Let us scale the disutilities to prices, i.e., scale disutilities so that $d_{ij} = p_j$ for any $j \in X_i$ and $d_{ij} \geq p_j$ for any $j \notin X_i$. Note that this is without loss of generality since the properties of EF1, EFX, FPO are scale-invariant.

Let $\rho = \min_{i \in N} p(X_i)$ be the earning of the least earner. For each $i \in N$, let $j_i = \arg \max_{j \in X_i} p_j$ be the highest price chore in X_i , and let $S_i = X_i \setminus \{j_i\}$. Since X is pEF1, we know $p(S_i) = p_{-1}(X_i) \leq \rho$, for all $i \in N$. We define $N_0 = \{i \in N : p_{j_i} \leq \rho\}$, and $N_H = \{i \in N : p_{j_i} > \rho\}$. We make the following observations:

1. For any $i \in N_0$, we have $d_i(X_i) = p(X_i) = p(S_i) + p_{j_i} \leq 2\rho$, since X is pEF1 and $i \in N_0$.
2. For any $i \in N_H$, we have $d_i(S_i) = p(S_i) \leq \rho$, since X is pEF1.
3. For any $i \in N$ and $k \in N_0$, we have $d_i(X_k) \geq p(X_k) \geq \rho$, by using the MPB condition and the definition of ρ .
4. For any $i \in N$ and $h \in N_H$, we have $d_i(j_h) \geq p_{j_h} > \rho$, by using the MPB condition and the fact that $h \in N_H$.

It is now straightforward to show X satisfies the conditions of Definition 2 for $\lambda = 2$.

- (i) Using observations (1) and (3), we have $d_i(X_i) \leq 2\rho \leq 2 \cdot d_i(X_k)$ for any $i, k \in N_0$.
- (ii) Using observations (1) and (4), we have $d_i(X_i) \leq 2\rho \leq 2 \cdot d_i(j_h)$ for any $i \in N_0$ and $h \in N_H$.
- (iii) Using observations (2) and (3), we have $d_i(S_i) \leq \rho \leq d_i(X_k)$ for any $i \in N_H$ and $k \in N_0$.
- (iv) Using observations (2) and (4), we have $d_i(S_i) \leq \rho \leq d_i(j_h)$ for any $i, h \in N_H$. \square

Since X is 2-EFX-friendly, Theorem 3 implies that we can compute a 2-EFX allocation in polynomial time from X . Thus, we establish the existence of 2-EFX allocations for all instances with additive disutilities.²

4.2 Computing 4-EFX Allocations

The first constant-factor approximation of EFX for allocating chores to agents with general, additive disutilities was shown by Garg, Murhekar, and Qin (Garg, Murhekar, and Qin 2025). At a high-level, their approach was to (1) define and show the existence of a fractional earning-restricted (ER) competitive equilibrium (x^0, p) , (2) rounding it to an integral MPB allocation (X, p) that is 2-pEF2, and (3) constructing an EFX re-allocation of up to $2n$ ‘‘high’’ price chores, and (4) performing chore swaps on a subset of the agents. The 4-EFX guarantee is eventually obtained through intricate arguments that track the prices of bundles during the course of the algorithm. We re-prove their result using our framework, thereby simplifying both their algorithm and its analysis.

Theorem 6. (Garg, Murhekar, and Qin 2025) *For any chore division instance with additive disutilities, a 4-EFX allocation can be computed in polynomial time given an earning restricted competitive equilibrium as input.*

We defer the full proof to the full version, and present an overview here. When $m \leq 2n$, an EFX allocation can be computed in polynomial time (Kobayashi, Mahara, and Sakamoto 2023; Garg, Murhekar, and Qin 2025); see Theorem 8 for a simplified proof using our framework. Hence, we assume $m > 2n$. We then compute an ER equilibrium (x^0, p) with each agent having an earning requirement $e_i = 1$ and a uniform earning restriction of $\beta = \frac{1}{2}$ on each chore. Let $H = \{j \in M : p_j > \frac{1}{2}\}$ and $L = \{j \in M : p_j \leq \frac{1}{2}\}$

²To be precise, (Mahara 2025) shows that given an instance I , there is a perturbed instance I^ε that admits a pEF1 allocation (X, p) . Theorem 5 implies the existence of a 2-EFX allocation Z for I^ε . For small ε , Z is also 2-EFX for I .

be the set of high price and low priced chores, respectively. We scale the disutilities to prices so that $d_{ij} = p_j$ for $j \in x_i^0$ and $d_{ij} \geq p_j$ for $j \notin x_i^0$. (Garg, Murhekar, and Qin 2025) showed that such a solution can be rounded in polynomial time to an integral MPB allocation (X, p) with the following properties:

- (i) For any i , $|X_i \cap H| \leq 2$.
- (ii) For any i with $|X_i \cap H| = 2$, $p(X_i \setminus H) \leq \frac{1}{2}$.
- (iii) For any i with $|X_i \cap H| = 1$, $p(X_i \setminus H) \leq 1$.
- (iv) For any i with $|X_i \cap H| = 0$, $p(X_i \setminus H) \leq \frac{3}{2}$.
- (v) For any i , $p(X_i) \leq \frac{1}{2}$.

Since agents can earn at most $\frac{1}{2}$ from each chore due to the earning restriction, it must be that $|H| \leq 2n$. Let Z be an EFX allocation of the chores in H . Let Y be the allocation given by $Y_i = (X_i \setminus H) \cup Z_i$, i.e., the high price chores are re-allocated according to Z . We define $N_H = \{i \in N : |Z_i| = 1\}$ to be the set of agents who receive a single high price chore. Let $Z_i = \{j_i\}$ for each $i \in N_H$.

Define $N_0 = N \setminus N_H$. Note that $N_0 = N_L \sqcup N_H^2$, where $N_L = \{i \in N : Z_i = \emptyset\}$, and $N_H^2 = \{i \in N : |Z_i| \geq 2\}$. Moreover, since Z is EFX, either $N_L = \emptyset$ or $N_H^2 = \emptyset$. As shown in (Garg, Murhekar, and Qin 2025), it is possible to allocate Z so that $p(Y_i \setminus H) \leq 1$ for every $i \in N_H^2$.

Using the above observations, we prove that:

Lemma 4. *The above allocation Y is 4-EFX friendly.*

Theorem 3 then implies that a 4-EFX allocation can be computed in polynomial time from Y using Algorithm 1.

Remark 1. We simplify the algorithm of (Garg, Murhekar, and Qin 2025) through our approach. There, the order of performing chore swaps involving agents in N_H is determined as follows. First, the set of chores $H' = \{j_i : i \in N_H\}$ is re-allocated to the agents in N_H through a specific min-cost matching of H' to N_H , computed via a linear program (see Lemma 10 of (Garg, Murhekar, and Qin 2024a)). The dual variables of this LP correspond to a set of chore prices $q \in \mathbb{R}^{|H'|}$. Then, chore swaps involving agents $i \in N_H$ are carried out in non-decreasing order of q_{j_i} . Instead, through our approach, the re-allocation and swap order is determined through a simple round-robin allocation of H' to N_H . This also improves run-time by using a linear-time algorithm instead of solving an LP for matching.

4.3 Bivalued Instances

For $\{1, k\}$ -bivalued instances, Lin, Wu, and Zhou (2025) proved that a $(2 - 1/k)$ -EFX and PO allocation can be computed in polynomial time. This improved the previous known guarantees of 3-EFX and PO (Garg, Murhekar, and Qin 2025) and $O(n)$ -EFX (without PO) (Zhou and Wu 2024). We re-prove this result using our framework.

Theorem 7. (Lin, Wu, and Zhou 2025) *For any chore division instance with $\{1, k\}$ -bivalued disutilities, a $(2 - 1/k)$ -EFX and PO allocation always exists, and can be computed in polynomial time.*

For $\{1, k\}$ -bivalued instances, it is known that an MPB allocation (X, p) that is pEF1 can be computed in polynomial time (Ebadian, Peters, and Shah 2022; Garg, Murhekar,

and Qin 2022). Moreover, all prices are either 1 or k . If X is $(2 - 1/k)$ -EFX, then we are done. Therefore we assume that X is not $(2 - 1/k)$ -EFX. We prove that X is weakly $(2 - 1/k)$ -EFX-friendly, as per Definition 4.

Lemma 5. *The above allocation X is weakly $(2 - 1/k)$ -EFX-friendly.*

Since X is weakly $(2 - 1/k)$ -EFX-friendly, Theorem 4 implies that we can compute a $(2 - 1/k)$ -EFX allocation X' in polynomial time from X . Moreover, we can show that X' is PO by arguing that chore swaps maintain the MPB condition for bivalued instances (see Lemma 22 of (Garg, Murhekar, and Qin 2024a) or Lemma 3.11 of (Lin, Wu, and Zhou 2025)). This proves Theorem 7.

4.4 Small Number of Chores

When the number of chores is at most twice the number of agents, an EFX allocation can be computed in polynomial time. This was first proved by Kobayashi, Mahara, and Sakamoto (2023) using matching-based techniques, and later by (Garg, Murhekar, and Qin 2025) who gave a faster and simpler algorithm using chore swaps. We re-prove this result using our framework.

Theorem 8. (Kobayashi, Mahara, and Sakamoto 2023; Garg, Murhekar, and Qin 2025) *For any chore division instance with additive disutilities where $m \leq 2n$, an EFX allocation exists and can be computed in polynomial time.*

5 Discussion

In this paper, we established the existence of 2-EFX allocations of indivisible chores to agents with additive disutilities. This was previously known only in special cases like bivalued instances (Lin, Wu, and Zhou 2025) or $n = 3$ agents (Afshinmehr et al. 2024), and represents a substantial improvement over the previous known existence of 4-EFX allocations (Garg, Murhekar, and Qin 2025). We achieve this through a general, unifying framework for obtaining approximate-EFX allocations. Moreover, we obtain simpler proofs of existing results (Afshinmehr et al. 2024; Lin, Wu, and Zhou 2025; Garg, Murhekar, and Qin 2025) through our approach.

There remain many important questions for future work, such as improving the approximation factor beyond 2. Another interesting question is if constant-EFX allocations can be computed in polynomial time. Currently, constant-EFX allocations can only be obtained by initializing our framework with a price-EF1 and PO allocation, or a suitable rounding of the earning restricted equilibrium—both of which are not known to be polynomial time computable. However, since our framework is efficient, any advancement on the computation of such allocations would imply a polynomial time algorithm for constant-EFX. Lastly, it is interesting to see if the framework can be applied for other restricted domains, for goods, or for other concepts like charity or surplus.

Acknowledgments

Jugal Garg was supported by NSF Grant CCF-2334461.

References

- Afshinmehr, M.; Ansaripour, M.; Danaei, A.; and Mehlhorn, K. 2024. Approximate EFX and Exact tEFX Allocations for Indivisible Chores: Improved Algorithms. arXiv:2410.18655.
- Amanatidis, G.; Aziz, H.; Birmpas, G.; Filos-Ratsikas, A.; Li, B.; Moulin, H.; Voudouris, A. A.; and Wu, X. 2023. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence*, 322: 103965.
- Aziz, H.; Caragiannis, I.; Igarashi, A.; and Walsh, T. 2019. Fair Allocation of Indivisible Goods and Chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 53–59.
- Aziz, H.; Li, B.; Moulin, H.; and Wu, X. 2022. Algorithmic fair allocation of indivisible items: a survey and new questions. *SIGecom Exch.*, 20(1): 24–40.
- Aziz, H.; Lindsay, J.; Ritossa, A.; and Suzuki, M. 2023. Fair Allocation of Two Types of Chores. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 143–151.
- Barman, S.; and Krishnamurthy, S. 2019. On the Proximity of Markets with Integral Equilibria. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 1748–1755.
- Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018. Finding Fair and Efficient Allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, 557–574.
- Bhaskar, U.; Sricharan, A. R.; and Vaish, R. 2020. On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources. *CoRR*, abs/2012.06788.
- Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Econ. Comput.*, 7(3).
- Christoforidis, V.; and Santorinaios, C. 2024a. On the pursuit of EFX for chores: non-existence and approximations. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI '24*. ISBN 978-1-956792-04-1.
- Christoforidis, V.; and Santorinaios, C. 2024b. On the Pursuit of EFX for Chores: Non-existence and Approximations. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, (IJCAI)*, 2713–2721.
- Ebadian, S.; Peters, D.; and Shah, N. 2022. How to Fairly Allocate Easy and Difficult Chores. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*.
- Foley, D. 1967. Resource Allocation and the Public Sector. *Yale Economic Essays*, 7(1): 45–98.
- Garg, J.; and Murhekar, A. 2025. Existence of 2-EFX Allocations of Chores. arXiv:2507.19461.
- Garg, J.; Murhekar, A.; and Qin, J. 2022. Fair and Efficient Allocations of Chores under Bivalued Preferences. *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 5043–5050.
- Garg, J.; Murhekar, A.; and Qin, J. 2023. New Algorithms for the Fair and Efficient Allocation of Indivisible Chores. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, 2710–2718.
- Garg, J.; Murhekar, A.; and Qin, J. 2024a. Constant-Factor EFX Exists for Chores. arXiv:2407.03318.
- Garg, J.; Murhekar, A.; and Qin, J. 2024b. Weighted EF1 and PO Allocations with Few Types of Agents or Chores. *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*.
- Garg, J.; Murhekar, A.; and Qin, J. 2025. Constant-Factor EFX Exists for Chores. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC '25*, 1580–1589. New York, NY, USA: Association for Computing Machinery. ISBN 9798400715105.
- Kobayashi, Y.; Mahara, R.; and Sakamoto, S. 2023. EFX Allocations for Indivisible Chores: Matching-Based Approach. In *Algorithmic Game Theory (SAGT)*, 257–270.
- Li, B.; Li, Y.; and Wu, X. 2022. Almost (Weighted) Proportional Allocations for Indivisible Chores. In *Proceedings of the ACM Web Conference (WWW) 2022*, 122–131.
- Lin, Z.; Wu, X.; and Zhou, S. 2025. Approximately EFX and PO Allocations for Bivalued Chores. arXiv:2501.04550.
- Liu, S.; Lu, X.; Suzuki, M.; and Walsh, T. 2024. Mixed fair division: A survey. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 22641–22649.
- Mahara, R. 2025. Existence of Fair and Efficient Allocation of Indivisible Chores. arXiv:2507.09544.
- Mas-Colell, A.; Whinston, M.; and Green, J. 1995. *Microeconomic Theory*. Oxford University Press.
- Procaccia, A. D. 2020. Technical Perspective: An Answer to Fair Division’s Most Enigmatic Question. *Commun. ACM*, 63(4): 118.
- Zhou, S.; and Wu, X. 2024. Approximately EFX allocations for indivisible chores. *Artif. Intell.*, 326: 104037.